

SEGURIDAD EN REDES TELEMÁTICAS

Justo Carracedo Gallardo

Universidad Politécnica de Madrid



MADRID • BUENOS AIRES • CARACAS • GUATEMALA • LISBOA • MÉXICO
NUEVA YORK • PANAMÁ • SAN JUAN • SANTAFÉ DE BOGOTÁ • SANTIAGO • SÃO PAULO
AUCKLAND • HAMBURGO • LONDRES • MILÁN • MONTREAL • NUEVA DELHI • PARÍS
SAN FRANCISCO • SIDNEY • SINGAPUR • SAN LUIS • TOKIO • TORONTO

SEGURIDAD EN REDES TELEMÁTICAS

No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sea electrónico, mecánico, por fotocopia, por registro u otros métodos, sin el permiso previo y por escrito de los titulares del Copyright.

Derechos reservados © 2004, respecto a la primera edición en español, por **McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.**

Edificio Valrealty
Basauri, 17, 1.ª planta
28023 Aravaca (Madrid)

ISBN: 84-481-4157-1

Depósito legal: M. 17.763-2004

Editora: Concepción Fernández Madud
Asistente editorial: Amelia Nieva
Diseño de cubierta y figuras: Guadalupe Carracedo
Compuesto en Marasán, S. A.
Impreso en Cofás, S. A.

IMPRESO EN ESPAÑA - PRINTED IN SPAIN

*A todos mis alumnos,
junto con quienes tanto he aprendido.
A Loly, por el apoyo, el aguante y la compañía.*

CONTENIDO

Prólogo	XIII
Capítulo 1. PANORÁMICA DE LA SEGURIDAD EN REDES: HACIA UNA SEGURIDAD CÍVICA	1
1.1. Seguridad para la edad adulta de la telemática	1
¿Por qué es necesaria la seguridad en las redes?	2
¿Qué se entiende por seguridad en redes?	4
Una norma de referencia	4
1.2. Ataques, protocolos y servicios de seguridad	6
Amenazas y ataques	6
Piratas, caballos de Troya, gusanos y otras especies	9
Servicios, mecanismos y protocolos de seguridad	10
1.3. Principales servicios de seguridad	12
Servicio de Autenticación	13
Servicio de Confidencialidad de los datos	14
Servicio de Integridad de los datos	15
Servicio de No Repudio	15
Servicio de Control de Acceso	17
Servicio de Anonimato	18
1.4. Sistemas criptográficos e infraestructuras de seguridad	19
Terminología	20
Criptografía moderna. Criptosistemas de clave secreta	21
Criptosistemas de clave pública	23
Firma digital, certificados y PKIs	24
Uso combinado de la criptografía simétrica y asimétrica	26
1.5. Comunicaciones seguras: hacia una Seguridad Cívica	26
Requisitos de los usuarios, análisis de riesgos y políticas de seguridad	26
Estados, ciudadanos, criminales e ingenieros	27
La confianza en el sistema	29
Seguridad Cívica	31
Capítulo 2. ESCENARIOS DE COMUNICACIÓN Y UBICACIÓN DE LOS SERVICIOS	33
2.1. Escenarios y dominios de seguridad	33
2.2. Determinación de requisitos y análisis de riesgos	35
Los requisitos de usuario	35
Metodologías de análisis y gestión de riesgos	36
2.3. Terceras partes de confianza, TTPs	39
Ubicación de las TTPs	39
Servicios y funciones	41
Uso de múltiples TTPs. Infraestructuras de Seguridad	42

VIII CONTENIDO

2.4.	Interfaces de usuario y tarjetas inteligentes	43
	Características de las tarjetas inteligentes	43
	Interfaces de usuario	46
2.5.	Ubicación de los servicios de seguridad	47
2.6.	Cortafuegos	50
	Funciones que cumple un cortafuego	51
	Diferentes configuraciones de cortafuegos	52
	Limitaciones de los cortafuegos	55
2.7.	Normas y organismos de normalización	55
	ISO e ITU-T	57
	El IETF y las RFCs	58
	ETSI y otros	59
Capítulo 3. FUNDAMENTOS TEÓRICOS DE LA CRIPTOGRAFÍA		61
3.1.	Teoría de la información. Criptosistemas de secreto perfecto	61
	Información e incertidumbre	62
	Redundancia de un lenguaje y criptoanálisis	64
	Secreto perfecto	65
	Un criptosistema de secreto perfecto	65
	Distancia de unicidad	67
	Confusión y difusión	67
3.2.	Teoría de números	67
	Un conjunto finito de números con los que poder operar	68
	Principio de la aritmética modular	70
	Elementos inversos respecto a la multiplicación	72
	Cálculo de inversos. Función indicadora de Euler	73
	Teorema chino de los restos	76
	Campos de Galois del tipo $GF(q^n)$	77
3.3.	Problemas de difícil solución	82
	Complejidad de algoritmos	82
	Algunos problemas de interés	83
Capítulo 4. CRIPTOGRAFÍA SIMÉTRICA, ESTEGANOGRAFÍA Y MARCAS DE AGUA		85
4.1.	De la criptografía clásica a la criptografía moderna	85
	Características principales de la criptografía simétrica	87
	Cifradores de flujo y cifradores de bloque	89
4.2.	Cifrado en bloque: modos de operación	90
4.3.	Criptosistema DES	94
	Funcionamiento de DES	94
	Las claves y el descifrado	97
4.4.	Algoritmo IDEA	98
4.5.	Otros cifradores de bloque	101
4.6.	Un algoritmo criptográfico para el siglo XXI	103
	Hacia un estándar de cifrado avanzado (AES)	103
	La computación cuántica	104
	Rijndael: el heredero de DES	105
	El ciclo básico de Rijndael	107

Esquema de cifrado y descifrado	111
La utilización de Rijndael	112
4.7. Cifrado múltiple	113
4.8. Cifradores de flujo	115
Cifrador de Vernam	116
La secuencia cifrante	117
Generadores de secuencias cifrantes	117
Cifradores de flujo como combinación de varios LFSR	120
Variedad y cualidades de los cifradores de flujo	123
4.9. Un caso aparte: la Esteganografía	123
Dónde insertar el mensaje oculto	124
Ocultación de la información	126
4.10. Marcas de agua	128
Esteganografía, Criptografía, marcas de agua y seguridad cívica	129
Técnicas de marcado	131
Notas al Capítulo 4	133
 Capítulo 5. MECANISMOS DE SEGURIDAD BASADOS EN CRIPTO- GRAFÍA DE CLAVE PÚBLICA	 135
5.1. Características principales de los criptosistemas asimétricos	135
La fortaleza del criptosistema y el tamaño de las claves	137
Gestión y distribución de claves	139
Tamaño de los mensajes y velocidad de procesamiento	139
Una nueva nomenclatura	141
5.2. Provisión de confidencialidad, autenticación e integridad sólo con criptosistemas de clave pública	143
5.3. Mecanismo de firma digital	146
Resumen de m (valor hash)	148
Algoritmo de firma RSA	148
Algoritmo de firma DSA	149
5.4. Uso combinado de la criptografía simétrica y asimétrica	150
Cifrado con clave de sesión generada mediante el mecanismo de envoltura digital	151
5.5. Otros mecanismos criptográficos	153
5.6. Criptosistema RSA	153
Cifrado con la clave pública del receptor	155
Cifrado con la clave privada del emisor	156
Ejemplos de cifrado usando RSA	156
Generación de claves y fortaleza del RSA	159
¿Es fiable el uso de RSA para la Seguridad en Redes?	160
5.7. Algoritmo de Diffie-Hellman para intercambio de claves	161
5.8. Otros algoritmos de clave pública	162
5.9. Cifrado con clave simétrica concertada entre las partes	163
a) Esquemas basados en algoritmos similares al de Diffie-Hellman	163
b) Esquemas basados en algoritmos del mismo tipo que RSA	164
5.10. Una puerta abierta: criptografía de curvas elípticas	165
5.11. ¿Cómo afectará la evolución de la criptografía a la seguridad en redes?	168
Notas al Capítulo 5	169

Capítulo 6. DISTRIBUCIÓN DE CLAVES EN REDES TELEMÁTICAS. CERTIFICADOS Y AUTORIDADES DE CERTIFICACIÓN	171
6.1. Distribución en red de claves simétricas	171
Comunicación extremo a extremo	172
Comunicación centralizada	173
Claves de sesión o claves de tráfico	175
6.2. Depósito y recuperación de claves (<i>Key Escrow</i>)	176
Técnicas para el depósito y captura de claves	177
Pros y contras de la recuperación de claves	179
Situación de las propuestas de recuperación de claves	181
6.3. Distribución en red de claves públicas	181
Dominio homogéneo con pocos usuarios	182
El Certificado	185
Una sola CA en el dominio	186
Varias CAs organizadas jerárquicamente cooperando entre sí	189
Varias CAs de igual jerarquía cooperando entre sí	191
Seguridad de los sistemas basados en certificados	192
Otras formas de organizar la certificación	193
6.4. Formato de los certificados	194
Versiones 1 y 2 del certificado X.509	194
Versión 3 del certificado X.509	197
Los nombres de las entidades en la X.509 v3	200
Extensiones del certificado X.509 v3	201
6.5. Notación ASN.1	205
Tipos y valores	207
Tipos simples	208
Identificadores de objetos	210
Tipos estructurados	212
Otros elementos de la notación	214
Módulos, macros y parametrización	216
Reglas de codificación	217
Para qué sirve ASN.1	221
6.6. Definición formal del certificado X.509	222
Tipos ENCRYPTED, HASHED, SIGNATURE y SIGNED	223
Especificación de los campos del certificado	226
El campo de extensiones	227
Un ejemplo de certificado X.509 v3	230
6.7. Revocación y suspensión de certificados	237
Formato de las Listas de Certificados Revocados (CRLs)	239
Extensiones específicas relacionadas con las CRLs	240
Notas al Capítulo 6	242
Capítulo 7. INFRAESTRUCTURAS DE SEGURIDAD (PKIs Y TTPs) ...	245
7.1. Qué se entiende por infraestructura de seguridad	245
Componentes de una infraestructura de seguridad	248
Características de la infraestructura derivadas de la política de seguridad.	250
7.2. Infraestructuras de certificación. Generalidades	251

Certificación sin CAs	252
Varias CAs de igual jerarquía	254
Múltiples CAs organizadas jerárquicamente	256
Ligeras variaciones del modelo jerárquico puro	258
7.3. El Directorio X.500 como repositorio en las infraestructuras de certificación.	261
Esquema general del Directorio	262
Árbol de Información del Directorio (DIT)	265
Operaciones sobre el Directorio	267
Nombres X.500	268
7.4. Infraestructura de certificación PEM	269
Distintos tipos de CAs presentes en el modelo PEM	270
Subordinación de nombres y políticas de certificación	272
Ventajas e inconvenientes del modelo PEM	273
7.5. Otras infraestructuras con CAs organizadas jerárquicamente	275
Una infraestructura para las administraciones públicas	275
SET: una infraestructura de propósito específico	276
7.6. Hacia modelos flexibles y adaptados a las necesidades de cada grupo de usuarios	279
Elementos de diseño	279
Determinación de las políticas permitidas	281
Restricciones a la política y correspondencia entre políticas	286
Restricciones a la política mediante restricciones en los nombres	288
7.7. La validez del certificado. CRLs y servicios OCSP	290
Revocación en un dominio con una sola CA	291
Disminución del volumen de información: delta-CRLs	292
Puntos de distribución de CRLs y CRLs Indirectas	293
Servicios OCSP	296
7.8. Infraestructura de gestión de privilegios, PMI	298
Certificados de Atributos y Autoridades de Atributos	299
Organización y componentes de una PMI	301
Relación entre la PMI y la PKI	302
7.9. Autoridades de sellado de tiempo: TSAs	303
Componentes del servicio de sellado de tiempo	304
Protocolo de sellado de tiempo: formato de los mensajes	306
Notas al Capítulo 7	308
Capítulo 8. ESCENARIOS DE FIRMA Y AUTENTICACIÓN EN REDES TELEMÁTICAS	311
8.1. Generalidades	311
Autenticación de entidades	312
Autenticación simple, autenticación mediante desafío y autenticación fuerte	313
Autenticación del origen de los datos: MAC, Hash y Firma	318
8.2. Autenticación mediante criptosistemas de clave secreta	319
Códigos de autenticación de mensajes (MAC)	319
Seguridad de los algoritmos MAC	321
Servidores de autenticación: sistema Kerberos	322
8.3. Funciones resumen, o funciones Hash	324
La robustez de las funciones Hash	325

XII **CONTENIDO**

	Uso del valor hash como autenticador de mensajes	326
	Algoritmos hash más significativos	328
8.4.	Firma simple, firma digital y firma electrónica	330
8.5.	Algoritmos de firma digital	332
	Algoritmo de firma digital RSA	332
	Algoritmo de firma digital DSA	334
8.6.	Firma electrónica: una mejora de la firma digital	337
	Validez de la firma digital: el ataque del cumpleaños	337
	Insuficiencias del mecanismo de firma digital	338
	Características de la firma electrónica	339
8.7.	Planteamientos de la firma electrónica	342
	Un marco normativo para la Unión Europea	342
	Definiciones	343
	Necesidad de estándares reconocidos	345
8.8.	Propuestas técnicas: formatos de firma electrónica	346
	Escenario de la Firma Electrónica	348
	Estructura de la Firma Electrónica	349
	Firma con sello de tiempo	351
	Múltiples firmas	353
	Identificación de los signatarios	353
	Firma con información completa para validación	355
8.9.	Especificación formal de la firma electrónica	355
	Mensaje firmado: el tipo SignedData	356
	Información para cada uno de los firmantes: el tipo SignerInfo	359
	Atributos que deben ir firmados	361
	Atributos que no necesitan ser firmados	363
	Especificación de otros formatos de firma electrónica	364
8.10.	Validación de la firma electrónica	365
	Notas al Capítulo 8	368

Capítulo 9. SERVICIOS DE CONTROL DE ACCESO Y NO REPU- DIO

		371
9.1.	Generalidades sobre control de acceso	371
	Amenazas, mecanismos y políticas	372
	Funciones de Control de Acceso	373
9.2.	Mecanismos de Control de Acceso	375
	Listas de Control de Acceso (ACLs)	376
	Esquema basado en Capacidades	379
	Esquemas basados en etiquetas	381
9.3.	Políticas de Control de Acceso	382
	DACs y políticas basadas en la identidad	383
	Políticas MAC	384
	Políticas RBAC	386
9.4.	Certificados de atributos para el control de acceso	388
	Especificación formal del Certificado de Atributos	389
	Tipos de atributos	393
	Extensiones en los certificados de atributos	396
9.5.	Generalidades sobre el servicio de No Repudio	400
	Necesidad de los servicios de No Repudio	400

Fases del servicio de No Repudio	401
9.6. Diferentes tipos de No Repudio	403
No Repudio de Origen	403
No Repudio de Depósito y No Repudio de Envío	404
No Repudio de Entrega y No Repudio de Transporte	406
No Repudio de Creación y No Repudio de Conocimiento	411
Notas al Capítulo 9	411
Capítulo 10. ESPECIFICACIÓN DE POLÍTICAS DE SEGURIDAD: CERTIFICACIÓN Y FIRMA	413
10.1. Políticas de seguridad y proveedores de servicios	413
10.2. CPS y política de certificación	415
Política de certificación	415
Declaración de Prácticas de Certificación (CPS)	416
Relación entre CPS y Política de Certificación	417
10.3. Ligazón entre el certificado, la política de certificación y la CPS	419
Políticas de Certificación (<i>certificatePolicies</i>)	419
Restricciones sobre la política (<i>policyConstraints</i>)	421
10.4. Conjunto de estipulaciones para la certificación	422
Nombres, entidades y aplicabilidad	423
Estipulaciones generales: cuestiones legales y prácticas	424
Identificación y autenticación de entidades	427
Requisitos de operación	429
Controles de seguridad físicos, procedimentales y personales	430
Controles técnicos de seguridad	431
Perfiles del Certificado y de las CRLs	435
10.5. Política y prácticas de sellado de tiempo	435
Distinción entre Política y Prácticas	436
Políticas de sellado de tiempo	437
Prácticas de sellado de tiempo	438
10.6. Una especificación formal de la Política de Firma	440
Atributo identificador de la Política de Firma	440
Estructura global de la especificación sobre Política de Firma	443
Reglas que marca la política	444
Reglas para la entidad firmante y la entidad verificadora	447
Reglas sobre los certificados y su revocación	448
Reglas sobre el sellado de tiempo	451
Condiciones para los atributos y restricciones para los algoritmos	452
Notas al Capítulo 10	453
Capítulo 11. SERVICIOS DE ANONIMATO PARA LA SOCIEDAD DE LA INFORMACIÓN	457
11.1. Criptografía al servicio del anonimato	458
Firma a ciegas sin tercera parte	458
Firma a ciegas: algo más que firmar sin ver	462
Firma a ciegas arbitrada	463
Secreto dividido	467
Secreto compartido	469

XIV **CONTENIDO**

11.2.	Tarjetas inteligentes al servicio del anonimato	472
	Estructura interna de la tarjeta «clásica»	473
	Protecciones, ficheros y autenticaciones	475
	Estructura y funcionamiento de las Tarjetas Java	477
11.3.	Anonimato en los medios de pago	480
	Escenarios de dinero digital anónimo	481
	Características del dinero digital anónimo	486
11.4.	Mecanismos y protocolos para conseguir dinero digital anónimo	488
	Formato del dinero digital	488
	La firma del dinero por parte del Banco	489
	El problema de la reutilización del dinero	492
	El problema de la rastreabilidad	494
11.5.	Votación telemática	495
	Del voto electrónico al voto telemático	495
	Determinación de requisitos para la votación telemática	497
11.6.	Escenarios y protocolos de votación telemática	501
	Autenticación y autorización del votante	503
	Entrega del voto a la Urna	507
	Apertura de la Urna y recuento	510
	Verificación individual del proceso	512
	Auditabilidad y verificación global	516
	Otras características y requisitos del voto telemático	520
11.7.	La tarjeta como elemento constitutivo de los sistemas que proporcionan servicios de anonimato	522
	Autenticación ante la tarjeta	522
	La Aplicación Cliente	526
11.8.	Plataformas para la democracia digital	531
	Diferentes tipos de plataformas	532
	Implantación de sistemas para la Democracia Digital	534
11.9.	Seguridad cívica para la Sociedad de la Información	535
	Los criminales atacan de nuevo	535
	La fortaleza de los sistemas	535
	Necesidad de una cooperación multidisciplinar	537
	Notas al Capítulo 11	538
	Reseña sobre autorizaciones	541
	Bibliografía	543
	Índice analítico	547

PRÓLOGO

La expansión que han alcanzado las redes telemáticas y la vulnerabilidad que presentan debido a su propia configuración hace que el conocimiento de los conceptos y de las técnicas que conforman la Seguridad en Redes sea materia obligada para todas las personas que pretendan estar al tanto de los problemas y de las soluciones que se plantean en este espectacular aumento de las comunicaciones mediante ordenador. Estos problemas y soluciones no se abordarán siempre, ni exclusivamente, desde el ámbito de la ingeniería telemática, pero sí ocuparán un lugar importante dentro de estas tecnologías.

El presente es un libro de telemática dirigido a universitarios en sentido amplio, es decir, pretende ser de utilidad tanto a estudiantes que están tratando de alcanzar un título de grado en el que se incluyan estudios de telemática como a profesionales egresados de facultades y escuelas que en su momento no cursaron estas materias con la profundidad hoy en día requerida. Debido a ello, el tratamiento de los temas está orientado a servir de complemento y de valor añadido (no de introducción) a otras materias que conforman los currículos en los que se aborda el estudio de las redes de comunicación entre ordenadores.

Cabe esperar que los conocimientos y habilidades que a lo largo de estos capítulos se desgranar sean de utilidad a quienes, dentro del área de la Telemática, pretendan dedicarse al diseño, implementación y puesta a punto de sistemas. La selección que he hecho de los distintos temas y de su tratamiento viene avalada por una larga experiencia en la impartición de asignaturas regladas y en múltiples cursos de especialización sobre esta materia (dirigidos a alumnos de la más diversa procedencia y formación), así como por las necesidades prácticas que he detectado durante mi participación en el desarrollo de una amplia gama de sistemas en los que se requieren comunicaciones seguras entre las distintas entidades y agentes telemáticos que en ellos intervienen.

De una parte, este libro pretende ser autocontenido, y de otra pretende servir de base o apoyatura para el estudio de otros servicios y aplicaciones telemáticas. En cuanto a lo primero, se ha procurado hacer una selección mínima de los conocimientos de criptografía (y de los conceptos teóricos en que se fundamenta) imprescindibles para entender adecuadamente los planteamientos y las soluciones presentes en los servicios de seguridad que más adelante se abordan. No se pretende, por tanto, sustituir a los muchos y buenos tratados existentes sobre criptografía, pero sí ofrecer al lector una base suficiente para que el libro sea legible de cabo a rabo sin la exigencia de lecturas complementarias (otra cosa es la conveniencia de tales lecturas, que reiteradamente se recomienda).

En cuanto al carácter complementario de este texto, cabe decir que en los once capítulos que lo componen solamente se abordan los servicios de seguridad fundamentales para la protección de las comunicaciones, pero no se analiza su presencia concreta en todo el abanico de servicios y de aplicaciones presentes en el amplio ámbito de las redes telemáticas, tarea que requiere otro espacio y otra dedicación. Dando por supuesto que, a fecha de hoy, no es de recibo decir que se conoce tal o cual servicio o

aplicación telemática si no se conocen las correspondientes protecciones de seguridad, ese estudio puede abordarse de dos formas distintas. Estudiar antes el servicio telemático y después los servicios de seguridad complementarios sería la primera de ellas, la segunda sería adquirir primero los conocimientos necesarios de seguridad y después estudiar el servicio completo (con seguridad incluida). La redacción de este texto no determina ni una ni otra opción, que puede utilizarse indistintamente dependiendo del criterio de quien lo haga, aunque para el estudio de las aplicaciones telemáticas todo parece indicar que el método más adecuado a seguir es el reflejado en segundo lugar. En lo que toca a agradecimientos, quiero destacar dos grupos de personas bien diferenciadas: aquellas con las que he trabajado de forma más directa y aquellas otras que han realizado propuestas y contribuciones que han tenido una repercusión importante en los trabajos que yo he realizado y, consecuentemente, en los contenidos de este libro.

Entre las primeras quiero recordar a los alumnos junto con los que, a través de trabajos de clase, seminarios y proyectos fin de carrera, hemos abordado tanto cuestiones teóricas varias como experimentaciones prácticas. Asimismo, quiero mostrar mi agradecimiento a todos los compañeros con los que he colaborado desde principios de los noventa en distintos cursos y proyectos de I+D dedicados al tema de la Seguridad en Redes. Como, afortunadamente, han sido muchos, opto por no exponer aquí la relación completa, aunque sí quisiera destacar a dos de ellos con los que de forma más estrecha he colaborado durante los últimos años. A José David Carracedo le debo haberme animado a profundizar en lo importante que es, para el adecuado diseño de los sistemas telemáticos, tener en cuenta las expectativas y los temores de los ciudadanos, y haber colaborado activamente en la promoción de grupos de trabajo multidisciplinarios [Carr99] [Carr01] en los que han colaborado investigadores pertenecientes tanto al campo de la ingeniería telemática como al campo sociopolítico. A Ana Gómez, además de las aportaciones que he obtenido de nuestro trabajo conjunto (situación que también hago extensiva a los restantes miembros del grupo), le debo los trabajos que viene llevando a cabo en la promoción y dirección de varios proyectos (sobre votación telemática y sobre sistemas para la Democracia Digital) en los que se inscriben las experiencias más recientes en las que se apoyan algunos de los capítulos de este libro. De entre las personas pertenecientes al segundo grupo antes citado, quiero mostrar mi agradecimiento a todos aquellos que han colaborado en la ingente tarea de generación de los muchos estándares y recomendaciones en los que me he apoyado en los diversos capítulos (principalmente del 6 en adelante). De forma especial quisiera destacar a tres personas que han tenido una repercusión importante en la orientación de los temas abordados en este libro. A Whitfield Diffie, aparte de mi reconocimiento por el paso de gigante que supuso su contribución al descubrimiento de los algoritmos criptográficos de clave pública, quiero agradecerle su participación, en 1991, en el *workshop* organizado por el proyecto europeo *COST 225 Secure Communications*, del que yo formaba parte, en donde presentó un trabajo sobre el papel de los algoritmos criptográficos en la seguridad en Sistemas Abiertos (es decir, en las redes telemáticas) en un momento en el que estas cuestiones y dependencias estaban aún insuficientemente planteadas. A Steve Kent, además del reconocimiento por sus propuestas pioneras (RFC 1422) acerca de la infraestructura de certificación PEM, quiero agradecerle su participación en otro *workshop* organizado por el proyecto antes citado, celebrado en 1994, justo en el momento en el que yo asumía la coordinación del proyecto, en el que nos presentó sus criterios acerca de la orientación que deberían tomar las PKIs, criterios que he tenido muy en cuenta para la organización del Capítulo 7. A Warwick Ford, además de mi reconocimiento por su iniciativa acerca de dotar de extensiones

al Certificado X.509, y de mi gratitud por sus publicaciones y su participación en varias RFCs que hemos usado aquí como referencia ([Ford94], [ChFo99] y [HPFS02], entre otras), quiero agradecerle su ayuda y sus consejos en la elaboración del artículo [LoCa97] que en 1997 publicamos Lourdes López (a quien también expreso aquí mi agradecimiento) y yo acerca del uso de las extensiones del Certificado, que ha servido como base para una parte importante del Capítulo 7.

Para terminar, quiero también anticipar mi agradecimiento a todos aquellos lectores que me comuniquen las erratas que hayan podido encontrar (que me temo que quedarán, a pesar de la revisión detallada que se ha hecho) y a quienes me hagan saber sus opiniones y sus críticas, tanto sobre lo que está escrito como sobre lo que falta. Pueden hacérmelo saber a través de la dirección de correo: libro.seguridad@proyectos.diatel.upm.es. A todas ellas les prometo tener muy en cuenta sus sugerencias en futuras versiones. Asimismo, quiero agradecer a Ana Gómez y a Sergio Sánchez la ayuda que me han prestado en la revisión de alguno de los temas aquí contenidos.

Madrid, marzo de 2004
Justo Carracedo Gallardo

CAPÍTULO

1

PANORÁMICA DE LA SEGURIDAD EN REDES: HACIA UNA SEGURIDAD CÍVICA

Las redes, debido a su dispersión geográfica y a los múltiples equipos y sistemas que de ellas forman parte, presentan un marco idóneo para posibles ataques y operaciones no autorizadas. El uso malicioso de las redes puede afectar tanto a la seguridad de los sistemas como a la validez de la información que se almacena o transfiere. Modificar y falsificar un documento representado en formato electrónico es mucho más sencillo que hacerlo sobre un documento escrito en papel. Asimismo, debido al acceso remoto, sin ninguna de las cortapisas que introduce una comunicación presencial, es también relativamente sencillo hacerse pasar en la red por quien realmente no se es, con los consiguientes riesgos de pérdida de fiabilidad de la información para quien la recibe.

Esta vulnerabilidad de las redes hace que sea necesario introducir medidas que las protejan contra usos maliciosos. Ello se consigue con una adecuada aplicación de las técnicas de Seguridad en Redes. En este capítulo se estudian los principales conceptos de Seguridad en Redes, que van desde la determinación de las vulnerabilidades hasta la definición de los mecanismos criptográficos y de los servicios telemáticos de seguridad que sirven para protegerlas.

Esta protección de las redes permite que los derechos y libertades de que gozan los ciudadanos en otros ámbitos de comunicación convencionales puedan ser ejercidos también (y más eficazmente) en los entornos de Comunicación Mediante Computadores, CMC. Al entorno configurado conforme a esta perspectiva es a lo que denotamos como *Seguridad Cívica*, que es el tema que se aborda al final del capítulo.

1.1. SEGURIDAD PARA LA EDAD ADULTA DE LA TELEMÁTICA

Lo primero que cabe preguntarse antes de abordar el estudio de una tecnología es en qué consiste, qué interés científico y tecnológico tiene, qué novedad representa y, ante todo, qué beneficios puede reportar al entorno social en el que se pretende implantar.

¿POR QUÉ ES NECESARIA LA SEGURIDAD EN LAS REDES?

Quizás a fecha de hoy esta pregunta sea algo ociosa porque se ha establecido un consenso bastante generalizado acerca de la necesidad de proteger las redes contra usos indebidos. Desde el principio, los sistemas informáticos han presentado una vulnerabilidad muy significativa que ha hecho necesaria la adopción de medidas para evitar actuaciones incorrectas. La más llamativa de ellas, la que primero se tuvo en consideración, fue el *control de acceso*. Obviando hacer referencia aquí a las prevenciones de seguridad física basadas en vigilantes especiales y en la limitación de acceso a determinados locales, cuando se tenían máquinas en las cuales podían trabajar simultáneamente más de una persona (los típicos centros de cálculo), lo primero que se pensó fue crear mecanismos que sirvieran para que los usuarios tuvieran otorgados ciertos privilegios y solamente en base a esos privilegios pudieran hacer qué cosas en la máquina. La famosa *password* o *palabra de paso* es, por así decir, la hermana menor de las medidas de seguridad que propician la identificación del usuario y la implantación de controles de acceso en los sistemas aislados.

Cuando aparecen las redes surge un fenómeno nuevo: los gestores de las máquinas empiezan a no tener un control directo sobre el sistema en su conjunto: las redes propician una suerte de «evasión» a este control. El caso más sencillo puede estar constituido por una red de área local que se extienda a lo largo de un edificio. Si existe información sensible en los servidores de esta red (por ejemplo calificaciones de alumnos en una institución universitaria) existe el peligro de que cualquiera que tenga la habilidad suficiente puede «pincharse» en algunos de los muchos cables de interconexión y, si las medidas de protección no son las adecuadas, pueda tener acceso a su contenido o, incluso, a su modificación. Situaciones que en sistemas aislados podrían afrontarse con medidas sencillas de tipo físico (vigilancia y todo eso) resultan ahora mucho más difíciles de resolver.

Si, además, no se trata sólo de redes locales sino de redes públicas de área extensa, por ejemplo redes IP o X.25, donde la información sensible atraviesa una red en la que el gestor de los sistemas de una determinada organización «no manda», es decir, no tiene capacidad de actuar, cuando esto ocurre, las dudas y las sospechas y los temores sobre actuaciones indebidas aumentan (no digamos ya cuando se trata de múltiples redes interconectadas, como es el caso de Internet). El riesgo de aparición de intrusos es grande y difícil de prevenir porque el usuario de estas redes desconoce casi todo respecto a su estructura, ubicación de nodos, vías de interconexión, etc.

Por todo ello, podemos aseverar que la cuestión de la seguridad es en las redes mucho más importante que en los sistemas informáticos aislados, debido a su mayor vulnerabilidad. Si a ello se une el hecho, al que estamos asistiendo actualmente, de una explosión en la demanda de uso de los servicios telemáticos, estamos en condiciones de lanzar la afirmación, algo contundente, de que hasta que no se introduzca seguridad en los servicios telemáticos no se alcanzará lo que podemos llamar *la edad adulta de la Telemática*. Es decir, es necesaria la incorporación de servicios de seguridad para que los servicios telemáticos puedan adquirir cierto grado de madurez.

Aunque ciertamente algo contundente, esto es fácilmente constatable acudiendo a ejemplos de servicios «pretelemáticos», plenamente implantados, como es el caso del facsímil (*fax*), en los que la información transmitida no puede ser considerada como un documento con validez jurídica o comercial plena debido, precisamente, a la falta de seguridad en su uso: no es posible demostrar que el mensaje fue enviado, que fue recibido o que no ha sido alterado su contenido. Si el correo electrónico, el EDI, el acceso a Web y la transferencia de ficheros pretenden sustituir mayoritariamente

el actual intercambio de información mediante papel, deberán de ofrecer, al menos, las mismas garantías que en la actualidad los medios tradicionales ofrecen.

¿Por qué el telegrama o la carta postal certificada sí tienen validez jurídica y administrativa? Entre otras razones, porque lo entrega una persona que representa al servicio de Correos o al servicio de Telégrafos. No es que sea precisamente el Notario del Reino, pero es una persona que «da la cara» y respalda este acto de comunicación pudiendo confirmar la entrega del documento que le fue encomendado. A simple vista aparecen multitud de dudas sobre la seguridad de estos sistemas tradicionales: dudas sobre las personas encargadas de llevarlo a cabo, sobre la identidad e idoneidad de los receptores, etc. Sin embargo, **socialmente** son aceptados: la gente se fía suficientemente de ellos. Y sin embargo no se fían del correo electrónico tal y como hoy se presenta. Para decir que vamos a sustituir el intercambio de información en papel por el correo electrónico, tenemos que ser capaces de dar una confianza social igual, al menos, a la que ofrece el correo postal. Si no se tienen en cuenta estos requisitos, no será posible su reemplazamiento, a pesar de las innegables ventajas que, en todos los órdenes, ofrece este servicio telemático.

¿Qué compañía u organización social va a tener sus ficheros distribuidos entre distintas oficinas del país accesibles y gestionables mediante un protocolo de transferencia y gestión de ficheros, tipo FTP o FTAM, si no tiene garantías de que alguien extraño a su organización tenga impedido entrar en ellos?

La proliferación de *páginas Web* está suponiendo un cambio sustancial en el concepto de difusión de la información debido a la combinación de información multimedia con la existencia de vínculos que permiten saltar a otra página u objeto sin más que hacer *clic* en un botón. Cuando la información es de libre distribución, esta forma de proceder resulta suficiente y atractiva, pero, no nos confundamos, la realidad nos dice que la tendencia es que parte de esta información sólo sea posible de consultar si previamente el usuario ha sido identificado y se comprueba que satisface determinadas condiciones, frecuentemente de compromiso de pago. Otro tanto puede decirse del tan nombrado *comercio electrónico*, bien a través de Web o de otros sistemas telemáticos.

Hace relativamente poco tiempo que se empezaron a diseñar e implantar redes telemáticas. No obstante, no es necesario insistir en el auge y expansión que las Comunicaciones Mediante Computadores, CMC, ya han alcanzado y en el hecho de que su penetración prosigue extendiéndose cada vez a más esferas sociales. Algunos autores han visto en este proceso un cambio cualitativo en nuestras sociedades, un proceso constituyente de la llamada *Sociedad de la Información* o *Sociedad Red* [Cast96]. Si esa vulnerabilidad de las redes de la que venimos hablando no se corrigiese, ello daría lugar a que ciertos derechos y libertades de que gozan los ciudadanos en otros ámbitos de comunicación convencionales no pudiesen ser ejercidos eficazmente en entornos CMC.

En suma, si en la *Sociedad de la Información* que se avecina gran cantidad de las comunicaciones comerciales, administrativas, o simplemente personales, van a llevarse a cabo usando sistemas telemáticos, es evidente que es necesario implementar mecanismos que den garantía a los ciudadanos de que sus *derechos cívicos* elementales no van a ser vulnerados. Por todo ello es por lo que decíamos más arriba que no es necesario esforzarse en exceso para demostrar que resulta necesario e imprescindible proteger las redes contra usos indebidos.

¿QUÉ SE ENTIENDE POR SEGURIDAD EN REDES?

En principio, el concepto *seguridad en redes* no tiene un significado esencialmente distinto de la noción de seguridad que, en general, aparece en nuestra vida ordinaria: lo que se pretende es *minimizar* (no anular) la vulnerabilidad de los sistemas en su conjunto.

Decimos minimizar porque una idea que pertenece al ámbito ordinario de la protección de recursos y que también está presente en la seguridad en redes es que *no es posible la seguridad total*. Es decir, cuando se construya un escudo de protección contra determinado ataque, siempre será posible concebir un ataque más fuerte que pueda romper ese escudo. ¿Cómo habremos de operar, pues, desde un esquema de ingeniería (que siempre será un esquema con condicionantes económicos) para resolver el problema?

La respuesta es bien clara: las medidas de seguridad que se han de implantar deberán ser proporcionales al *bien* que se intente proteger. Un ejemplo de la vida ordinaria ayudará a entender lo que esto significa: si alguien quiere proteger una vivienda para poderse tomar unas vacaciones sin sobresaltos, posiblemente la simple instalación de una puerta blindada en la entrada de su domicilio pueda ser suficiente para que alguien con un nivel de inteligencia normal no esté interesado en violentarla, ya que el trabajo que le va a costar penetrar dentro es posiblemente mayor que el beneficio que va a obtener; pero si se rumorea que dentro quedan guardadas las joyas de la corona (no hace falta matizar de qué corona) la puerta blindada al uso resulta tan eficaz como cerrar la entrada con un pedazo de cartón.

Llegados a este punto conviene anticipar que, afortunadamente, y a pesar de las salvaguardas establecidas en los párrafos anteriores, con los medios técnicos con los que actualmente contamos, la seguridad que podemos ofrecer es altísima. Todavía existe el problema de la aceptación social de esta validez, pero a pesar del estado incipiente en el que aún se encuentra esta tecnología, es posible proporcionar una seguridad varios órdenes de magnitud superior a la que hoy en día se ofrece en el mundo ordinario del intercambio de documentos en papel. A esta aceptación social está contribuyendo la creciente aparición de normativa y resoluciones judiciales que reconocen valor legal a las operaciones realizadas usando medios electrónicos.

Los mecanismos de seguridad que los simples mortales pueden instalar son tan buenos, tan eficaces, que existe una verdadera desazón entre algunas dependencias gubernamentales acerca de las pocas posibilidades de ruptura de las protecciones establecidas por ciudadanos corrientes. A la altura de estos comentarios introductorios, no es menester abordar el debate acerca de la bondad o no de la prevalencia de los poderes públicos sobre los derechos individuales o viceversa, debate que podrá ser mantenido más juiciosamente cuando se tenga un conocimiento más preciso de los mecanismos que pueden ser puestos en juego.

UNA NORMA DE REFERENCIA

En este como en otros temas relacionados con la comunicación entre ordenadores, la existencia de normas emanadas de organismos internacionales es de capital importancia. De una parte, la presencia de ISO e ITU-T en lo relativo a sistemas conformes con el modelo OSI y de otra Internet y sus RFCs, marcan las pautas a seguir en cuanto a la estructura y funcionamiento de los protocolos y servicios de seguridad. Como en

otros casos, las soluciones no son excluyentes y pueden aplicarse a distintos ámbitos y distintos sistemas sin más que transportar adecuadamente los planteamientos.

El Modelo de Referencia para la Interconexión de Sistemas Abiertos [ISO 7498] es la norma matriz de las arquitecturas telemáticas jerarquizadas en niveles, habiéndose derivado de ella toda la pléyade de normas que han sido desarrolladas durante la década de los ochenta y parte de los noventa bajo el paraguas de ISO y del CCITT (luego convertido en ITU-T). Otro tanto cabe decir de las normas generadas desde los correspondientes comités de Internet dentro del contexto de la pila de protocolos TCP/IP.

Cuando, dentro de ISO, se pretendió elaborar una norma raíz para el tema de Seguridad, se le encomendó a un subcomité específico, el SC 20, su desarrollo. Los

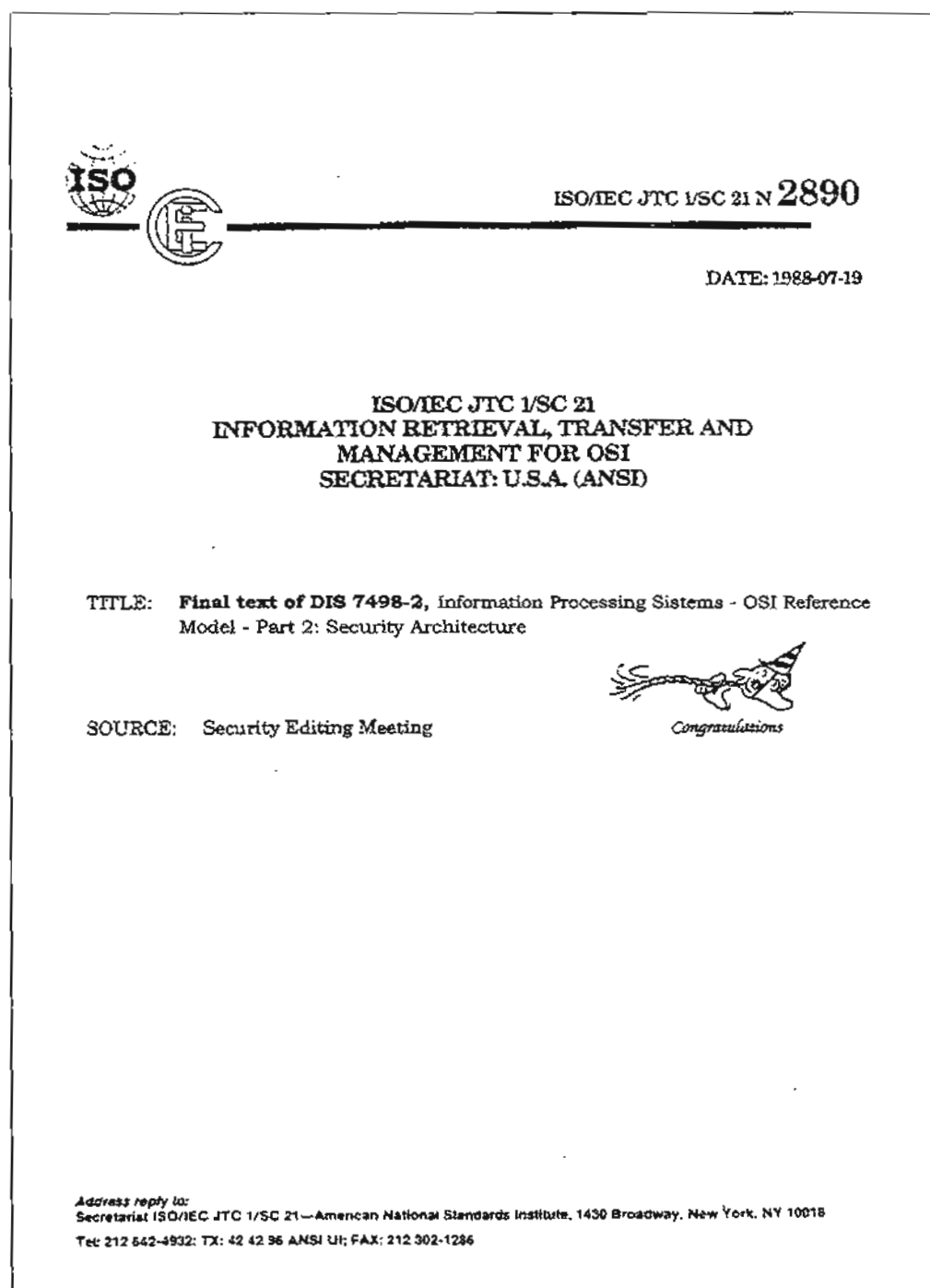


Figura 1.1. Portada de la norma *Security Architecture*.

trabajos de este comité no fueron aprobados y el comité fue disuelto, según parece, por la falta de visión global que sus propuestas contenían y la excesiva focalización del tema de la Seguridad en aspectos relacionados con la Criptografía que, si bien es pieza clave en las soluciones que se adopten, no deja de ser, como se verá más adelante, un simple elemento constituyente de la solución global.

En 1988 fue aprobada la propuesta elaborada por el SC 21, siendo ubicada en el máximo nivel jerárquico de la estructura de normas de ISO, ya que el Modelo OSI «clásico» pasó a ser la norma ISO 7498-Part 1, en tanto que la nueva formulación sobre seguridad pasó a ser la *ISO 7498, OSI Reference Model, Part 2, Security Architecture*. En la Figura 1.1 aparece la portada de la propuesta aprobada tal y como fue distribuida a los miembros del comité, en la que alguien incluyó un dibujito de auto-felicitación por tan venturosa conclusión, circunstancia informal un tanto anómala en un organismo tan circunspecto como ISO. Como explosión y desglose de esta norma han ido apareciendo gran cantidad de recomendaciones específicas sobre los distintos aspectos y servicios.

Además, en abril de 1990 fue creado dentro de ISO un nuevo subcomité específico para temas de seguridad en redes, el SC 27 «*Security Techniques*», para continuar con estas tareas de normalización. Por otra parte, dentro del mundo Internet se vienen produciendo múltiples RFCs sobre el tema. A diferencia de lo que ocurre en otros campos de la Telemática, en el caso de la Seguridad en Redes, la existencia de organismos normalizadores diferentes (principalmente, ITU-T e ISO de una parte y los comités de la Internet Society de otra) no suele representar un conflicto de cara a la compatibilidad entre protocolos, debido al hecho de que los trabajos suelen basarse en los mismos criterios y, habitualmente, las especificaciones emanadas de los distintos comités son aceptadas de forma cruzada.

1.2. ATAQUES, PROTOCOLOS Y SERVICIOS DE SEGURIDAD

En los párrafos siguientes se pasará revista a una serie de conceptos de seguridad que permitan entender más adecuadamente el problema y establecer un vocabulario con el que referirse a los diferentes aspectos. La conceptualización y terminología que se utilizará pertenece en su mayor parte a la antes referida norma genérica *Security Architecture*.

AMENAZAS Y ATAQUES

Una **amenaza** (*threat*, en inglés) es cualquier violación potencial de la seguridad. La información que circula, se procesa y se almacena en una red está sometida a varios tipos de amenazas que pueden ser clasificadas, principalmente, en cuatro grupos:

- Destrucción de la información u otros recursos, quedando éstos inutilizados o desaparecida la información en ellos contenida.
- Modificación de la información, produciendo añadidos, sustracciones o permutas entre sus distintas partes.
- Robo de información o publicación indebida de ésta, de forma que personas diferentes a las legítimamente implicadas tengan conocimiento de ella.
- Interrupción del servicio, consistente en que un determinado usuario deja de tener acceso a un recurso o servicio de la red.

Por otra parte, las amenazas pueden ser *accidentales* o *intencionales*. Amenazas accidentales son aquellas que aparecen de forma no premeditada: disfunciones en los sistemas, fallos de software, operaciones indebidas por parte de algún usuario inexperto, etc. El método para su tratamiento y prevención en el ámbito de las redes debe ser similar a los procedimientos que se siguen en el caso de sistemas informáticos aislados u otros sistemas: revisión periódica de equipos, prueba del correcto funcionamiento de los programas informáticos, mantenimiento de las instalaciones, formación adecuada del personal para evitar errores humanos, etc. Deberán ser tenidas en cuenta en un *Análisis de Riesgos* global, pero su incidencia en los protocolos específicos de seguridad es secundaria.

Otra cosa son las *amenazas intencionales* que presuponen la participación maliciosa de un sujeto o entidad que pretende hacer un uso indebido de la red. Una amenaza intencional se denomina un **ataque** (*attack*, en inglés). Los ataques pueden ser clasificados en *ataques activos* y *ataques pasivos*. Los primeros son aquellos que alteran el comportamiento normal del recurso o servicio telemático que está siendo atacado: una información desaparece, o es cambiada, o un sistema envía los datos hacia direcciones no previstas, etc. Los ataques pasivos, en cambio, no provocan ninguna modificación en el funcionamiento del sistema salvo el uso indebido de sus prestaciones. Es decir, si se produce, por ejemplo, un robo de información a causa de un pinchazo en la red, ésta seguirá fluyendo de forma normal sin variaciones en su contenido y seguirá siendo consumida por los usuarios legitimados (además del ladrón, claro está).

Desde un punto de vista subjetivo los ataques pasivos ofrecen una imagen más asustante, debido, quizás, a que hayamos visto demasiadas películas de espías. No obstante, no puede afirmarse a priori que los ataques pasivos sean más peligrosos que los ataques activos. Ello dependerá de cada caso concreto y de la naturaleza del valor que sea vulnerado.

Entre los tipos de ataques más significativos que pueden presentarse en los entornos telemáticos, cabe destacar los siguientes:

a) **Suplantación de personalidad** (*Masquerade*)

Se produce cuando una persona o entidad suplanta la personalidad de otra, se enmascara como tal, con fines ilícitos. Alguien se hace pasar por quien no es para realizar una función en la red para la que no está autorizado. Este es quizás uno de los ataques más fáciles de imaginar y que con más frecuencia pretende ser perpetrado. Para facilitar al lector el posterior acceso a documentación en inglés se ha incluido aquí y en las restantes definiciones, junto al nombre en español, la correspondiente versión inglesa del término. Algunos autores traducen *masquerade* por *maskarada*, decisión que aquí no se ha seguido por entender que en español ese término tiene más resonancia carnavalesca y lúdica que de fraude malicioso de suplantación de la verdadera identidad. En la Figura 1.2 se representa esquemáticamente este ataque (y los otros tres que hemos destacado) reflejando la actuación de un *usuario malicioso M* que interfiere la comunicación entre la usuaria **A** y el usuario **B**.

b) **Divulgación o repetición del contenido** (*Replay*)

Ocurre cuando una entidad repite un mensaje, o parte de él, para dirigirlo a un destinatario no autorizado. Se trata también de un ataque de muy posible aparición, del cual un ejemplo fácil de imaginar puede ser el caso de alguien que accede indebidamente a un nodo de la red y obtiene parte de la información que por allí circula para

su provecho o para el de un tercero. El típico «pinchazo» de una línea de comunicación para obtener información de forma fraudulenta es también un ataque de divulgación indebida del contenido.

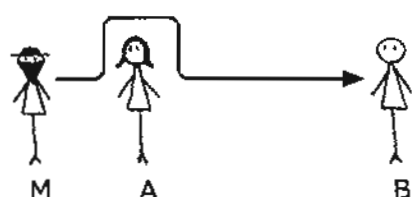
c) Modificación de mensajes (*Modification*)

Consiste en la alteración del contenido de un mensaje (evitando que esta alteración sea detectada) con la finalidad de que su destinatario adopte una decisión o tenga una percepción de la realidad distinta de aquella que se produciría caso de haber recibido el mensaje tal y como fue emitido. El término *mensaje* debe ser entendido aquí (y en las sucesivas ocasiones que de aquí en adelante sea usado) en un sentido amplio, considerado como una pieza de información que circula por la red o se encuentra almacenada en uno de sus sistemas.

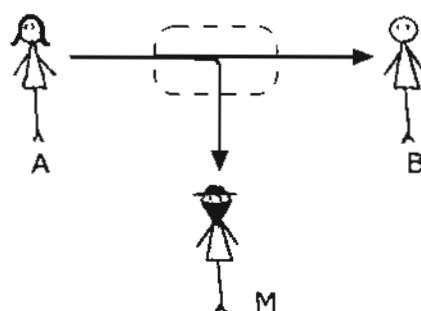
La modificación del mensaje puede consistir en la sustracción o adición de partes de un documento, o en la introducción de variaciones en su contenido. Si alguien accede a una base de datos donde está reflejada su nómina y es capaz de modificar su contenido puede regalarse con una subida de sueldo tan sustancial como su prudencia le dicte.

d) Denegación del servicio (*Denial of Service*)

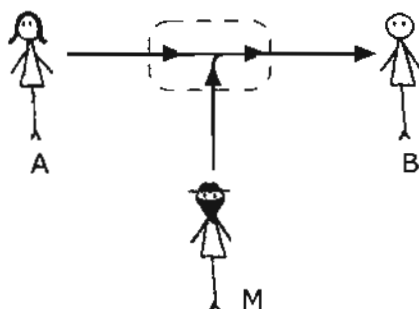
Se produce cuando se consigue, con fines fraudulentos, que una entidad no cumpla deliberadamente su cometido, o se impide que otras entidades cumplan con las funcionalidades que tienen encomendadas. Un ejemplo de competencia ilícita puede con-



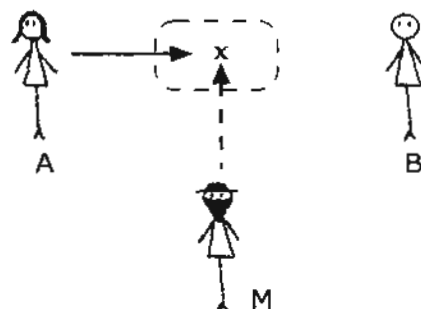
a) Suplantación de personalidad



b) Divulgación del contenido



c) Modificación del mensaje



d) Denegación del servicio

Figura 1.2. Ataques más significativos.

sistir en que un atacante, con el fin de evitar que determinado usuario reciba una posible oferta, modifique ciertas tablas de enrutamiento de la red con objeto de conseguir que ese usuario no reciba correo electrónico durante cierto tiempo o no reciba el procedente de un origen concreto, pudiendo producir con ello un grave quebranto económico o una grave distorsión en su normal desempeño.

PIRATAS, CABALLOS DE TROYA, GUSANOS Y OTRAS ESPECIES

Los cuatro tipos de ataques descritos en los párrafos anteriores resumen las actuaciones malintencionadas más significativas que pueden presentarse en las redes, frente a las que deberán establecerse, por tanto, las medidas de protección. Existen, además, algunos otros términos que hacen referencia a otros tipos de ataques, calificados en cuanto *al modo* de ser llevados a cabo, cuyos efectos o perjuicios puede considerarse que caen dentro de las cuatro categorías antes descritas.

Así, se suele distinguir entre **Ataques Externos** (*Outsiders Attacks*) y **Ataques Internos** (*Insiders Attacks*) para discernir si el atacante pertenece o es ajeno a la organización en cuestión. Naturalmente, el tema de los *insiders* es particularmente peligroso y difícil de corregir: en todas las organizaciones resulta necesario confiar a determinadas personas la custodia de los recursos. Si los propios gestores de la seguridad se corrompen, la cosa se pone difícil. Cuando se organicen entornos de seguridad habrá que procurar que una misma persona no ostente demasiadas responsabilidades, y que responsabilidades críticas sean compartidas por más de una persona.

Se dice que existe una **trapdoor** (llamada en español *trampilla*, *puerta trasera* o *puerta falsa*) cuando una entidad o sistema es alterado para permitir a un atacante concreto un uso no autorizado del recurso. Tiene relación con el diseño deliberadamente doloso de determinados sistemas, que funcionando de forma adecuada y protegida para todos los usuarios, permiten, sin embargo, a uno en concreto (perteneciente al entorno del diseñador del producto) «colarse por la trampilla» que fue instalada de forma deliberada e ilícita. También recibe este nombre la propiedad que presentan algunos programas por la que consiguen anular las limitaciones que tienen impuestas siguiendo alguna pauta secreta conocida por el propio programador.

Otro tipo posible de ataque es el **Caballo de Troya** (*Trojan Horse*), denominado así en recuerdo del regalito que Ulises le hizo a los troyanos por aquel asunto de Helena, que tanto dio que hablar. Cuando se introduce un Caballo de Troya en un sistema, éste adquiere funcionalidades y privilegios no autorizados, además de ejercer sus funciones legítimas. Está también relacionado con vicios de diseño introducidos maliciosamente en el proceso de fabricación de un producto, o con defectos de funcionamiento de los sistemas que permiten que un usuario habilidoso que se haya conectado de forma lícita pueda «prosperar» desde dentro en cuanto a la adquisición de privilegios que excedan a los que, en base a su identidad, el sistema le había concedido inicialmente.

Esto último tiene mucho que ver con las «malas compañías», es decir, con usuarios o sistemas que residen en una misma red u organización y que no cuidan debidamente sus protecciones. Alguna gente que quiere atacar redes intenta «colarse» en una red legal entrando a través de un elemento que no esté bien protegido y una vez dentro convertirse en usuario «normal» de esa red, desde cuya situación procura hacer uso indebido de los recursos de esa red o atacar a otra red ante la cual los usuarios de la anterior tengan ciertos privilegios. Puede ir, así, «escalando» y adquiriendo nuevos privilegios que aumentan la peligrosidad del ataque. Esto es lo que suelen

hacer los *hackers*, que son personas habilidosas capaces de entrar en sistemas cuyo acceso es restringido.

Los *hackers* y los *crackers* son gente que, de alguna manera, se dedica a perturbar el normal desenvolvimiento de las redes. La diferencia entre ambos es algo sutil, ya que a los primeros se les presume una cierta intención lúdica o respaldada por un convencimiento moral de que «las redes son para los que las trabajan». Los *crackers*, en cambio, ofrecen una imagen más perversa y rompedora: son, directamente, los malos de la película. Desde el punto de vista de los intereses comerciales, suele calificarse a los hackers como *piratas informáticos*, por aquello de la navegación a través de las redes. En contrapartida con esa acepción peyorativa, los partidarios del software libre y de la necesidad de que los códigos fuente sean de público conocimiento y uso se autodenominan como «hackers».

Con frecuencia, los *piratas informáticos* suelen ser gente muy joven que tienen además una leyenda gloriosa alimentada por algunas noticias de prensa, más o menos verosímiles, que descubren cómo un adolescente, desde su casa, ha conseguido acceder a redes desde las que se controlan importantes resortes armamentísticos. Estas noticias, de ser ciertas, tendrían más que ver con la incompetencia de ciertos adultos (¿en qué manos, o en qué dedos, estamos?) que con las listezas de esos jóvenes.

Los **virus** son programas bien conocidos que, una vez instalados, destruyen parte de la información almacenada o de los recursos del sistema. En un principio sólo se transmitían a través de disquetes, pero, con el auge de la transferencia de ficheros y de documentos a través de las redes (principalmente vía correo electrónico), representan una amenaza que es necesario conjurar. Pueden crear copias de sí mismos, replicar mensajes de correo y, en general, dañar los datos y los recursos del sistema atacado.

Gusanos (*worms*) suele llamárseles a programas autorreplicables que se transmiten por las redes y provocan la saturación de los sistemas (negación del servicio). Es famoso el denominado *gusano de Internet*, que en 1988 produjo un ataque masivo que afectó a múltiples ordenadores y a consecuencia del cual su autor fue condenado a pagar una fuerte multa. A veces se llama **bacterias** a programas de efectos similares a los de los gusanos, pero diseñados no para transmitirse a través de las redes sino para atacar a ordenadores aislados. Por otra parte, suele llamarse **bomba lógica** a un programa cuyo código ejecuta una determinada perturbación cuando se produce una determinada condición, por ejemplo una fecha.

Como puede apreciarse, toda esta terminología es muy poco rigurosa y raramente volverá a utilizarse en los capítulos que siguen. Para continuar analizando los principales conceptos de seguridad conviene fijar nuestra atención en los posibles ataques que se pueden presentar en las redes telemáticas tal y como los catalogamos en el anterior apartado. Lo realmente sustantivo es el tipo de vulneración que representan, no la forma de llevarla a cabo.

SERVICIOS, MECANISMOS Y PROTOCOLOS DE SEGURIDAD

Un **Servicio de Seguridad** protege las comunicaciones de los usuarios ante determinados ataques. Por eso, reflexionar sobre cuáles son los distintos ataques de que pueden ser objeto las redes telemáticas equivale a pensar en cuáles son los servicios de seguridad que han de establecer las correspondientes protecciones.

Un Servicio de Seguridad no difiere, desde un punto de vista conceptual, de cualquier otro servicio telemático, tal y como se definen en las arquitecturas de redes

jerarquizadas según niveles o capas. Así, se dice que si un determinado *servicio de seguridad* es proporcionado por un *nivel N*, se garantiza a las entidades (N+1) usuarias una determinada protección contra ciertos ataques.

Otro tanto ocurre con el concepto de **Protocolo de Seguridad**. De forma análoga a cualquier otro protocolo telemático, un *protocolo de seguridad* consiste en un conjunto de reglas y formatos que determinan el intercambio de piezas de información, en el que intervienen dos o más entidades (N), y está diseñado para conseguir que sean prestadas a las entidades (N+1) usuarias determinados *servicios de seguridad*. Como puede apreciarse, esta misma definición sería aplicable, en general, a cualquier protocolo telemático a excepción del término «de seguridad» que aparece al final. Además, cabe resaltar que, como en cualquier otro, la especificación del protocolo ha de ser completa y no ambigua.

Por ejemplo, el Correo Electrónico X.400 sirve para el envío y recepción de mensajes bajo determinadas circunstancias. En realidad, el protocolo P1, el conjunto de MTAs y la estructura de la UA permanecen transparentes para el usuario, quien lo único que *ve* es el *servicio* que se le presta (las facilidades) y *cómo* se le presta (la mayor o menor comodidad en el uso de la Interfaz de Usuario). Si añadimos un determinado Servicio de Seguridad al ya existente de Correo Electrónico, el usuario percibirá que la recepción y entrega de mensajes se realiza garantizándole, además, una protección contra ciertos ataques.

Los **Mecanismos de Seguridad** son utilizados para implementar un determinado *servicio de seguridad* o una combinación de ellos. Podríamos decir que los *Mecanismos de Seguridad* son las piezas lógicas, los «ladrillos», con los que se construyen los *protocolos de seguridad*, los cuales son los encargados de proporcionar los *servicios de seguridad*.

Por lo general, los *mecanismos de seguridad* se apoyan en **técnicas criptográficas**, es decir, en la actualidad, la mayoría de los *mecanismos de seguridad* que han sido desarrollados son *mecanismos criptográficos*. Esta permanente presencia de la **Criptografía** y la indudable importancia que el comportamiento de los **criptosistemas** (a los que se dedica el siguiente epígrafe) tienen en el resultado final, no puede oscurecer la idea de que *la Seguridad en Redes es mucho más que Criptografía*. Salvando las distancias, se puede presentar como ejemplo la realización y comportamiento de una base de datos y su relación con la estructura y funcionamiento de la CPU, con sus instrucciones-máquina y sus registros y todo eso. Es indudable la presencia de la CPU y la repercusión que su mejor o peor comportamiento tiene en el funcionamiento global de la base de datos. Es más que conveniente que un diseñador de bases de datos sepa cómo está constituido y cómo funciona un ordenador, pero un conocimiento preciso de ello no le releva de la dificultad de acometer la complejidad del diseño de la base de datos, y es perfectamente posible trabajar en este tipo de diseño sin tener presentes detalles del funcionamiento interno de la máquina. Lo cual no quita, claro está, que una misma persona esté encantada con saber mucho y en detalle de ambas cosas.

En realidad, lo que le importa al usuario final son los servicios que se le ofrecen y la manera en que la **Interfaz de Usuario** le permite hacer uso de las facilidades que el servicio le brinda. La amabilidad de la Interfaz y su adaptación a la forma de trabajar del usuario constituye un punto clave para la extensión de los servicios telemáticos a usuarios corrientes y molientes que no tengan ni quieran tener más familiaridad con los sistemas informáticos que la estrictamente necesaria. En el caso de la Seguridad en Redes, debido a la complejidad inherente a los mecanismos criptográficos, así como a toda la parafernalia que rodea a las múltiples transformaciones que



Figura 1.3. Elementos para la provisión de servicios de seguridad.

hay que realizar sobre la información, esta necesidad se hace aún más apremiante. Es, por tanto, exigible a los diseñadores y fabricantes de sistemas que la dificultad que para el usuario ya representa acceder a un determinado servicio telemático no se vea incrementada en exceso por el hecho de incluir, añadidos, servicios de seguridad.

Es importante percatarse de que los *servicios de seguridad* son un caso particular de los *servicios telemáticos* y representan, en la mayoría de los casos, un valor añadido. Por eso, casi nunca aparecen de forma aislada, sino mejorando la funcionalidad de otro servicio telemático más convencional. Así, por ejemplo, puede decirse que el PGP (Pretty Good Privacy) es un protocolo que proporciona un servicio de correo electrónico que, además, ofrece los servicios de autenticación del origen de los datos, integridad y confidencialidad. (En el siguiente epígrafe se describen estos tres servicios.) Es decir, es un correo electrónico que ofrece funcionalidades similares a las otras aplicaciones corrientes de correo, pero que añade las protecciones que ofrecen los tres servicios de seguridad citados.

1.3. PRINCIPALES SERVICIOS DE SEGURIDAD

Para hacer frente a los *ataques* que puedan presentarse, en la norma de ISO antes referida (ISO/IEC 7498, Part 2, *Security Architecture*) que en 1988 puso los cimientos de la nueva conceptualización de la Seguridad en Redes, se definen cinco servicios básicos de seguridad: *Autenticación*, *Confidencialidad*, *Integridad*, *Control de Acceso* y *No Repudio*.

Hay quienes consideran un sexto servicio básico: *Disponibilidad* (*Availability*). Hace referencia a la protección que es necesario introducir para que las distintas partes del sistema que componen la red estén disponibles para ser utilizadas por quienes dispongan de autorización para ello. Podríamos considerar que esto no constituye realmente un servicio telemático (provisto por un protocolo), sino más bien un conjunto de facilidades y medidas de seguridad que sirven para contrarrestar, en parte, los ataques de denegación de servicio, por lo que no lo incluiremos en la relación de servicios que se discuten en este epígrafe, aunque sí se hará referencia a este tipo de protecciones en el siguiente capítulo.

Además de estos servicios «clásicos» que se definieron a finales de la década de los ochenta, debido a los requisitos que exigen algunos de los nuevos servicios telemáticos que están actualmente siendo vislumbrados (aunque aún escasamente implantados) cabe hablar de un nuevo servicio de seguridad en las redes telemáticas: *Anonimato*.

A continuación se comentan particularizadamente cada uno de estos seis servicios telemáticos de seguridad.

SERVICIO DE AUTENTICACIÓN

El *Servicio de Autenticación* (en inglés, *Authentication*) sirve para garantizar que una entidad comunicante (una persona o una máquina) *es quien dice ser*. En la literatura puede encontrarse también bajo la denominación de «servicio de autenticación», ya que ambos términos son perfectamente válidos en español, aunque «autenticación» es de origen más antiguo y más auténtico (y más breve). Este servicio protege contra un ataque muy fácilmente perpetrable en las redes: la *suplantación de personalidad* (*masquerade*) mediante el cual una entidad remota se hace pasar por quien no es.

Puede tratarse de *Autenticación de entidad simple*, en cuyo caso sólo uno de los participantes en la comunicación (puede ser tanto la entidad origen de los datos como la entidad destino) está obligado a demostrar su identidad. Un ejemplo de ello puede ser el acceso a un servidor remoto que contenga una base de datos que almacene información por cuyo consumo el cliente debe pagar. Previo a autorizarle el acceso y anotarle el correspondiente cargo, si el protocolo de acceso tiene implementado el Servicio de Autenticación, se garantiza a los gestores del Servidor que el usuario que está tratando de acceder a la base de datos es la persona o entidad que proclama ser.

En este mismo ejemplo puede darse el caso de que también el cliente quiera estar seguro de que el servidor al que se ha conectado es el auténtico y no una suplantación maliciosa. Piénsese, por ejemplo, que la información que reciba el cliente puede provocar en él una decisión importante que a alguien, con ánimo fraudulento, le interese orientar. En este caso se requeriría un servicio de *Autenticación mutua* mediante el cual la operación de aseguramiento de la identidad de las entidades se realiza en ambos sentidos de la comunicación.

Podrían distinguirse en este servicio dos calidades, una de las cuales es la *Autenticación débil* y está apoyada en el uso más o menos sofisticado de palabras de paso (*passwords*) o de identificadores, mientras que cuando el resultado es más eficaz la denominamos **Autenticación Fuerte** (*Strong Authentication*), que requiere del intercambio de mensajes cifrados y, posiblemente, del concurso de una Tercera Parte de Confianza, **TTP** (*Trusted Third Party*). Las TTPs son agentes especializados que intervienen en las comunicaciones seguras, cuya constitución y comportamiento será comentado más pormenorizadamente en el siguiente capítulo. Las *tarjetas inteligentes*, que también estudiaremos en el próximo capítulo, representan un componente de seguridad importantísimo de cara a la implantación tanto del *Servicio de Autenticación* como de otros servicios de seguridad.

En otros casos se emplean mecanismos criptográficos robustos que aportan la necesaria seguridad en la protección del acceso.

SERVICIO DE CONFIDENCIALIDAD DE LOS DATOS

Este servicio (*Data Confidentiality* en inglés) proporciona protección para evitar que los datos sean revelados, accidental o deliberadamente, a un usuario no autorizado. Es decir, garantiza que los datos tan sólo van a ser *entendibles* por el destinatario o destinatarios del mensaje. Para ello, el mensaje se alterará de tal manera que aquellas personas que no sean los destinatarios autorizados, aunque lo capturen, no podrán ser capaces de entender su significado.

Cuando es proporcionado, protege a los usuarios de las redes contra un ataque, muy apreciado por ciertos organismos que en teoría están para proteger al ciudadano, consistente en el «pinchazo» (*wiretapping*) de una comunicación para tener acceso, de forma indebida, a la información que por allí circula. Para llevar a cabo estos pinchazos, en la actualidad no es necesario recurrir a escaleras y alicates, ya que se puede acceder indebidamente a un nodo de la red y obtener parte de la información que por allí circula. Este tipo de ataques se encuadraría bajo lo que antes hemos denominado *ataques de divulgación o repetición del contenido (replay)*.

Desde siempre (pensemos, por ejemplo, en el correo postal) se ha considerado como un derecho cívico imprescindible que los poderes públicos garanticen la confidencialidad en las comunicaciones convencionales mediante papel. La verdad es que este derecho ha sido violado de manera reiterada a lo largo de la Historia tomando como justificación una pretendida y más que dudosa defensa de los intereses colectivos. Ahora sí, con la ayuda de un *servicio de confidencialidad* robusto podemos conseguir que la información que circule por las redes, o se almacene en ellas, sólo esté disponible para sus legítimos destinatarios, manteniéndola a salvo de miradas curiosas y malintencionadas.

Con frecuencia, principalmente en la literatura técnica (no así en la de origen social o político) procedente de países de habla inglesa, tiende a considerarse *confidencialidad* como casi sinónimo de *privacidad*. En cuanto a lo que aquí entendemos por el significado de este término, cabe decir que en algunos sitios se traduce la palabra inglesa *privacy* por *intimidad*. La idea de intimidad está más relacionada con la «zona espiritual íntima y reservada de una persona o grupo», como la define el diccionario de la Real Academia. Aquí se ha optado por traducir «*privacy*» por el neologismo *privacidad* (no recogido en el diccionario de la Academia), resaltando su carácter de derecho ciudadano a mantener protegido aquello que afecta a comportamientos sociales que sólo incumben a una persona o un grupo reducido de ellas. Podríamos, por tanto, considerar que la privacidad es la extensión de la intimidad a aspectos más formales y públicos relacionados con las sociedades modernas y sus dinámicas de mercantilización.

Si bien es cierto que en multitud de casos la existencia de un *servicio de confidencialidad* fiable y robusto es fundamental e imprescindible para la obtención de la *privacidad*, no deben confundirse ambos conceptos, ya que, como tratará de justificarse más adelante, el uso coordinado de los servicios y políticas de seguridad es el que proporciona, en su conjunto y dependiendo de cada caso, la necesaria *privacidad* en las operaciones que realizan los ciudadanos a través de las redes. De hecho, en muchos casos es necesaria la inclusión del Servicio de Anonimato, que veremos más adelante, para obtener la privacidad.

SERVICIO DE INTEGRIDAD DE LOS DATOS

El *Servicio de Integridad de los datos* (en inglés, *Data Integrity*) garantiza al receptor del mensaje que los datos recibidos coinciden exactamente con los enviados por el emisor de los mismos, de tal forma que puede tener garantías de que a la información original no le ha sido añadida, ni modificada, ni sustraída alguna de sus partes. Es decir, el receptor de la información (o el proveedor del servicio) detectará si se ha producido o no un ataque de *modificación del mensaje*, lo que le permitirá rechazar o dar por buenos los datos recibidos.

No es necesario remarcar la importancia que tiene este servicio para una adecuada expansión de los servicios telemáticos. En los procedimientos ordinarios de intercambio de información mediante papel u otros soportes convencionales no ha resultado muy difícil para los falsificadores alterar mediante sustituciones el contenido de un mensaje. Pero se requiere bastante habilidad para llevarlo a cabo sin que se note el fraude. Además, las posibilidades de sustitución que tiene el falsificador están muy limitadas por el formato del documento: puede sustituir una palabra o una frase por otra de similar tamaño, pero no puede insertar párrafos demasiado grandes sin alterar sustancialmente la estructura del documento. Estas dificultades hacen que los usuarios ante, por ejemplo, un papel firmado no teman en exceso este tipo de ataques.

No obstante, cuando se trata de mensajes o datos en soporte electrónico, las posibilidades de hacer modificaciones sin dejar huella están al alcance de cualquiera. Piénsese, por ejemplo, en que alguien quiera presentar ante un organismo cualquiera, como elemento probatorio, un mensaje de correo impreso en el que se respalde cualquier circunstancia favorable a quien lo presenta. Resulta evidente que si no presenta una prueba robusta (por lo general será una prueba criptográfica) de la integridad del mensaje, nadie en su sano juicio va a aceptar ese mensaje como una prueba válida (cualquiera puede redactar e imprimir lo que le parezca oportuno y darle el formato que tendría un mensaje de correo). Por todo ello, resulta absolutamente imprescindible la provisión del servicio de integridad cuando se trata de intercambiar mensajes, con cierta garantía, a través de redes telemáticas.

Como se verá en próximos capítulos, es relativamente sencillo desarrollar protocolos que proporcionen, a un tiempo, los servicios de *autenticación*, de *integridad* y de *confidencialidad*. La mayor simplicidad de estos procedimientos deriva del hecho de que pueden implementarse bajo el paradigma de *extremo a extremo (end to end)*, es decir, participando solamente la entidad emisora y la entidad receptora, sin el concurso de una TTP que haga de intermediaria en la transferencia del mensaje. No obstante, con frecuencia se confunde la *autenticación de origen de los datos* con el *no repudio de envío* que, como más adelante se justifica, requiere de mayores salvaguardas. La mayoría de protocolos de correo seguro, entre ellos el conocido PGP al que antes hicimos referencia, proporcionan a sus usuarios estos tres servicios.

SERVICIO DE NO REPUDIO

En una primera aproximación (en el Capítulo 9 se estudia con mayor detenimiento) relacionaremos este servicio con el intercambio de mensajes a través de redes telemáticas para dar garantías respecto a su emisión y recepción. Como su nombre sugiere, sirve para evitar que alguno de los participantes en la comunicación niegue (repudie) haber formado parte de ella. El nombre elegido para este servicio (*Non-repudiation*, en inglés) no es excesivamente afortunado por cuanto está cargado de connotaciones

que nos remiten a un pasado (todavía presente en algunos lugares) poco favorable para los derechos de la mujer. No obstante, este término está ya suficientemente extendido y aceptado, por lo que seguiremos utilizándolo haciendo caso omiso de esas connotaciones. Podríamos distinguir (Figura 1.4) tres situaciones:

a) No Repudio con prueba de origen

En este caso, el receptor del mensaje adquiere una prueba, demostrable ante terceros, del origen de los datos recibidos. En muchos casos pueden bastar las garantías que el emisor introduce en el mensaje cuando aplica los mecanismos que aseguran la autenticación del origen de los datos. En otros casos pueden requerirse evidencias acerca la relación existente entre el autor de un determinado mensaje y la entidad que procede a enviarlo a través de la red. Piénsese, por ejemplo, en que alguien escriba y firme una carta vejatoria hacia otra persona pero que no se atreve a enviársela por correo postal. Si un tercero localiza ese escrito y lo pone en el correo, el receptor tendrá pruebas demostrables de quién es el autor de la carta, pero no de que además de escribirla ha sido él quien ha tomado la decisión ofensiva de enviársela.

b) No Repudio con prueba de envío

El receptor o el emisor del mensaje adquieren una prueba, demostrable ante terceros, de la fecha y hora en que el mensaje fue enviado. Este servicio trata de emular al que frecuentemente presta el Servicio Postal cuando al entregar una carta certificada en la estafeta se solicita que una copia del documento que se quiere enviar sea sellada con una marca de tiempo precisa, de tal manera que pueda servir posteriormente como prueba ante determinados actos administrativos que requieren que, por ejemplo, una solicitud sea entregada antes de una fecha concreta bien en el registro de entrada de la oficina de destino o bien en una dependencia de Correos. (Dependiendo de qué entidad reciba la prueba, se puede desdoblar en dos este servicio.)

c) No Repudio con prueba de entrega

El emisor del mensaje adquiere una prueba, demostrable ante terceros, de que los datos han sido entregados al receptor adecuado. Continuando con el ejemplo del correo, este servicio equivaldría al envío de cartas con acuse de recibo, en cuyo caso el Servicio de Correos devuelve, convenientemente cumplimentada, una cartulina que sirve de evidencia y justificación de que la carta ha llegado a su destinatario.

En estos dos últimos ejemplos, Correos se comporta como una entidad intermedia entre el enviante y el receptor, lo que traducido al mundo de las redes significa que para implantar ese servicio es necesaria alguna entidad que actúe como Tercera Parte de Confianza (TTP). Esto equivale a decir que para que este servicio pueda ser provisto es necesario que exista, en alguna medida, una *infraestructura de seguridad* que haga de garante y proporcione las evidencias exigidas.

Esta es la causa por la que el *servicio de No Repudio* es más difícil de implantar que la terna *Autenticación-Confidencialidad-Integridad* que, como antes dijimos, pueden implementarse (en su versión más simplista) solamente en la entidad emisora y en la entidad receptora sin el concurso de ninguna TTP intermediaria. No obstante esta mayor dificultad, no es necesario poner mucho énfasis para trasladar la convicción de lo imprescindible que resulta la implantación de servicios de *No Repudio* si lo que se pretende es que dentro de la Sociedad de la Información, la mayoría de las comuni-

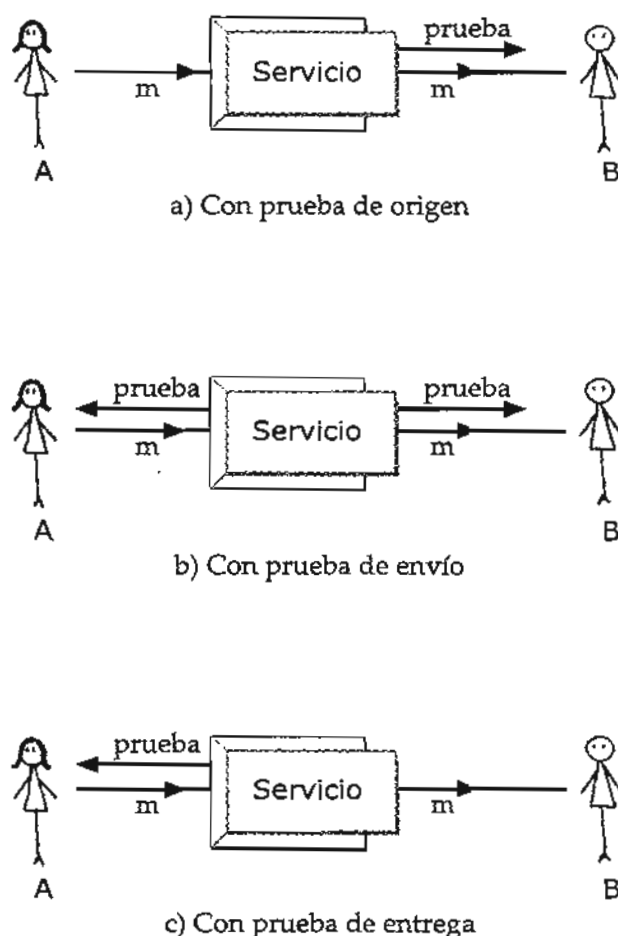


Figura 1.4. Tres casos de No Repudio.

caciones que convencionalmente han venido desenvolviéndose mediante el intercambio de documentos en papel se vean sustituidas por transferencias de documentos digitales a través de redes telemáticas. No es concebible la implantación de operaciones con las Administraciones Públicas a través de la red sin que este tipo de servicios esté operativo.

SERVICIO DE CONTROL DE ACCESO

Este servicio (*Access Control* en inglés) sirve para evitar el uso no autorizado de los recursos de la red. Puede servir para permitir que sólo quien esté autorizado para ello pueda conectarse a una determinada máquina, y para que, una vez conectado, cada usuario sólo pueda tener acceso a aquellas facilidades para las que ha adquirido permisos. De forma resumida, podríamos decir que este servicio permite especificar *quién puede hacer qué*, es decir, qué usuarios pueden hacer determinadas operaciones.

En realidad, este servicio no es privativo de las redes telemáticas, sino que ya estaba implantado en los accesos a ordenadores aislados instalados en los centros de cálculo, en los que en función del «login» se permitía acceder a determinados procesadores o programas. Por ejemplo, el compilador de C podría ser un recurso compartido por todos, pero no así el resultado de la compilación de un programa concreto. Es decir, aparece siempre que se presente un escenario en el que distintos usuarios con distintos privilegios accedan concurrentemente a un mismo recurso.

En las redes telemáticas se puede presentar la necesidad de disponer de un servicio de *control de acceso* en dos situaciones distintas. Éstas son:

- a) El acceso remoto a servidores de todo tipo como bases de datos, impresoras, servidores de correo, etc. El usuario accede regularmente bajo una arquitectura cliente-servidor y, tras identificarse, los mecanismos en que se apoya el servicio determinan a qué partes del servicio se les concede acceso.
- b) El acceso a los terminales desde los que el usuario se conecta a la red. Ellos pueden incluir desde ordenadores protegidos a tarjetas inteligentes. Con frecuencia son estos puntos externos de las redes los que necesitan ser protegidos de forma más estricta, mientras que en otros casos se permite el acceso desde cualquier terminal y los controles se centran solamente en el servidor accedido.

En la mayoría de los casos este servicio se implementa íntimamente ligado a la provisión previa de un servicio de autenticación. Una vez que el usuario ha demostrado que es quien dice ser se le aplican las restricciones o permisos correspondientes.

No obstante, se dan otras circunstancias en las que el control de acceso se lleva a cabo mediante *credenciales*, esto es, piezas de información que representan una serie de privilegios a quien las porte, independientemente de su identidad. Un ejemplo en el mundo real (esto no quiere decir que el mundo de las Comunicaciones Mediante Ordenador no lo sea) puede ser un boleto de entrada para acceder a una butaca concreta de una sala de cine: el privilegio se adquiere al comprarla y su aplicación efectiva no depende de la identidad del portador.

En cuanto al grado de implantación de este servicio, cabe decir que, desafortunadamente, con demasiada frecuencia el control se limita a decir sí o no al intento de acceso al recurso, sin mayores detalles posteriores. Cuando se trata, por ejemplo, del acceso a una base de datos plural y compleja, una adecuada implementación debería ofrecer un acceso muy pormenorizado, regulando categorías de usuarios y categorías de actuaciones (no puede ser lo mismo tener derecho a leer la información que tener derecho a insertar o modificar datos).

También con demasiada frecuencia el control de acceso se aplica a través de mecanismos de autenticación muy débiles, como puede ser una palabra de paso (*password*) o un PIN (número de identificación personal). En otros casos se emplean mecanismos criptográficos robustos que aportan la necesaria seguridad en la protección del acceso. Las *tarjetas inteligentes*, que estudiaremos en el próximo capítulo, representan un componente de seguridad importantísimo de cara a la implantación tanto del servicio de *Control de Acceso* como de los restantes servicios de seguridad.

SERVICIO DE ANONIMATO

Se trata de conseguir que la identidad de la persona que realiza una determinada operación telemática permanezca oculta ante algunos de los actores presentes en esa operación. Se trata de emular en la Red situaciones de la vida real (si se puede seguir haciendo esta distinción por mucho tiempo) en las cuales es conveniente mantener cierto anonimato. Si dentro del correo postal es posible enviar cartas de forma anónima, también el correo electrónico debe permitir esa posibilidad.

Veamos también otros casos en los que este tipo de requisitos se presentan. En primer lugar, supongamos, por ejemplo, un buzón de sugerencias o de quejas dispues-

to para que éstas sean depositadas en él de forma anónima. Otro caso parecido sería la realización de encuestas anónimas a un grupo de alumnos. Si estas tareas queremos llevarlas a cabo por vía telemática (ello sería cómodo, ágil y eficaz) sería necesario que el agente telemático que recibiese los mensajes no fuese capaz de relacionar las opiniones vertidas con los autores de las mismas. En el primero de los dos ejemplos podría permitirse que fuese cualquier persona quien depositase sus quejas en el buzón, y que formulase más de una. Pero en el segundo de ellos sería necesario autenticar primero al alumno (sólo un grupo de ellos puede opinar) y garantizar después que no se conocerá cuál ha sido su opinión.

Esta misma complejidad (tener que combinar la autenticación de los actores y el mantenimiento del anonimato en relación con los datos que han sido depositados) se produce también en el *voto telemático* [CGMP02] en su acepción de esquema evolucionado del *voto electrónico*. En la proyección telemática del escenario de votación convencional habrá que garantizar que solamente depositan su voto en la «urna» las personas autorizadas para ello y que sólo votan una vez y por una sola opción. Por supuesto, debe permanecer oculto quiénes fueron los que votaron cada una de las opciones. Además, el votante debe disponer de mecanismos que le permitan comprobar que su voto ha sido contabilizado adecuadamente en la opción que eligió.

Otro caso interesante es el del dinero digital. Es latoso tener que llevar dinero en los bolsillos, pero tiene una enorme ventaja: podemos comprar de forma anónima. Por ello, el dinero digital mediante el que se compre a través de la red deberá ser también anónimo, cosa que no ocurre en la actualidad en las operaciones de *Comercio Electrónico*. El uso de tarjetas de crédito convencionales es cómodo y eficaz, pero permite que se creen registros en los que se relacione qué compramos, cuándo y dónde. Sería necesario que el uso de las tarjetas de crédito estuviera dotado no sólo de confidencialidad, sino además protegido con servicios de anonimato, de forma que el banco sepa cuánto dinero se ha gastado, pero no dónde ni en qué, y el vendedor tenga certeza de que cobra el importe, pero no tenga capacidad para saber de quién. Este tipo de comportamiento sería el que se correspondería con un tipo especial de tarjetas inteligentes que operasen bajo una infraestructura telemática: las *Tarjetas de Crédito Anónimas* [Carr02].

Por lo general, para proveer el servicio de anonimato en un escenario de relativa complejidad se necesitan Agentes Telemáticos especializados (TTPs) además de las TTPs necesarias para soportar una infraestructura de seguridad convencional. Será necesario, también, utilizar mecanismos criptográficos avanzados, algo más complejos que los que se refieren de forma resumida en el siguiente epígrafe.

1.4. SISTEMAS CRIPTOGRÁFICOS E INFRAESTRUCTURAS DE SEGURIDAD

La criptografía ha sido usada desde muy antiguo, aunque modernamente, con el concurso de los computadores, resulta de más fácil uso y mucho más eficaz que la que se conseguía con los sistemas tradicionales. En la Figura 1.5 aparece el esquema genérico de un sistema criptográfico, en el que puede verse cómo un subsistema produce un mensaje cifrado a partir de un mensaje original, mediante la aplicación de una transformación de éste que depende de un parámetro denominado *clave de cifrado*. En el extremo opuesto, el mensaje original puede ser recuperado en otro subsistema mediante la aplicación de la *clave de descifrado*. La gracia está en encontrar los algoritmos adecuados para el proceso de cifrado y descifrado, así como para el ade-

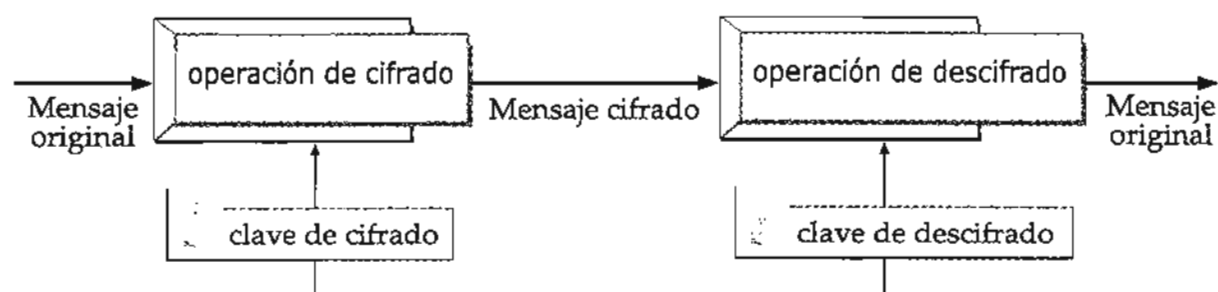


Figura 1.5. Cifrado de un mensaje.

cuando transporte y compartición de las claves por parte del usuario que emite el mensaje original y el usuario que recibe éste en el extremo opuesto.

TERMINOLOGÍA

A continuación se presenta una lista de términos relacionados con la criptografía. En algunos casos se ha añadido el correspondiente nombre en inglés para facilitar el posterior acceso a normativa y documentación especializada (que habitualmente se encuentran escritas en este idioma).

Criptografía (*Cryptography*)

Arte y ciencia de conservar (mantener) los mensajes seguros (ocultos). Etimológicamente viene del griego *kriptos*, que significa oculto.

Criptoanálisis (*cryptoanalysis*)

Arte y ciencia de romper mensajes cifrados, descubriendo la clave, el mensaje en claro o ambos.

Criptología

Rama de las matemáticas que abarca la criptografía y el criptoanálisis. Por lo general, se usa el término *criptografía* con esta misma acepción globalizadora.

Criptógrafo (*cryptographer*)

Persona que se dedica al estudio y práctica de la criptografía. Se usa también (muy raramente) para denominar a una máquina o sistema que sirve para cifrar.

Criptoanalista

Persona que se dedica al criptoanálisis.

Mensaje (o texto) en claro (*plaintext, cleartext*)

Mensaje o documento original que va a ser objeto de cifrado.

Mensaje (o texto) cifrado, criptograma (*ciphertext*)

Mensaje o documento cifrado.

Cifrado, cifrar (*encipherment, encipher*)

Operación de transformación de un *mensaje en claro* en un *mensaje cifrado*. Es la nomenclatura propuesta por ISO para esta funcionalidad.

Descifrado, descifrar (*decipherment, decipher*)

Operación de transformación de un *mensaje cifrado* en un *mensaje en claro*. Es la nomenclatura propuesta por ISO para esta funcionalidad.

Algoritmo criptográfico

De forma similar a cualquier otro algoritmo computacional, consiste en un conjunto de reglas que definen de forma precisa y no ambigua una secuencia de operaciones y transformaciones de los datos tal que dicha secuencia termine en un tiempo finito. Dichas operaciones dependerán de la clave utilizada en cada momento. Un algoritmo debe contemplar la operación de cifrado y la operación de descifrado.

Criptosistema

Un criptosistema utiliza un algoritmo criptográfico para cifrar y descifrar los mensajes. Se trata, por tanto, de un sistema que implementa un algoritmo.

Encriptado, encriptar (*encryption, encrypt*)

Equivale a *cifrar*. En inglés es de uso corriente y preferente. Algunos autores desechan su uso en español por entender que tiene connotaciones fúnebres. Aquí será usado poco frecuentemente.

Desencriptado, desencriptar (*decryption, decrypt*)

Equivale a *descifrar*. Recibe igualmente el mismo rechazo que la anterior. También es usada por algunos autores bajo la acepción **descriptado, descriptar**.

Codificar y decodificar (o decodificar)

Podría entenderse (y así se emplea por algunos autores) como una técnica de cifrado por medio de códigos. No obstante, como tanto la Telecomunicación como la Computación, desde sus orígenes, están fuertemente ligadas a la codificación para posibilitar la representación y la transferencia de la información, sin intención alguna de ocultación, desaconsejamos la utilización de este término dentro de la criptografía, y con mayor motivo cuando esté orientada a su aplicación en redes telemáticas.

CRIPTOGRAFÍA MODERNA. CRIPTOSISTEMAS DE CLAVE SECRETA

Puede decirse que la Criptografía es tan antigua como la Historia: desde que aparece la habilidad de comunicarse mediante mensajes escritos aparece la habilidad de encriptar los mensajes para que sólo las personas que estén en conocimiento de determinado secreto sean capaces de entenderlo. Tradicionalmente, la Criptografía ha estado ligada al mundo de la guerra y de la diplomacia, lo que llevaba consigo su

consideración como secreto de Estado y su exclusión en las relaciones personales y comerciales de los ciudadanos. Esta manera de ocultar los mensajes, que podríamos denominar *criptografía clásica*, obedece genéricamente a un esquema similar al representado en la Figura 1.5 y tiene una característica distintiva: tanto las claves como el algoritmo son secretos.

La que podemos considerar como *criptografía moderna* asienta sus bases teóricas en siglo XIX, pero su verdadera aparición hay que situarla en la década de los setenta del pasado siglo, cuando aparecen algoritmos disponibles para usos civiles implementables mediante un programa de ordenador. La característica fundamental de esta criptografía moderna es que el algoritmo de cifrado debe ser de conocimiento público, y sólo la clave debe permanecer secreta. La seguridad del sistema reside en la robustez del algoritmo y en la adecuada custodia de la clave para que sólo sea conocida por las personas autorizadas para ello.

Dentro de la criptografía moderna pueden distinguirse dos tipos de sistemas criptográficos:

- Criptosistemas de clave secreta.* En ellos, la clave de cifrado y de descifrado es la misma: es una *clave secreta* que comparten el emisor y el receptor del mensaje. Debido a esta característica son denominados también *criptosistemas simétricos*.
- Criptosistemas de clave pública.* Se distinguen porque cada usuario o sistema final dispone de dos claves: una *privada*, que debe mantener *secreta*, y una *pública*, que debe ser conocida por todas las restantes entidades que van a comunicar con ella. Se los conoce también como *criptosistemas asimétricos*.

El Capítulo 4 está dedicado por entero al estudio de la criptografía de clave secreta. En este apartado se adelanta, de forma resumida, una descripción de sus características más importantes con objeto de adquirir una panorámica general de los conceptos más significativos de la Seguridad en Redes.

La Figura 1.6 refleja el esquema básico de un criptosistema simétrico. El proceso de obtención del mensaje cifrado c es:

$$c = E_k(m)$$

donde k es la clave secreta, E_k representa la operación de cifrado con esa clave y m es el mensaje en claro.

Simétricamente, el proceso de recuperación del mensaje original, mediante la operación de descifrado, D_k , en el que se utiliza la misma clave secreta k , se representa como:

$$m = D_k(c)$$

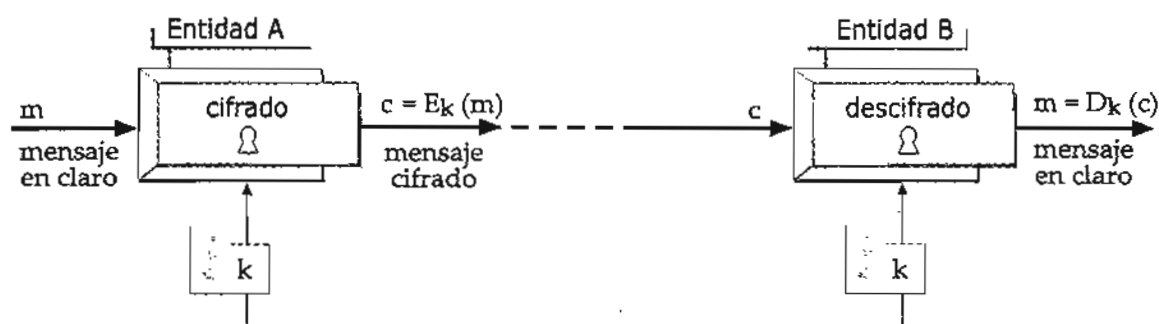


Figura 1.6. Esquema de la criptografía simétrica.

La simetría del proceso se basa, por tanto, en que este tipo de cifradores se comporta tal que:

$$D_k [E_k (m)] = m$$

El primer algoritmo significativo de clave secreta fue el DES. Otros algoritmos importantes son el IDEA y el recientemente aprobado Rijndael. Todos ellos se describen con cierto detalle en el Capítulo 4.

CRITOSISTEMAS DE CLAVE PÚBLICA

En un criptosistema de clave pública, un determinado usuario o sistema final A guarda su clave privada kS_A que solamente es conocida por él. Sin embargo, la clave pública kP_A puede ser perfectamente conocida por todos los demás usuarios pertenecientes a un determinado entorno de seguridad. Se parte del supuesto de que es computacionalmente impracticable deducir el valor de la clave privada a partir del conocimiento de la clave pública. El algoritmo se comporta de manera que si el mensaje se cifra usando una de las claves, se puede descifrar utilizando la otra. El Capítulo 5 se dedica por entero a los criptosistemas de clave pública, adelantándose aquí algunos de los conceptos más importantes.

En la Figura 1.7 se recoge el esquema de cifrado con la clave pública del receptor. Si A desea enviar un mensaje encriptado a B, de tal manera que pueda ser leído por B y sólo por B, lo cifrará con la clave pública de B, kP_B , obteniendo el criptograma c:

$$c = E_{kP_B} (m)$$

En el extremo receptor, B, único poseedor de kS_B , podrá recuperar el mensaje original descifrando c con su clave privada:

$$m = D_{kS_B} (c)$$

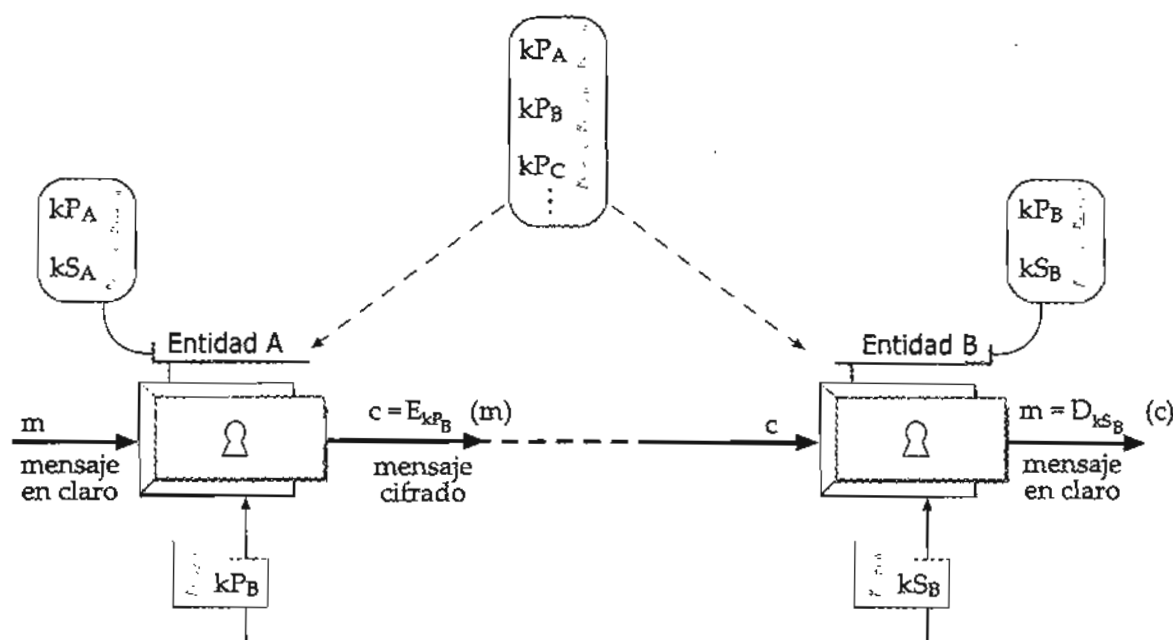


Figura 1.7. Cifrado con la clave pública de la entidad receptora.

De forma equivalente a la operación de cifrado y descifrado que podría haberse llevado a cabo mediante un criptosistema simétrico, en la operación antes descrita se proporciona el *servicio de confidencialidad*.

Por otra parte, algunos algoritmos de clave pública, como es el caso de RSA (que se describe con detalle en el Capítulo 5), tienen la propiedad de que cualquiera de las dos claves, pública y privada, puede ser usada indistintamente para cifrar, usando la otra para el descifrado. Si A cifra el mensaje con su propia clave privada (según se recoge en la Figura 1.8) se obtiene:

$$c = E_{k_{SA}}(m)$$

En el extremo opuesto, B, que conoce la clave pública de A, podrá recuperar el mensaje en claro mediante la operación:

$$m = D_{k_{PA}}(c)$$

Es evidente que otros usuarios o sistemas terminales (C, D, etc.) conectados a la red, también podrán realizar esta misma operación, ya que todos conocen k_{PA} . En este caso, por tanto, no puede hablarse de «encriptado» u ocultación, es decir, no proporciona el servicio de confidencialidad, debido a lo cual, en la comunicación de A con B podría añadirse el mensaje m para facilitarle las cosas al receptor sin ninguna merma de seguridad. Lo que sí se proporciona son los servicios de *integridad* y de *autenticación del origen de los datos*, porque los receptores pueden tener confianza en que el mensaje ha sido cifrado por A, que sólo A es capaz de haberlo cifrado y que los datos recibidos no han sufrido añadidos ni modificaciones de cualquier tipo.

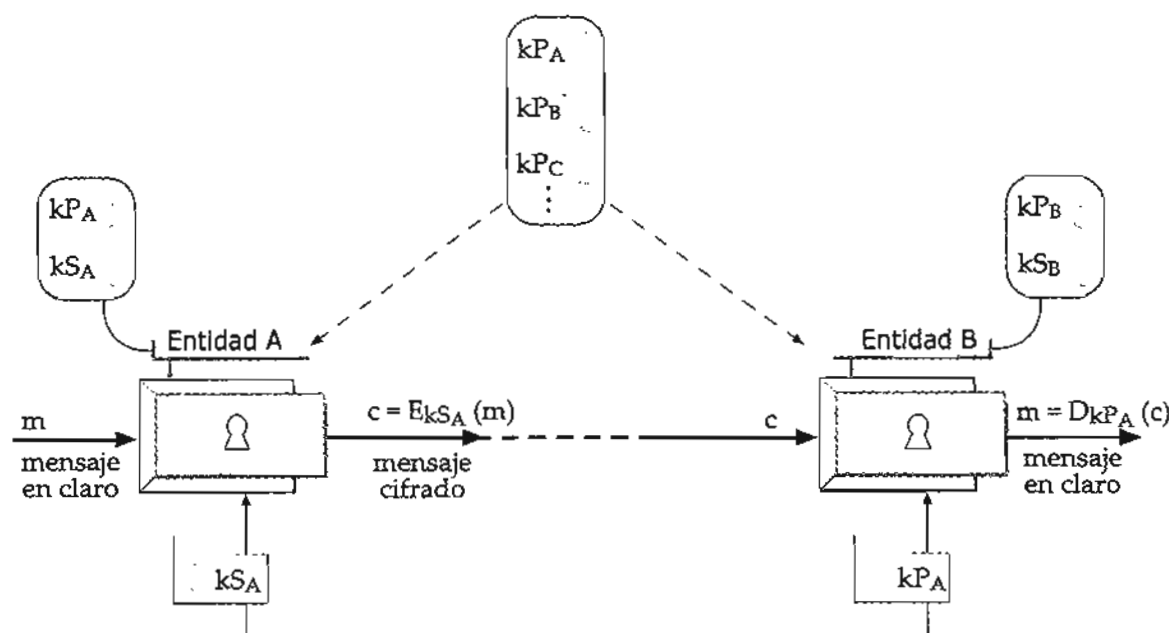


Figura 1.8. Cifrado con la clave privada de la entidad emisora.

FIRMA DIGITAL, CERTIFICADOS Y PKIs

Debido a estas características del cifrado con la clave privada, esta operación o mecanismo criptográfico recibe a veces el nombre de *firma*. No obstante, a lo que propiamente se llama *firma digital* es a un mecanismo criptográfico donde lo que realmente se cifra con la clave privada del signatario o firmante es una muestra reducida,

un resumen o *hash* de tamaño fijo (120 o 160 bits, según el algoritmo) e independiente del tamaño del mensaje a firmar. El receptor del mensaje firmado, mediante unas operaciones de comparación, puede adquirir suficientes garantías acerca de quién es el autor de la firma y sobre la integridad del mensaje recibido.

La firma digital es un mecanismo de seguridad de gran importancia porque cumple, en el mundo de las CMC, la misma función que la firma caligráfica en las comunicaciones convencionales que utilizan el papel como soporte (aunque de forma muchísimo más fiable y segura en «nuestro» caso). Debido a esta importancia, el tema de la firma digital será analizado primeramente en el Capítulo 5 y posteriormente, con mayor nivel de detalle, en el Capítulo 8, dedicado a los servicios de autenticación.

Para la provisión del servicio de anonimato son de utilidad todos los mecanismos criptográficos antes referidos, además de otros específicos entre los que cabe destacar, por más llamativo, el mecanismo de *firma opaca* o *firma a ciegas* (*blind signature*). Un equivalente de la *firma opaca* «en el mundo del papel» podría consistir en que A prepara un documento para que sea firmado por B sin que B vea lo que está firmando. Para ello lo introduce en un sobre junto con un trozo de papel carbón, cerrando después el sobre. B firma en el sobre y la firma se estampa en el documento. A recupera el sobre, lo abre, tira el papel carbón y obtiene el documento firmado por B. De forma similar, el mecanismo criptográfico de firma a ciegas se basa en la multiplicación del mensaje por un *factor de opacidad* (un número) que sólo A conoce y que puede ser retirado por A una vez que haya obtenido la firma de B. Naturalmente, en los escenarios de seguridad donde se implante este tipo de firma se han de proporcionar las necesarias salvaguardas para que B se decida a firmar sin conocer el contenido de lo que firma. En el Capítulo 11, dedicado al servicio de anonimato, se describe, entre otros, este mecanismo.

Cuando, en apartados precedentes, se describieron los servicios de seguridad, se hizo varias veces referencia a la conveniencia de que existan **TTPs** (*Trusted Third Parties*), agentes telemáticos especializados que intervienen en las comunicaciones seguras. En los escenarios de comunicaciones seguras que será necesario implantar para satisfacer las necesidades de los ciudadanos deben existir múltiples TTPs de funcionalidades diversas, constituyendo la que denominamos una **Infraestructura de Seguridad**, entramado de agentes telemáticos que resulta imprescindible para la provisión de servicios avanzados de seguridad. Además de la adecuada definición técnica de las funciones que cumplen, será fundamental e imprescindible disponer de una reglamentación jurídica que sirva para proteger la privacidad de los ciudadanos, de forma que determine las responsabilidades con que deben pechar tanto sus gestores como quienes utilicen los servicios.

El **certificado digital** es una pieza de información, un documento de capital importancia, en el que se asocia el nombre de una entidad con su clave pública (pareja de la clave privada correspondiente) durante un determinado periodo de validez. El certificado es emitido por una TTP especial denominada **Autoridad de Certificación, CA**. Puede afirmarse que no puede existir un escenario de seguridad robusto donde no existan una o varias CAs. En la normativa legal vigente acerca de la validez de la firma digital, tanto en España como en otros países, se dedica especial y pormenorizada atención a fijar las exigencias y los requisitos que deben cumplir tanto los certificados como las CAs que los emiten. A la infraestructura de seguridad necesaria para la provisión y gestión de certificados suele denominársele *Infraestructura de Clave Pública, PKIs* (de su nombre en inglés, *Public Key Infrastructure*). En el siguiente capítulo se comentará de nuevo resumidamente el tema de las TTPs, las CAs y los certificados. En el Capítulo 6 se describe pormenorizadamente la estructura del certificado y el Capítulo 7 está especialmente dedicado a las infraestructuras de seguridad y PKIs.

USO COMBINADO DE LA CRIPTOGRAFÍA SIMÉTRICA Y ASIMÉTRICA

En los criptosistemas de clave secreta, la transformación de los datos se apoya en operaciones lógicas o algebraicas muy simples cuyo cometido es alterar el orden de los bits que aparecen en el mensaje original y sustituir sus valores por otros conforme a las reglas del algoritmo de cifrado. Por el contrario, los algoritmos de cifrado de clave pública se basan en operaciones matemáticas de cierta complejidad. Por esta razón, los sistemas simétricos son más rápidos y eficaces que los asimétricos, pero éstos son más seguros.

Por otra parte, si se utiliza solamente criptografía simétrica, si el número de entidades que participan en la comunicación es relativamente grande, y se desea mantener comunicaciones seguras dos a dos, el número de claves secretas que es necesario mantener convierte en impracticable un escenario de comunicación de este tipo. Por contra, si lo que se utilizase fuese criptografía de clave pública, solamente sería necesario mantener un par de claves por cada usuario, con lo que la gestión de claves es mucho más sencilla y robusta.

Por todo ello, para la provisión de los servicios de seguridad, las técnicas actuales se basan en el uso combinado de ambos sistemas. Es frecuente usar un sistema de clave pública para intercambiar una clave secreta simétrica que será utilizada en una transacción concreta, de tal forma que esta clave secreta sólo sirve para esa transferencia, siendo desechada una vez que ésta termine, reduciendo con ello el riesgo de interceptación. En el Capítulo 5, una vez estudiados los mecanismos basados en criptografía de clave pública, se describe con detalle este mecanismo.

1.5. COMUNICACIONES SEGURAS: HACIA UNA SEGURIDAD CÍVICA

Mediante la incorporación de servicios de seguridad en las redes telemáticas se ha de conseguir, al menos, garantizar que los derechos y salvaguardas actualmente reconocidos en las comunicaciones convencionales sean respetados también en la proyección y plasmación que éstos tienen en las CMC. Pero hay más: siendo esto necesario, no es suficiente: mediante una adecuada utilización de los servicios de seguridad, los ciudadanos podrán disponer de herramientas que les permitan alcanzar niveles de independencia y privacidad como jamás antes en la Historia se les habían presentado. Al entorno configurado conforme a esta perspectiva es a lo que denotamos como *Seguridad Cívica*, acepción que ampliaremos más adelante.

REQUISITOS DE LOS USUARIOS, ANÁLISIS DE RIESGOS Y POLÍTICAS DE SEGURIDAD

Lo primero que hay que pensar cuando se intente desarrollar un sistema telemático o securizar un determinado entorno es conocer los **requisitos** y las **necesidades** de los usuarios a quienes va a beneficiar ese sistema. No merece la pena desarrollar un sistema telemático técnicamente perfecto que incluya innovaciones notables y use las más avanzadas técnicas, si el entorno social al que va dirigido, es decir, los ciudadanos para los cuales ha sido concebido, no confían en él o no responde a sus necesidades reales. Será necesario, para un diseño global adecuado, tener en cuenta los requisitos tanto técnicos como sociales [Carr01]. Por tanto, para abordar el desarrollo de aplica-

ciones telemáticas complejas, dotadas de servicios de seguridad, es imprescindible que al mismo tiempo que se realizan los trabajos de ingeniería requeridos, se hagan, mediante una cooperación multidisciplinar, los análisis sociológicos correspondientes para determinar la utilidad y la viabilidad de los sistemas. Evidentemente, si lo que es necesario hacer es muy sencillo y está claro desde un principio, no será necesaria una iniciativa de este tipo.

Por otra parte, cuando se aborda el desarrollo de un nuevo sistema telemático seguro o se pretende securizar un determinado escenario de comunicación ya existente, será necesario averiguar cuáles son los riesgos de vulneración a los que está sometida la información que allí se transfiere o se almacena. Este tipo de tareas de **análisis de riesgos**, cuando el sistema es complejo, pueden resultar muy laboriosas y necesitar bastante tiempo y presupuesto. Para llevarlas a cabo se han desarrollado **metodologías** específicas, apoyándose en las cuales, quienes realizan estos trabajos, de forma sistemática y ordenada, van analizando todos los recursos del sistema (equipos, programas, terminales, protocolos de comunicación, operarios, etc.). Debido a su complejidad, puede considerarse una especialidad dentro del mundo de la securización de sistemas. Nuevamente hay que decir que cuando los sistemas son muy simples los riesgos y los ataques contra los que hay que protegerse se pueden detectar, por sentido común, mediante una elemental evaluación.

Es decir, que para establecer comunicaciones seguras dentro de un determinado entorno o dominio de seguridad, será necesario realizar, de forma metódica, un *análisis de los riesgos* allí existentes y detectar los *ataques* contra los que es menester defenderse, de lo cual se deducirán los *servicios de seguridad* que son necesarios en cada escenario de comunicación. Para proveer los servicios será necesario que alguien implante los correspondientes *protocolos de seguridad*, lo que conlleva el uso de determinados *mecanismos de seguridad* y *algoritmos criptográficos*, además del concurso de las varias y distintas *TTPs* que constituyan la *Infraestructura de seguridad* (o solamente la *PKI*) en la que se inscribe el dominio de seguridad.

Como componente globalizador de todo esto está la **política de seguridad** que se defina en ese entorno por quienes hayan sido comisionados para ello. En la especificación de la política de seguridad deben tenerse en cuenta todos estos elementos y decidir cuáles de ellos serán usados en cada caso, determinando, consecuentemente, qué tipos de claves han de ser utilizadas, cómo han de ser distribuidas y, en general, las reglas de comportamiento y las responsabilidades (legales o de carácter interno) de todos los actores que participan en la comunicación. En una primera aproximación, podemos considerar como **dominio de seguridad** al conjunto de usuarios y sistemas que operan bajo una misma *política de seguridad*.

En el siguiente capítulo se tratará de clarificar un poco más todo este asunto.

ESTADOS, CIUDADANOS, CRIMINALES E INGENIEROS

Si se estableciera una comparación entre las comunicaciones telemáticas, de una parte, y de otra el correo postal y las comunicaciones telefónicas, se podría decir que en estos dos últimos casos, los poderes públicos han garantizado desde hace mucho tiempo la confidencialidad de las comunicaciones en tanto y cuanto no exista una orden judicial que permita dejar de respetar ese derecho (otra cosa es la imagen que los ciudadanos tienen de cómo y con qué escrúpulo han respetado los poderes públicos y policiales, tanto los legítimos como los dictatoriales, esos derechos). Por contra, en el caso de mensajes cifrados con una clave secreta o con la clave pública del receptor,

la confidencialidad se mantiene en cualquier caso, no siendo posible sortearla (desde fuera del sistema) ni por orden judicial ni por otro motivo, a menos que se capture la clave de cifrado o que se consiga romper la seguridad con métodos criptoanalíticos.

Hay quienes piensan que con esta dificultad de actuación por parte de los poderes públicos se estaría dando pábulo a que criminales y personas cuyo objetivo sea romper la convivencia social puedan usar impunemente las comunicaciones telemáticas seguras. Consecuentemente, para defender los derechos y libertades colectivas sería necesario poner cortapisas a un esquema de comunicación de este tipo.

Bajo este criterio, en 1993, la presidencia de los Estados Unidos de Norteamérica anunció la iniciativa *Key-Escrow*, más popularmente conocida como *debate Clipper Chip*. La esencia de la iniciativa consiste en dotar a todos los equipos comunicantes (desde teléfonos a ordenadores) de un circuito integrado protegido contra manipulación (*tamper-proof*) que opera bajo un algoritmo y una clave de cifrado. Una copia de cada una de estas claves individuales se divide en dos componentes, de suerte que cada una de las partes es almacenada en dos agencias gubernamentales distintas. Una tercera agencia puede reunir ambas piezas por orden judicial (o policial) y proceder a interceptar el tráfico generado por la máquina en que está alojado el chip (en el Capítulo 6, al hablar de la distribución de claves, se describe con cierto detalle este sistema).

En este debate participaron multitud de personas procedentes tanto del campo de la Técnica como de la Política y la Sociología. Parece que en este primer envite la cosa no ha prosperado, aunque en un futuro puede que vuelva a aparecer bajo esta o bajo otra modificada forma (parece que soplan vientos proclives a estas tesis «defensivas»). Si esto se impone, ¿cabe esperar que este tipo de salvaguardas van a representar una barrera insalvable para los grupos organizados que quieran actuar fuera de la ley? Antes bien, cabe pensar que cualquier organización mínimamente competente dentro de la Mafia, del narcotráfico o del terrorismo será capaz de establecer vías seguras y ocultas que se escapen a las limitaciones del Key-Escrow.

Los que sí estarían bajo vigilancia serían los ciudadanos corrientes (no me atrevo a decir «y decentes»), que son los que usan equipos adquiridos de forma convencional y acceden a través de redes de dominio público. Es decir, que sería peor el remedio que la enfermedad.

Para ilustrar lo sencillo que resulta buscar vías secretas de comunicación, baste aquí una primera y resumida referencia a una técnica de ocultación de mensajes fácil de usar y difícil de descubrir: la **Esteganografía**, a la que se dedica un epígrafe en el Capítulo 4. Es probablemente tan antigua como la criptografía clásica y también mantiene en secreto el método de cómo obtener el mensaje escondido. Antiguamente se usaban tintas invisibles, gráficos camuflados y otras zarandajas, pero en su versión electrónica moderna se apoya principalmente en introducir ligeras variantes en una imagen digitalizada. Si se sustituyen algunos bits de algunos *píxeles* por los bits de un mensaje secreto, de forma que la imagen conserve una buena calidad de representación, sólo el receptor que conozca el truco de dónde buscar el mensaje puede, computacionalmente hablando, descubrir el mensaje secreto. Además, el mensaje secreto puede ser, a su vez, un mensaje cifrado con un algoritmo criptográfico (público o secreto).

Es evidente que un esquema como el que acabamos describir se escaparía de los controles globales que pudiese establecer el *Clipper Chip* u otro ingenio equivalente. Por todo ello, es nuestro criterio que **los ingenieros** que se dediquen a la implantación de servicios de seguridad en las redes civiles deberán estar más preocupados en conseguir que los derechos y libertades de los ciudadanos sean respetados en las CMC que en el hecho de cazar malhechores. Además, en línea con la cooperación multidis-

ciplinar a que aludíamos más arriba, resultaría muy conveniente que esos ingenieros telemáticos, cuando la complejidad y la repercusión del sistema que desarrollan lo requiera, trabajen de forma conjunta con gente del campo sociopolítico para analizar las necesidades del sistema desde una doble perspectiva.

LA CONFIANZA EN EL SISTEMA

Desde hace muchos siglos se ha venido usando la criptografía para el intercambio de mensajes secretos. Históricamente, los militares, los servicios diplomáticos, los criminales y los amantes han sido los más interesados en este tipo de prácticas. Los sistemas usados en la criptografía clásica eran del tipo que hemos denominado simétricos o de clave secreta. Pero en estos sistemas clásicos no sólo la clave era secreta: también lo era el algoritmo. Es decir, gran parte de la seguridad del sistema se basaba en que eran secretos los procedimientos usados para transformar el mensaje original en un criptograma.

Por el contrario, como ya hemos comentado, el advenimiento de la criptografía moderna y su uso para fines civiles requiere que los algoritmos sean públicos y estén publicados (valga la redundancia) de forma detallada y minuciosa. Por ejemplo, el DES, primer criptosistema de uso generalizado, puede encontrarse claramente especificado en el estándar que lo define, de forma que cualquier usuario puede implementarlo si tiene, cosa muy común, las habilidades necesarias para ello (otra cosa será el problema de las patentes).

Pero además, y esto es lo verdaderamente significativo, los criptoanalistas pueden estudiar la robustez del sistema y hacer públicas sus conclusiones, de forma que los usuarios corrientes puedan tener elementos de juicio acerca de la conveniencia o no de usar tal o cual sistema. El secreto reside, pues, en las claves y en la robustez del algoritmo, no en su ocultación. Por supuesto, los criptosistemas de clave pública o asimétricos, que aparecieron después, cumplen también con estas características de publicidad y seguridad.

Consecuentemente, los ciudadanos y ciudadanas que participen en comunicaciones seguras a través de redes telemáticas, siendo Internet el más amplio escenario (pero no el único) posible para estas prácticas, pueden tener acceso a ciertas informaciones que les ayuden a conocer la fortaleza de los mecanismos usados en un determinado criptosistema. Esta clase de conocimientos, que será lo que determine la confianza de los usuarios en los sistemas, es producto del trabajo de los criptoanalistas.

Otra muy distinta es la confianza imperante cuando se trata de entornos muy cerrados de usuarios en los que existe una fuerte jerarquización de cara a determinar qué sistemas o procedimientos son seguros y cuáles han de ser los que se utilicen. Este tipo de entornos o dominios es el que se corresponde con las comunicaciones militares o de seguridad de los estados y las que puedan establecer los criminales o, incluso, las que puedan interesar a una corporación empresarial cuando se trate de dilucidar internamente acerca de decisiones estratégicas o de inversión. En estos casos, el carácter publicitado de los algoritmos no es, en sí, ningún requisito necesario para determinar su bonanza. Antes al contrario, cuanto menos información tenga un posible atacante, menos posibilidades tendrá de descubrir el mensaje oculto. Además, la decisión acerca de qué tipo de criptosistema es el más adecuado para ser usado, es decir, el que mayor confianza aporte, recae en la cúpula de la estructura jerárquica y piramidal en que está organizada, y no en los usuarios finales.

Básicamente, la inmensa mayoría de los ciudadanos no es en esa clase de seguridad en entornos cerrados en la que están interesados, debido a la naturaleza de la relación abierta que se establece en las comunicaciones telemáticas, en las que no necesariamente se conocen entre sí los individuos participantes. Si unos grandes almacenes, o un banco, nos propusieran utilizar, para comunicarnos con ellos, un algoritmo secreto de su propiedad, del cual nos pasarán para su uso un programa ejecutable, y nos aseguraran que ellos están convencidos de que proporciona, por ejemplo, una muy buena confidencialidad, es muy probable que muchos usuarios recelaran de algo que no tienen posibilidades de contrastar y sobre lo que el único dictamen procede de alguien de confianza más que relativa.

En este contexto, la confianza entre los distintos participantes en comunicaciones seguras abiertas está basada en el conocimiento que ellos tienen acerca de la fortaleza de los algoritmos criptográficos usados, de forma que les sea proporcionado un *razonable* nivel de seguridad. En palabras de Diffie y Landau [DiLa98], «un criptosistema se considera seguro cuando un oponente no lo puede romper bajo razonables circunstancias, en una razonable cantidad de tiempo, a un coste razonable». Aunque el término *razonable* aquí utilizado resulta bastante vago, debemos interpretarlo bajo una óptica economicista, relacionándolo con el tipo de valores o bienes puestos en juego durante el proceso de comunicación.

Por todo ello, para establecer servicios de seguridad en las comunicaciones que interesan a los ciudadanos, en lo que hay que pensar no es en si tal o cual algoritmo ha sido roto excepcionalmente en circunstancias teóricas y excepcionales, sino en la protección que necesitamos conforme al riesgo de vulneración existente (Figura 1.9). Así, en muchas ocasiones, el usar, por ejemplo, un cifrado con un algoritmo que ya está bastante obsoleto, como puede ser el caso del DES, puede resultar más que suficiente para contrarrestar una determinada amenaza. No obstante, siempre que sea

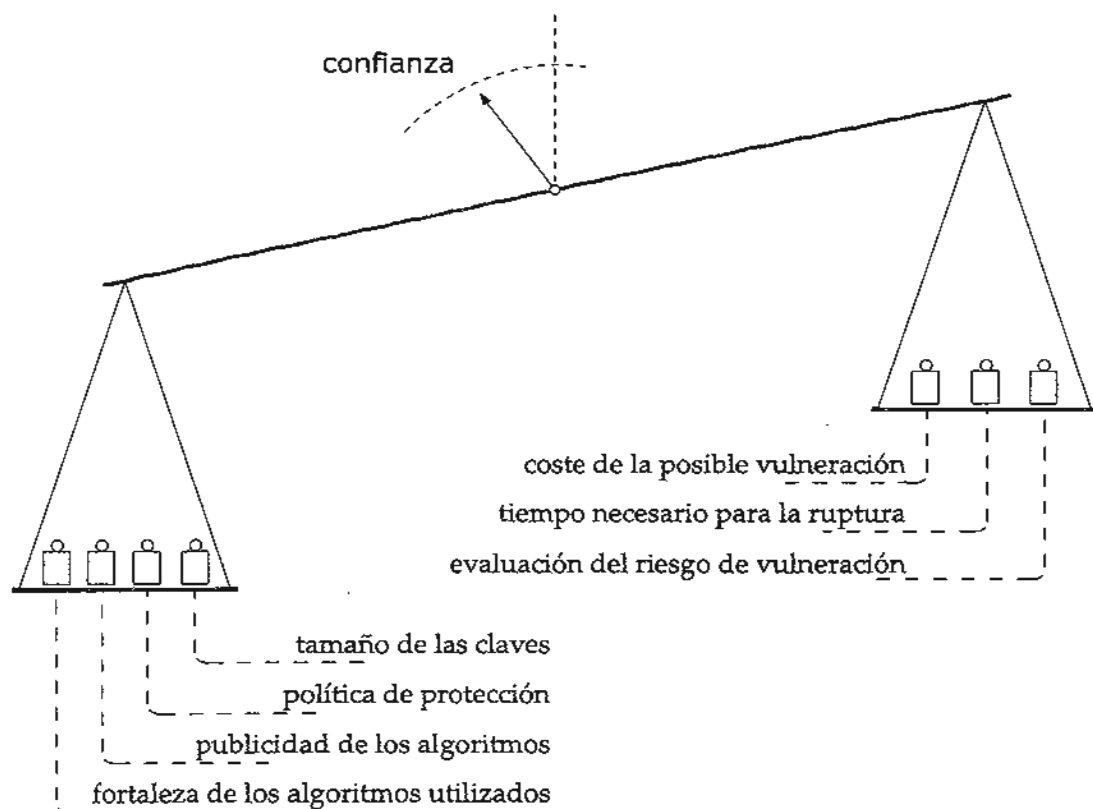


Figura 1.9. Confianza en los sistemas que interesan a los ciudadanos.

posible usar un algoritmo más robusto (que casi siempre será posible), tanto mejor. Lo que sí es bastante cierto es que con los algoritmos y la técnicas que aquí estamos describiendo, los ciudadanos pueden tener un elevado grado de confianza en que los criptosistemas que pueden ponerse a su servicio tienen un más que razonable nivel de seguridad.

Es posible, y bastante probable, que las agencias gubernamentales de inteligencia puedan estar en posesión de potentísimas máquinas y evolucionadísimas técnicas de descifrado y criptoanálisis capaces de violar la seguridad de los modestos mensajes que se intercambian bajo el paraguas de protección de la *seguridad cívica*. Es también posible que sean capaces de romper los códigos de unos pocos durante periodos no muy largos de tiempo, pero de lo que sí podemos estar seguros es de que no son capaces de vigilar a todos durante todo el tiempo. Además, cuanto más gente use los servicios de seguridad, más difícil será para estas agencias la violación masiva de las comunicaciones.

Además, cuando la potencia de cálculo de los ordenadores vaya en aumento y, consecuentemente, disminuya el tiempo que necesite un atacante malicioso para romper la seguridad de un sistema, siempre será posible (dado que también serán más potentes las máquinas de cifrado) aumentar el tamaño de las claves, lo que conlleva un aumento exponencial en la dificultad de su rotura. Es decir, en este cuento del ratón y el gato, siempre gana el ratón. Al menos en los escenarios que aquí nos interesan.

SEGURIDAD CÍVICA

De lo que hemos venido diciendo en los párrafos precedentes se puede deducir la concepción que aquí tenemos de lo que es la **seguridad cívica** en sentido amplio.

Sin pretender hacer una definición rigurosa y precisa, podríamos decir que lo que aquí se entiende por **seguridad cívica** es el conjunto de algoritmos, mecanismos de seguridad, protocolos, servicios, determinación de riesgos y, en suma, políticas de seguridad y métodos de trabajo que son necesarios para establecer comunicaciones seguras sobre redes telemáticas, *teniendo en cuenta las necesidades de la vida diaria de los ciudadanos normales*, permitiéndoles el ejercicio de los derechos cívicos que les son reconocidos en otros ámbitos convencionales de comunicación y mejorando (respecto de esos mismos ámbitos) los niveles de independencia y privacidad actualmente existentes.

Algunos sectores de la población temen que, inevitablemente, la informatización de la mayoría de las actividades de comunicación de los ciudadanos desembocará en una merma de su **privacidad** y de sus derechos. Por contra, cabe afirmar que la implantación de servicios de seguridad bajo la orientación de lo que aquí hemos denominado **seguridad cívica** no sólo garantiza los derechos y libertades ya existentes, sino que permite expandirlos y mejorarlos.

Naturalmente, el nivel de protección o confianza en los sistemas ha de ser proporcional al *bien* o *valor* que se intenta proteger. Muchas ciudadanas y ciudadanos podrían demandar sistemas de seguridad que les permitiesen algunas de estas actividades:

1. Comunicarse con amigos fuera del alcance de miradas indiscretas (básicamente, sólo un servicio de confidencialidad es necesario).
2. Consultar el balance de una cuenta bancaria o un registro personal contenido en la base de datos de una agencia gubernamental (sería necesaria la autenticación de usuario y un adecuado control de acceso).

3. Realizar operaciones bancarias convencionales y ordenar transferencias de fondos en conexión con su oficina bancaria, a salvo de la curiosidad o intromisión de posibles ladrones u otras agencias financieras (el usuario ha de estar seguro de que la otra parte de la comunicación es realmente su banco y no otra entidad maliciosa, por lo que necesita servicios de autenticación mutua, confidencialidad, integridad y no-repudio).
4. Realizar gestiones administrativas con las Administraciones Públicas, obteniendo pruebas, demostrables posteriormente, de que dichos trámites han sido llevados a cabo y la Administración en cuestión no pueda negar su tramitación (los mismos servicios que en el caso anterior serían requeridos).
5. Realizar compras a través de la red obteniendo determinados bienes a cambio de una cantidad de dinero deducible de una cuenta bancaria o a través de otro procedimiento de dinero electrónico (además de confidencialidad, integridad y no-repudio, debería poderse asegurar el anonimato del comprador para evitar que puedan confeccionarse listas que relacionen su nombre con todos los productos que compra y con el dinero que en esas operaciones se gasta, por lo que los procesos de autenticación deberán regularse separadamente para conseguir este objetivo).
6. Participar en los procesos de votación telemática que se implanten y en los procesos de Democracia Digital que se vayan poniendo a disposición de los ciudadanos, no sólo para recibir información de la Administración, sino interviniendo en las decisiones colectivas que se adopten (todos los servicios de seguridad serían necesarios, en uno u otro de los posibles escenarios que se establezcan, para alcanzar este ambicioso objetivo).

A la vista de estos ejemplos, podemos llegar a la conclusión de que el uso coordinado de los servicios de seguridad es el que proporciona, en su conjunto y dependiendo de cada caso, la necesaria privacidad en las operaciones que realizan los ciudadanos a través de redes telemáticas. Naturalmente, para que esta privacidad del ciudadano sea garantizada, es necesario también que tanto los bancos como las oficinas de las Administraciones Públicas a que se ha hecho referencia en los ejemplos anteriores no hagan posteriormente un uso indebido de los registros obtenidos; pero esa exigencia no es privativa solamente para el caso de las redes, sino que aparece en igual medida en las operaciones convencionales.

El término *seguridad cívica* se propone aquí en contraposición con otros requisitos que pueden darse en ámbitos militares, o de espionaje, o de cualquier otro entorno cerrado. En esos entornos, los bienes o valores puestos en juego pueden ser muy elevados (según se mire) y, por supuesto, muy diferentes a los que se manejan en la vida ordinaria. Por ello, aunque en los ejemplos de Seguridad Cívica que antes se han enumerado sólo aparecen actividades que tienen como protagonistas a usuarios finales, también entrarían dentro de esta categoría otras actividades de uso civil, como pueden ser las transacciones que se realicen entre empresas o las comunicaciones que se efectúen entre los distintos departamentos de una de ellas.

Con este tipo de cosas es con las que han de lidiar los ingenieros que, dentro del área de la Telemática, se dediquen al diseño, implementación y puesta a punto de sistemas que garanticen, para usos civiles, la seguridad en las redes telemáticas. Ayudar a conseguir esos conocimientos y esas habilidades es el objetivo perseguido en este y sucesivos capítulos.

CAPÍTULO

2

ESCENARIOS DE COMUNICACIÓN Y UBICACIÓN DE LOS SERVICIOS

El objetivo global que se pretende cubrir con este capítulo es completar la panorámica de seguridad en redes telemáticas iniciada en el Capítulo 1. Si bien en el anterior capítulo esa visión panorámica ha estado centrada en la descripción resumida de los principales conceptos y tecnologías, para concluir aclarando qué tipo de protecciones y qué tipo de seguridad queremos contribuir a alcanzar (la que hemos denominado *seguridad cívica*), en el presente capítulo detendremos nuestra atención en alguno de los componentes o conceptos a los que ya se había hecho referencia allí, pero que consideramos que es conveniente ampliarlos algo más antes de abordar con cierto detalle, en los próximos capítulos, el estudio de los mecanismos y de los servicios de seguridad.

Así, en primer lugar trataremos de aclarar en qué consisten un dominio de seguridad y un escenario de comunicación seguro, así como los pasos que es necesario dar para averiguar cuáles son los riesgos y las amenazas contra los que es necesario prevenirse, para poder determinar posteriormente cuáles son las salvaguardas y protecciones que es necesario introducir en ese entorno de comunicación.

Posteriormente, se comentará el tema de las distintas funcionalidades y configuraciones presentes en las TTPs (terceras partes de confianza) que son componentes casi obligados de cualquier escenario de comunicación seguro, y que son, además, los elementos constitutivos de las *infraestructuras de seguridad*. Finalmente, se comentará la ubicación de los servicios de seguridad, tanto en los niveles de una arquitectura jerarquizada como en la «puerta» de entrada de las redes privadas (los *cortafuegos* o *firewalls*).

2.1. ESCENARIOS Y DOMINIOS DE SEGURIDAD

El término *escenario de comunicación* lo utilizaremos en sentido amplio como un conjunto de entidades comunicantes, las más de las veces en entornos abiertos, en el que existirán terminales u ordenadores a los que acceden directamente los usuarios,

servidores de comunicaciones, redes locales que abarcan entornos geográficos y organizacionales reducidos, redes de área extensa (WANs) que sirven para interconectar redes o usuarios remotos, etc. Es decir, el conjunto de *entidades comunicantes finales* y todas las redes y *agentes telemáticos* específicos que intervienen en una determinada comunicación.

Cuando se trate de las comunicaciones que se establecen para la provisión de un servicio telemático convencional como puede ser, por ejemplo, el correo electrónico, los agentes que estarán presentes, además de los terminales con su correspondiente *interfaz de usuario*, serán los *agentes de usuario* (UAs) y los *agentes de transferencia de mensajes* (MTAs). Otro tanto se podría decir para las restantes aplicaciones y servicios telemáticos.

Cuando se trate de comunicaciones seguras, además de los agentes telemáticos específicos para la comunicación que se trate de securizar, es bastante probable que existan agentes especiales que sirvan para la adecuada prestación de los servicios de seguridad allí provistos. De forma genérica, diremos que estos agentes especiales serán TTPs (terceras partes de confianza) que operan conforme a las distintas funciones de seguridad que les son encomendadas. Estas TTPs, junto con los restantes agentes convencionales, forman parte de los escenarios de comunicación (en este caso, seguros) que se establezcan.

En un determinado entorno, el conjunto de TTPs que ayudan a establecer protecciones de seguridad a todos los servicios telemáticos allí presentes podemos asociarlo con una *infraestructura de seguridad*. Más adelante, en este mismo capítulo, se volverá a insistir sobre esta idea. Lo que sí podemos decir ahora es que, en principio y de forma general, no se puede decir que una infraestructura de seguridad sea un subconjunto de un escenario de comunicación, porque puede darse el caso de que se establezcan escenarios distintos utilizando una misma infraestructura de seguridad, y viceversa, un mismo escenario complejo podría hacer uso de más de una infraestructura.

Consideraremos como un *dominio de seguridad* al conjunto de usuarios y sistemas pertenecientes a un determinado escenario de comunicación que operan bajo una determinada *política de seguridad*. Puede darse el caso de que un determinado usuario pueda poseer, por ejemplo, varios pares de claves distintas (operando bajo un algoritmo asimétrico) certificadas bajo políticas distintas, de tal forma que pueda participar en distintos dominios de seguridad dependiendo, por ejemplo, de cuál sea la aplicación telemática en que esté involucrado.

El término *entorno* o *entorno de seguridad* lo hemos venido utilizando (y lo seguiremos haciendo en lo sucesivo) de forma genérica e imprecisa para hacer referencia a cualquier segmento del ciberespacio en el que esté establecida una determinada relación entre entidades. A veces puede tratarse de un escenario concreto, otras veces puede tratarse de un dominio de seguridad o, simplemente, de la parcela de uno de éstos.

Como ya adelantábamos, en la política de seguridad que se defina deberán tenerse en cuenta todos los elementos que intervienen en la comunicación y las reglas de comportamiento bajo las que han de operar. No menos importante que las definiciones puramente técnicas son las especificaciones concernientes a las responsabilidades que contraen todos los actores que participan en la comunicación. En la utilización de determinadas TTPs que ofertan sus servicios a un amplio número de usuarios, estas responsabilidades (que pueden tener repercusiones penales) se recogen en estipulaciones contractuales que se comprometen a cumplir todos los usuarios que suscriban ese servicio.

2.2. DETERMINACIÓN DE REQUISITOS Y ANÁLISIS DE RIESGOS

Para la determinación de nuevos escenarios de comunicación seguros o para la securización de escenarios de comunicación preexistentes será necesario conocer las necesidades de los usuarios y averiguar los peligros a que esas comunicaciones estarán sometidas.

LOS REQUISITOS DE USUARIO

Cuando se pretenda desarrollar un sistema telemático seguro será necesario conocer, como punto de partida, cuáles son los *requisitos* de los usuarios (entendidos éstos como individualidades o como organizaciones, según cada caso) que van a utilizar el sistema y beneficiarse con los servicios que les preste. Si el sistema es seguro, como estamos presuponiendo, en esos servicios telemáticos estarán también incluidos servicios de seguridad, ya que, como hemos dicho antes, los servicios de seguridad representan, en la mayoría de los casos, un valor añadido sobre los servicios convencionales.

A veces el conocimiento de cuáles son los requisitos de usuario puede obtenerse, por sentido común, mediante un somero análisis de la situación. Sin embargo, cuando se trata de sistemas avanzados que pretenden trasladar al mundo de las CMC escenarios de comunicación con fuertes implicaciones sociales y políticas (como puede ser el caso del voto telemático), la cosa ya no es tan sencilla y requiere de un proceso riguroso para conocer tanto los requisitos técnicos como sociales.

Con frecuencia los potenciales usuarios del sistema a desarrollar tienen una idea aproximada pero bastante clara de las protecciones que «les gustaría» que fuese capaz de llevar a cabo el sistema que se les propone desarrollar, lo que ocurre es que en muchos casos no son capaces de manifestarlas con una terminología rigurosa en términos de seguridad en redes. Por ello será necesario la utilización de técnicas y métodos sociológicos para averiguar cuáles son los requisitos «ideales» que demandan los ciudadanos. En otros casos, esta situación no se produce: los potenciales usuarios conocen y son capaces de enunciar con precisión cuáles son las protecciones de seguridad que necesitan, sin necesidad de ayudas complementarias.

Cuando de lo que se trata no es de desarrollar un sistema seguro de nueva planta, sino de proteger un determinado escenario de comunicación previamente establecido, en la mayoría de las situaciones que se presenten resultará mucho más fácil conocer cuáles son los requisitos de usuario en lo que se refiere a la nueva situación de seguridad que se pretende alcanzar.

En cualquier caso, será necesario elaborar una documentación pormenorizada y rigurosa en la que se reflejen cuáles son las demandas de los potenciales usuarios para que el sistema que se pretende desarrollar sea aceptado. Tomando como base el conocimiento de estos *requisitos de usuario* será necesario analizar cuáles de ellos se considera que son factibles de satisfacer mediante un desarrollo técnico, teniendo en cuenta los condicionantes que la tecnología disponible impone, es decir, determinando cuáles de esas demandas es posible implementar a un coste razonable (entendido esto en términos económicos y dependiendo del bien que se pretenda proteger). A partir de todo esto es que se elaborará un documento con los requisitos globales del sistema que se pretende desarrollar.

METODOLOGÍAS DE ANÁLISIS Y GESTIÓN DE RIESGOS

Como ya dejamos sentado en los primeros apartados del Capítulo 1, la seguridad total no existe: por muy fuerte que sea el escudo, siempre será factible (al menos desde un punto de vista teórico) encontrar un ataque que lo rompa. Ello nos obliga a encontrar una protección proporcional al bien que se pretende defender.

El análisis de riesgos se encarga de detectar las debilidades y de buscar remedio para ellas, cubriendo un doble objetivo:

- Estudiar los riesgos a los que está sometido un sistema de información y el entorno en que se encuentra inmerso.
- Dictaminar el establecimiento de las medidas (contramedidas) de seguridad apropiadas para prevenir y reducir esos riesgos.

De otra parte, la gestión de riesgos es el procedimiento que se diseña para implantar y mantener en el tiempo las contramedidas establecidas en el análisis de riesgos y así conseguir una seguridad adecuada y proporcionada en el funcionamiento del sistema.

Para abordar estas tareas de forma sistemática y fiable se han propuesto diversas metodologías cuya finalidad va a ser proporcionar una serie de recomendaciones de seguridad pensadas desde un punto de vista economicista, es decir, que el coste de las medidas de protección propuestas no supere el valor de las pérdidas (materiales o de otro tipo) que se producirían en el caso de que dichas medidas no se implantasen.

Entre todas las metodologías, existen fuertes coincidencias en los objetivos y en los conceptos y definiciones que manejan y difieren en el proceso a seguir y en la forma de plantear y organizar las medidas de los distintos parámetros puestos en juego. En cuanto a la forma de expresar la cantidad de riesgo o de valorar un bien, existen dos alternativas metodológicas bien diferenciadas. Si se trata de un bien o activo que hay que proteger sería:

- a) *Análisis cualitativo*. Se sopesa el *valor de uso* del activo, y permite conocer para qué sirve ese recurso.
- b) *Análisis cuantitativo*. Se sopesa el *valor de cambio*, o sea, cuánto vale el bien que puede estar amenazado.

Existen algunas metodologías, como es el caso de MAGERIT que ahora comentaremos, que utilizan ambas alternativas en sus análisis.

MAGERIT, Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información [Mage97], es una iniciativa de la Administración Pública española con objeto de crear unas normas para la elaboración de reglas de seguridad en los sistemas, que a su vez se encuentra dentro del marco en el que se están desarrollando criterios de certificación de seguridad (criterios que permitan valorar si un equipo o un producto que se quiera adquirir es seguro*). Ha sido promovido por el Consejo Superior de Informática a través de su comité técnico, el SSITAD.

* Con objeto de disponer de unos criterios y unos procedimientos para la evaluación de productos y sistemas de seguridad, la Comisión Europea aprobó en 1991 el documento ITSEC (*Information Technology Security Evaluation Criteria*) y en 1993 el manual ITSEM (*Information Technology Security Evaluation Manual*) que especifica la metodología a seguir. Estos criterios abarcan más elementos que su predecesor TCSEC (del gobierno de Estados Unidos).

Lo primero que hay que decir es que tanto MAGERIT como las restantes metodologías [Jara01][Jac01] abarcan todo lo relacionado con los sistemas de información y su entorno, más allá de lo estrictamente relacionado con las comunicaciones, las cuales serán, por tanto, un caso particular de todo ese conjunto.

El modelo formativo de MAGERIT se apoya en tres submodelos:

- *Submodelo de elementos de seguridad.* En él se recogen los componentes (con sus correspondientes atributos) que intervienen en el proceso.
- *Submodelo de eventos de seguridad* que relaciona entre sí, y en función del tiempo, los distintos componentes presentes en el Submodelo de Elementos.
- *Submodelo de procesos de seguridad* que representa la descripción funcional del proyecto de análisis a seguir. Se considera dividido en varias etapas que se aplican de forma sucesiva.

Los componentes presentes en el Submodelo de Elementos son bastante coincidentes con los recogidos en otras metodologías. Para cada uno de ellos se definen métricas (forma de medirlos). Son los siguientes:

- a) *Activos.* En otras propuestas se les denominan también *bienes y recursos*. Es cualquier elemento o recurso del sistema de información, o que esté relacionado con éste, que es necesario para que el sistema funcione. Se le asigna un valor y es necesario protegerlo. La lista de bienes puede ser muy extensa: edificios, sistemas de electricidad, servidores, terminales, datos, redes, protocolos, encaminadores (*routers*), y un larguísimo etcétera. Para evaluarlos se definen atributos y estados.
- b) *Amenazas.* Es cualquier posible fuente de peligro que puede desencadenar un incidente y producir daños materiales o pérdidas inmateriales. Se definen varios tipos de ellas.
- c) *Vulnerabilidades.* La vulnerabilidad de un activo representa la posibilidad o potencialidad de que una amenaza se materialice sobre ese activo. En términos matemáticos está relacionada con la probabilidad de que una amenaza previamente detectada tenga éxito. En MAGERIT está tipificada mediante tipos y atributos.
- d) *Impacto.* Es el resultado de la agresión sobre un activo. Cuando se produce la explotación de la vulnerabilidad por parte de una amenaza, el activo sufre un impacto, que se traduce en una pérdida de valor. Se distinguen varios tipos de impactos sobre los sistemas, alguno de los cuales puede dar lugar a responsabilidades penales.
- e) *Riesgo.* Se define como la posibilidad (la probabilidad) de que se produzca un impacto dado en un activo, es decir, que un agente amenazante explote una vulnerabilidad y provoque un efecto negativo en el sistema. Es el elemento resultante de toda la cadena de componentes antes mencionados y su medida es de suma importancia.
- f) *Salvaguardas.* Se define la *función de salvaguarda* como la acción que reduce el riesgo. Se distinguen:
 - *Salvaguardas preventivas*, que son las que ejercen una acción sobre la vulnerabilidad con objeto de neutralizar la materialización de la amenaza.
 - *Salvaguardas curativas*, que son las que actúan sobre el impacto tratando de restablecer el estado de seguridad del activo agredido.

A las funciones de salvaguardas se las denomina también *servicios de salvaguarda*, con objeto de armonizar la nomenclatura con la procedente de entornos OSI

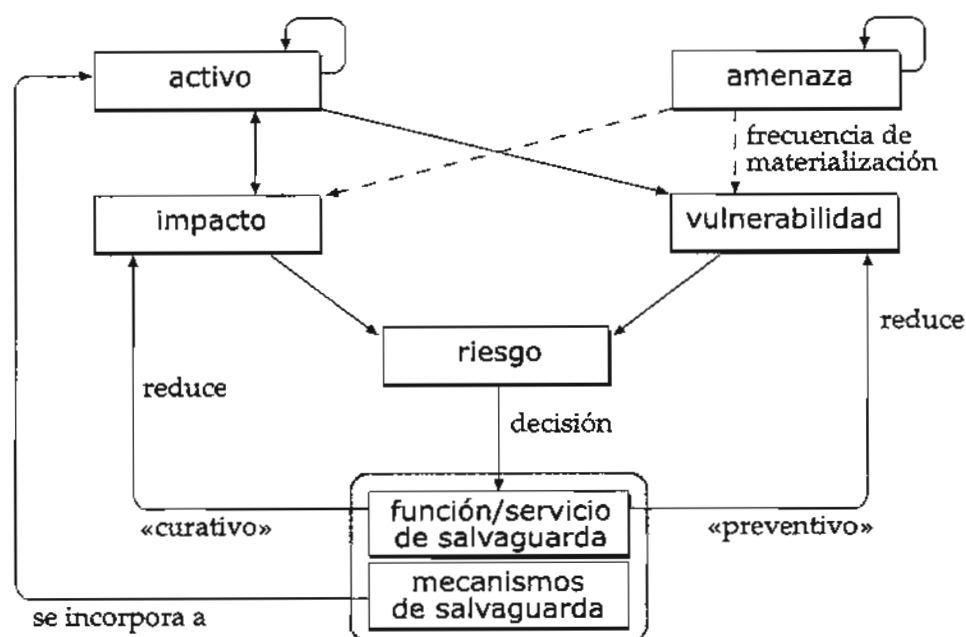


Figura 2.1. Relación entre los componentes del submodelo de elementos (Fuente: MAGERIT).

(recuérdese la definición de servicio que se recoge en el Capítulo 1, en concordancia con la norma ISO 7498-2). En esta misma línea se define:

Mecanismo de salvaguarda como el procedimiento o dispositivo (físico o lógico) que reduce el riesgo. El mecanismo de salvaguarda va íntimamente unido a la función de salvaguarda, ya que representa la concreción o materialización de esa función.

Como ya se dijo, los componentes que acabamos de describir son los que componen el *Submodelo de Elementos* de MAGERIT. El *Submodelo de Eventos* sirve para relacionar entre sí a los componentes antes descritos. Dentro de este Submodelo se distinguen, a su vez, varias «vistas» o tipos (Estático, Dinámico organizativo y Dinámico físico) que plasman otras tantas maneras de contemplar el sistema. En la Figura 2.1 se recoge la relación existente entre componentes conforme al tipo Estático del Submodelo de Eventos.

Por último, el *Submodelo de Procesos* está definido mediante cuatro etapas:

- Etapa 1. *Planificación*. Se establece y define el marco general para la realización del análisis y gestión de riesgos en relación con los problemas de seguridad del sistema objeto de estudio.
- Etapa 2. *Análisis de riesgos*. Se evalúa el riesgo del sistema en sus distintas vertientes y dominios, identificando y valorando los elementos que intervienen en el riesgo. Asimismo, se estiman y proponen los umbrales de riesgo deseables.
- Etapa 3. *Gestión de riesgos*. Esta etapa tiene como objetivo identificar y seleccionar las funciones de salvaguarda apropiadas para reducir el riesgo a un nivel aceptable.
- Etapa 4. *Selección de salvaguardas*. Por último, esta etapa tiene como objetivo seleccionar los mecanismos de salvaguarda que es conveniente implantar, establecer el procedimiento de su implantación y realizar la documentación global necesaria.

La Metodología MAGERIT está compuesta por un elevado número de normas contenidas en varios volúmenes, por lo que sería ocioso intentar reflejar aquí con

cierta claridad sus principales reglas y procedimientos. Por ello, lo único que hemos pretendido en los párrafos precedentes es recoger algunos aspectos de fácil asimilación con objeto de que el lector se haga una idea somera del alcance de estas técnicas, que, dada su amplitud, pueden considerarse una especialidad dentro del mundo de la securización de sistemas.

Por razones de oportunidad, en los capítulos que siguen no se aborda de nuevo esta temática, remitiendo al lector interesado a consultar directamente la bibliografía antes citada (en [Jara01] puede encontrarse una descripción más detallada de MAGERIT y un ejemplo de su aplicación práctica).

2.3. TERCERAS PARTES DE CONFIANZA, TTPs

Una TTP es un agente telemático, un sistema que, a requerimiento de partes, emite de forma automática piezas de información que pueden servir como *pruebas* o *evidencias* para que puedan ser provistos, con la debida eficacia, ciertos servicios de seguridad. Unas veces resulta imprescindible su uso para que el servicio sea implementable y otras veces resulta conveniente su instalación para reforzar y robustecer la seguridad.

Como su nombre sugiere, todos los elementos comunicantes pertenecientes al *dominio de seguridad* (conjunto de usuarios y sistemas que operan bajo una misma *política de seguridad*) consideran fiables y seguros los informes que emita una determinada TTP. Para ello, la gestión de las TTPs debe ser encomendada a personas y organizaciones de total confianza dentro del dominio. Además, a la TTP y a sus gestores se les supone una actitud imparcial en cuanto a los conflictos que puedan surgir entre los usuarios del dominio por motivos relacionados con el envío y recepción de mensajes y datos.

En líneas generales, el papel de la TTP se centra en la emisión de pruebas que puedan servir para la resolución de posibles litigios entre los entes participantes en las comunicaciones seguras. Por ello, para que estos informes tengan validez, es muy importante que los usuarios se aseguren de que la TTP es quien dice ser. Es decir, tienen que existir garantías (generalmente transferidas de forma extratelemática, usando medios convencionales) que permitan que el usuario reconozca, de forma inequívoca, los informes emitidos por una TTP.

UBICACIÓN DE LAS TTPs

Supongamos un escenario de comunicación en el que la Entidad A envía datos o mensajes a la Entidad B. Supongamos también que en este escenario simplificado está presente una TTP que coopera con esas dos entidades con objeto de facilitar la provisión de un servicio de seguridad. La TTP puede estar:

- a) Totalmente al margen de la comunicación activa entre ambos elementos comunicantes. Es decir, la TTP no opera interactivamente (Figura 2.2).
- b) Involucrada directamente en el proceso de intercambio de información. En este caso trabaja de forma interactiva (Figura 2.3).

En el primer caso, que podríamos denominar *fuera de línea* (*off-line*), la TTP sólo actuará a petición de partes, ya que no interviene directamente en el proceso de envío-recepción del mensaje y presta su servicio a un usuario cuando es requerida para ello.

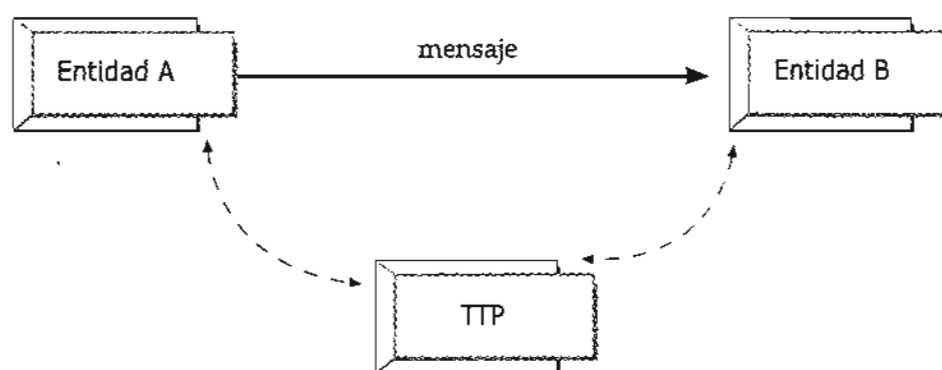
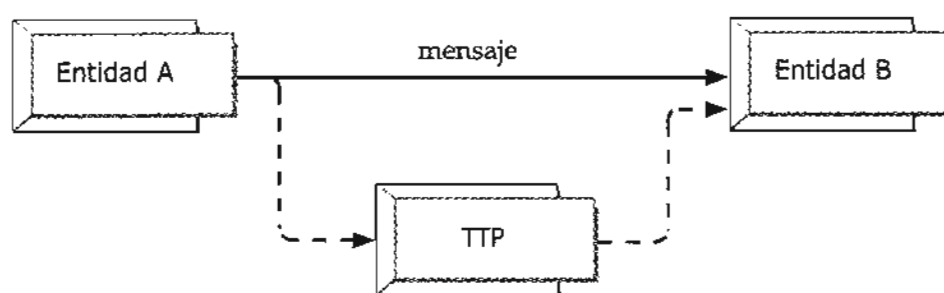


Figura 2.2. Comunicación usando una TTP *fuera de línea*.

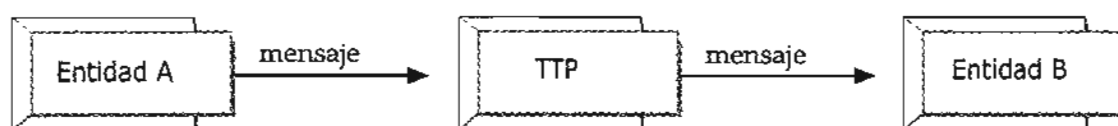
Resulta absolutamente imprescindible que en la comunicación entre las entidades y la TTP existan garantías robustas acerca de la autenticidad de la TTP y de las pruebas que emita.

En el segundo caso (Figura 2.3), la TTP estará presente en todas las transferencias de información que se produzcan aunque ello no le sea solicitado (o no sea deseado) por alguno de los actores de la comunicación. El quid de su operación está en la generación de *evidencias*, que puedan ser posteriormente almacenadas, recuperadas y verificadas en la resolución de disputas y litigios o, simplemente, como credenciales de que una determinada comunicación ha existido. Dentro de este caso aún cabe distinguir dos nuevas situaciones:

- b1) **TTP en línea (on-line)**, en el cual la TTP está activamente involucrada en el intercambio de información, pero no está incluida en el camino. Es decir, la Entidad A envía sus datos directamente a la Entidad B, pero la TTP monitoriza todas las transacciones, pudiendo generar las *evidencias* que le sean requeridas.



b1) TTP en línea



b2) TTP dentro de línea

Figura 2.3. TTP actuando interactivamente.

- b2) **TTP dentro de línea (in-line).** En este caso, la TTP está interpuesta en el camino de intercambio de datos entre el originador y el receptor del mensaje. Esto es, los datos son recibidos por ella y reenviados al receptor, generando y enviando, además, las correspondientes *evidencias*.

SERVICIOS Y FUNCIONES

Las funciones que desempeñan las TTPs son múltiples y variadas, dependiendo del escenario en el que actúen. Entre las más significativas, cabe destacar:

- Entidades que certifican la validez de una pieza de información.
- Dispositivos de monitorización de los eventos que se producen en la red.
- Fedatarios (o Notarios) electrónicos que dictaminan sólo a los efectos del dominio de seguridad en el que se encuentran. (La validez legal de estos dictámenes depende de otros factores adicionales.)

Con mayor o menor protagonismo, las TTPs deben estar presentes cuando se necesita proveer de forma robusta cualquier servicio de seguridad. Aunque en los Capítulos 6 al 9 se estudian todas ellas de forma pormenorizada, referiremos ahora las más significativas. Así, podemos destacar:

a) *Autenticación e Integridad*

Como ya se ha comentado y se justificará más adelante, la *firma digital* puede garantizar en gran medida la provisión de estos dos servicios. Para ello, resulta imprescindible que existan garantías acerca de la validez de las claves públicas. Esto es así porque la seguridad que tiene el receptor de un mensaje firmado por A acerca de la validez de la firma depende de la confianza que tenga en que ha sido la entidad A, y no otra, quien ha realizado el cifrado con su clave privada.

Esa confianza la adquiere el receptor si tiene garantías de que la clave pública de A (a la que tiene acceso por ser pública) pertenece realmente a A. Para ello se ha definido el *certificado digital*, que es una pieza de información en la que se asocia el nombre de una entidad con su clave pública (pareja de la clave privada correspondiente) durante un determinado periodo de validez.

Casos típicos de TTPs, todas ellas *fuera de línea (off-line)*, que sirven para este cometido son:

- **Autoridades de Certificación, CAs**, que firman y emiten *certificados*. Las CAs son de capital importancia y puede afirmarse que no puede existir un escenario de seguridad, donde se use criptografía de clave pública, en el que no existan una o varias CAs. Además de la fundamental tarea de generar certificados, las autoridades de certificación, CAs, pueden llevar a cabo otras muchas competencias como son generación y distribución de claves, almacén y gestión de certificados, etc., aunque la tendencia existente es crear TTPs especializadas para estas funciones complementarias.
- **Autoridades de Registro, RAs**, que sirven para dar de alta, con garantías, a los usuarios de un determinado dominio de seguridad.
- **Autoridades de Verificación de Firmas**, que pueden actuar de árbitro, dentro del dominio, cuando aparece un conflicto entre partes.
- **Autoridades de Sellado de Tiempo, TSAs**, que sirven para garantizar que determinados datos existían antes de una fecha y hora concretas.

b) *Confidencialidad*

Este servicio se apoya en la confianza acerca de la validez de las claves públicas con que se cifran los mensajes (o con las que se cifra la clave secreta con la que se encripta el mensaje). Por eso, la TTP que facilita su provisión robusta es también la CA, junto con los agentes que le auxilian en su cometido.

c) *Control de acceso*

Existen TTPs, denominadas *Autoridades de Atributos, AAs*, que generan *certificados de atributos* emitidos a nombre de un usuario concreto. Estos certificados son piezas de información firmadas por la TTP en las cuales se especifican los privilegios de uso que dicho usuario tiene ante un determinado servidor u otro recurso de la red.

d) *No Repudio*

Resulta imprescindible para la provisión de este servicio, en sus distintas variedades, el concurso de *TTPs de No Repudio* especializadas. Casi todas ellas son del tipo *en línea (on-line)* o *dentro de línea (in-line)* y emiten *evidencias* que son piezas de información firmadas que dan fe de la ejecución de determinados eventos.

Además de TTPs específicas, existen determinados sistemas que, en determinadas circunstancias, operan de forma similar. Así, en el correo electrónico X.400, el sistema de Transferencia de Mensajes, MTS, cuando está dotado de servicios de seguridad, puede ser considerado como una TTP distribuida. No obstante, si se utiliza un servicio de correo no OSI (por ejemplo SMTP, MIME o PGP) y se quiere implantar un servicio de No Repudio, resulta imprescindible la utilización de una TTP especializada que actúe como intermediaria.

Igualmente, el Servicio de Directorio X.500, DS, provisto de servicios de autenticación, integridad y control de acceso, puede ser también considerado como una TTP distribuida que permite depositar en su base de datos información de seguridad sensible dentro de un determinado dominio.

USO DE MÚLTIPLES TTPs. INFRAESTRUCTURAS DE SEGURIDAD

Cuando los escenarios de seguridad crecen en complejidad y aumenta el número de servicios con los que son provistos los usuarios, se hace necesario el uso de múltiples TTPs. Cuanto más amplio sea el rango de servicios ofrecidos, tanto más variado debe ser el tipo de TTPs que es necesario implementar. Del mismo modo, cuando el escenario de comunicación es disperso y complejo será necesario que existan múltiples TTPs del mismo tipo distribuidas teniendo en cuenta consideraciones tanto geográficas como topológicas y organizacionales.

Entre ellas es necesario establecer complicadas y estrictas reglas de dependencia para mantener la robustez del sistema. Tal es el caso del conjunto distribuido de CAs y otras TTPs relacionadas con la gestión de las claves públicas, que se conoce bajo el nombre de *Infraestructuras de Certificación* o *Infraestructuras de Clave Pública*, PKIs, acrónimo que se corresponde con su denominación en inglés, *Public Key Infrastructures*.

Puede asegurarse que la evolución y mejora de los escenarios de seguridad, y el refinamiento en la provisión de servicios, viene de la mano del empleo de un variado y especializado conjunto de TTPs. Tal es la situación que se perfila en los nuevos escenarios donde, en determinadas actuaciones, es necesario garantizar la privacidad

y el anonimato del usuario. Este es el caso del dinero electrónico, de las tarjetas de crédito anónimas o del voto telemático.

En todas estas aplicaciones, nuevas TTPs especializadas han sido diseñadas o están en vías de especificación. En el estado actual de la tecnología, todas estas TTPs se basan en el uso de criptosistemas de clave pública. Por tanto, todo este conjunto ordenado y variado de TTPs operando bajo reglas precisas y no ambiguas, bajo cuyo paraguas pueden tener lugar las múltiples comunicaciones seguras que necesitan los ciudadanos, puede considerarse que constituyen entornos ampliados de infraestructuras de seguridad de clave pública (PKIs).

Debido a ello, a veces, estos entornos ampliados son denominados también PKIs, aunque sus funcionalidades excedan a las estrictamente relacionadas con la distribución y gestión de certificados. Aquí los denominaremos, de forma genérica, *Infraestructuras de Seguridad*, según trataremos de justificar en el Capítulo 7.

2.4. INTERFACES DE USUARIO Y TARJETAS INTELIGENTES

Por una parte, las tarjetas inteligentes sirven para mucho más que para facilitar a los usuarios el acceso a los servicios de seguridad y, por otra, la problemática de las interfaces de usuarios comporta muchos más aspectos que los que puedan estar relacionados con el uso de tarjetas. No obstante, en una visión panorámica y general de la seguridad en redes telemáticas, es conveniente resaltar esta ligazón de ambos conceptos y tecnologías, ya que con gran frecuencia aparecen íntimamente unidos y relacionados.

CARACTERÍSTICAS DE LAS TARJETAS INTELIGENTES

Una tarjeta inteligente (*smart card* o *IC* en inglés) es una tarjeta de plástico de aspecto similar a las tarjetas de crédito convencionales, dotada de un circuito integrado (por eso se las ha llamado también *tarjetas chip*) que contiene una CPU o microprocesador, memoria volátil y memoria ROM. Esta memoria aloja las pequeñas aplicaciones de usuario y un reducido Sistema Operativo, el SCOS o *Smart Card Operating System*, también denominado *máscara* o *Smart Mask*. Utilizando estos componentes y una circuitería adicional, la tarjeta permite guardar datos particulares de cada usuario y datos para la aplicación específica. Dependiendo de la capacidad de la tarjeta, también se pueden almacenar en ella claves criptográficas, certificados, programas de cifrado, e incluso una serie de informaciones asociadas al entorno en el que se use la tarjeta, haciendo que ésta funcione como memoria *caché* de una aplicación o de una serie de aplicaciones.

En la Figura 2.4 se representa el aspecto externo de una tarjeta inteligente con microprocesador incorporado. Se ha supuesto que esa tarjeta puede servir como componente identificador, por lo que lleva espacio para el nombre de la persona titular, la foto, el nombre de la organización a la que pertenece, etc. Resulta evidente que todos estos elementos gráficos son accesorios y prescindibles, ya que lo único sustantivo es el microcomputador que lleva incorporado. Es más, a veces puede interesar que la tarjeta vaya en blanco y que las identificaciones necesarias se realicen solamente a través de los datos y algoritmos almacenados en la memoria del microcomputador.

La comunicación con el exterior se realiza a través de unos pequeños contactos físicos que la ponen en comunicación con la ULE, *Unidad de Lectura-Escritura* (en inglés, *WRW*, *Writing-Reading Unit* o *CAD*, *Card Acceptance Device*). Algunas tar-

jetas carecen de contactos directos y la transmisión se realiza mediante radiofrecuencia. La tarjeta puede llevar a cabo un conjunto limitado de funciones criptográficas y una serie de funciones de comunicación con el exterior a través de la ULE. Estos periféricos son muy simples y de bajo coste, de forma que su instalación en los PCs o terminales de usuarios no presentan relevantes problemas técnicos o económicos. Además, existen ULEs que se pueden insertar en la ranura destinada a los disquetes portátiles, con lo cual se reducen aún más los requisitos de equipamiento necesarios.

Aunque existen varias generaciones de tarjetas inteligentes que han ido resolviendo sucesivamente distintos problemas de seguridad, en el estado actual de la tecnología de Seguridad en Redes sólo tienen especial interés aquellas que están dotadas de mecanismos criptográficos, tanto de clave secreta o simétricos como de clave pública, tipo RSA. Al incorporar criptografía de clave pública, garantizan el almacenamiento de la *clave privada* del usuario de forma segura, ya que las operaciones de cifrado que deban realizarse usando esta clave se llevarán a cabo *dentro de la tarjeta*, garantizando que la clave nunca viaje por la red ni salga de los estrictos límites de la tarjeta.

Algunas veces, la generación del par de claves de cada usuario (privada y pública) se realiza en un centro especializado y autorizado para ello dentro del dominio de seguridad. En este caso, las claves son introducidas en la tarjeta en la fase de personalización, siendo ésta entregada al usuario para su uso y custodia. En ese momento, el centro generador de claves debe borrar la clave privada del usuario para que sea éste, y sólo éste, quien posea esa clave, responsabilizándose (incluso jurídicamente) de todas las operaciones de cifrado que con ella se realicen.

No obstante, es menester hacer notar que en las recomendaciones existentes acerca de la implantación de servicios de seguridad se preconiza que cada usuario, una vez recibida una primera versión de su par de claves (privada y pública), esté facultado para generar un nuevo par de claves, almacenando por sus propios medios la clave privada en su tarjeta inteligente y haciendo llegar la clave pública a una Autoridad de Certificación para que genere el correspondiente certificado. De esta manera es insoslayable la responsabilidad que el usuario contrae en cuanto al uso y custodia de sus claves, no existiendo resquicio alguno que le permita atribuir a terceros una posible firma que haya sido realizada usando su propia clave privada.

Algunas tarjetas inteligentes están capacitadas para generar de por sí un nuevo par de claves, con lo que se refuerzan estas salvaguardas de confinación de la clave privada dentro de la tarjeta sin posibilidad de dejarla almacenada, por olvido o por otras circunstancias, en la memoria de un ordenador personal, en un disquete o en cualquier otro soporte no suficientemente protegido. Es evidente que, siempre que ello sea posible, son tarjetas de este tipo las que realmente interesan para ser utilizadas en la securización de escenarios de comunicación.

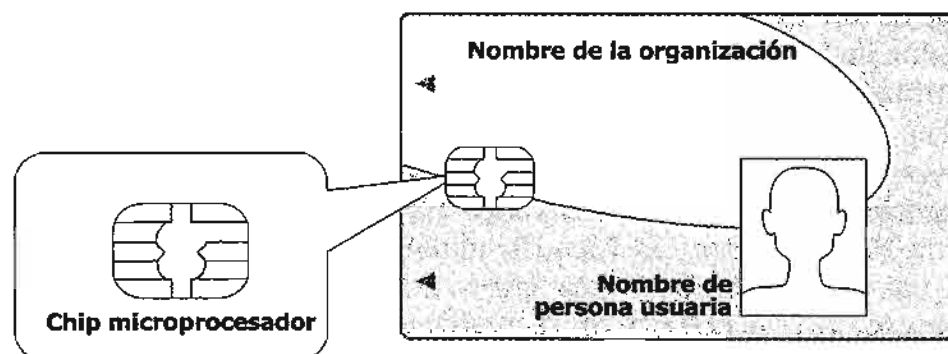


Figura 2.4. Tarjeta inteligente con microprocesador incorporado.

Las tarjetas inteligentes están dotadas de mecanismos de protección tanto físicos como lógicos. Estos últimos regulan el control de acceso a los datos y a las funciones de la tarjeta. Si una tarjeta es extraviada o le ha sido sustraída a su legítimo propietario, no es posible leer los datos contenidos en su memoria o realizar modificación o copia de éstos. Este tipo de protección física y lógica garantiza que aunque la tarjeta sea destruida, no sea posible acceder por medios electrónicos al contenido de la memoria del chip. Debido a esta propiedad, podemos decir que la tarjeta tiene *protección ante fraude*, o *protección ante manipulaciones*, o que es *resistente a manipulaciones* (en inglés, *tamper-proof* o *tamper-resistant*).

Para garantizar que sólo el usuario legítimo pueda hacer uso de la tarjeta, varios tipos de controles lógicos de acceso pueden ser implementados. Estos controles podrían basarse en:

- a) *Lo que el usuario sabe hacer.* Se trata de alguna actividad o habilidad privativa del usuario poseedor de la tarjeta. El ejemplo más extendido sería hacer uso de la firma caligráfica. Para autenticarse, se exigiría al poseedor de la tarjeta que realice su firma usando para ello un lápiz electrónico, de tal manera que puedan ser analizadas no sólo la forma y presión de los trazos de la firma, sino incluso la velocidad de ejecución. Presenta cierta complejidad técnica y se da el caso de que muchas personas al «estampar» la firma introducen variantes difícilmente analizables por medios automáticos, por lo que es un método de control de acceso poco extendido y no recomendable.
- b) *Por lo que el usuario conoce.* El caso más simple es un número personal de identificación, PIN, almacenado en la tarjeta, que el usuario debe teclear para permitir que la ULE o CAD pueda llevar a cabo las operaciones de comunicación. Puede ampliarse a palabras o contraseñas alfanuméricas más complejas y a intercambios reiterados de preguntas/respuestas, algunas de ellas relacionadas con datos personales que sólo el interesado puede conocer. El uso del PIN está muy extendido y ofrece, junto con un cuidado especial en la custodia de la tarjeta, garantías de seguridad bastante aceptables, aunque es conveniente reforzarla mediante un mecanismo complementario.
- c) *Por cómo el usuario es.* Se trata de registrar en la tarjeta algunas *características biométricas* del usuario. Es el procedimiento de autenticación más fiable de todos y acabará imponiéndose en todos los esquemas de comunicación a medida que los procedimientos tecnológicos se vayan perfeccionando y abaratando. El más sencillo y el que menos equipamiento físico requiere es el reconocimiento de voz, aunque parece que a veces se dan coincidencias, principalmente entre familiares directos, difíciles de discriminar. Como es bien sabido, la huella dactilar conlleva unas características particulares de cada usuario y está siendo utilizada con éxito, y con un coste no excesivo, en múltiples entornos prácticos, aunque también es posible falsificar extrayendo copias de la huella en materiales plásticos. También el iris del ojo puede ser usado como característica distintiva del usuario; es el método más difícil de falsificar y cada vez resulta más asequible su incorporación.

El empleo de mecanismos biométricos para garantizar la autenticación del titular de una tarjeta representa un avance notable para proteger su privacidad y sus derechos: sirve para evitar que alguien se haga pasar de forma malintencionada por quien no es, mejorando así notablemente la seguridad de las redes.

No obstante, el desarrollo de estas técnicas de identificación personal sigue su propio camino al margen de la utilización a la que hemos hecho referencia en el

párrafo anterior. En efecto, existe bastante interés en algunos círculos en desarrollar sistemas de identificación de personas (no sólo en base a sus características faciales, sino también analizando el iris del ojo) al margen de su deseo, para controlar el movimiento de ciudadanos (aquí pegaría mejor decir «individuos») en aeropuertos y otros lugares públicos. Estas prácticas conllevan un elevado riesgo de establecimiento de vigilancias sobre ciudadanos corrientes que pueden suponer la pérdida de alguno de sus derechos.

No ha lugar el sacar aquí a debate la bondad o los peligros de estas técnicas. Bástenos con resaltar que aunque las tecnologías empleadas en control de acceso y en vigilancia presentan coincidencias, las seguridades que procuran conseguir son distintas (cuando no contrapuestas). En el Capítulo 11 volveremos a insistir sobre estas cuestiones.

INTERFACES DE USUARIO

Cuando se intenta implantar un servicio telemático cualquiera hay que tener en cuenta que el peor o mejor acabado de las **interfaces** que dan acceso al usuario es, la mayoría de las veces, decisivo de cara al éxito o fracaso del sistema que se desarrolle para ello. La interfaz de usuario influirá no sólo en el grado de satisfacción subjetiva de las personas que lo utilicen, sino también, quizás debido a ello, en el grado de implantación y aceptación final del sistema en su conjunto.

Si además lo que se quiere llevar a cabo es la securización de una aplicación, debemos procurar que la introducción de estas tecnologías novedosas, que aportan protección a los servicios que ya de por sí proporcione la aplicación, se haga de manera tal que el usuario no se vea abrumado por detalles farragosos de estas técnicas, pero a la vez comprenda el significado de las operaciones de seguridad que está aplicando. El desconocimiento del significado de los procedimientos empleados puede conllevar un rechazo frontal del usuario ante una innovación que le resulte opaca y ajena.

En cuanto a las interfaces de usuario, se puede apreciar que en los últimos años los usuarios se han ido acostumbrando de forma creciente a manejar aplicaciones informáticas a través de ventanas e iconos. Esto permite esperar que al futuro usuario de servicios de seguridad no le vaya a suponer mucha dificultad el interpretar una presentación en la que se muestren como iconos los objetos disponibles (documentos, materiales criptográficos, evidencias, etc.) y como botones de una caja de diálogo los posibles tratamientos que a éstos se le pueden aplicar (firma, almacenamiento, verificación, etc.). Por ello, resulta imprescindible que en el desarrollo de *interfaces amables* sean tenidos en cuenta los aspectos ergonómicos que hagan que el usuario se sienta confiado en las operaciones que realiza y esté, al mismo tiempo, liberado de la complejidad que ellas comportan.

Como ya se ha dicho, las tarjetas inteligentes avanzadas, que empiezan a estar disponibles en el mercado, permiten llevar a cabo, en cooperación con la aplicación residente en el terminal de usuario (a la que éste tiene acceso a través de su interfaz), una serie de funciones de seguridad que sirven para proteger la mayoría de las operaciones necesarias para las transacciones e intercambios de información.

La incorporación de la tarjeta inteligente para acceso a terminales de usuario es en sí misma un elemento lleno no sólo de utilidades prácticas, sino también de matices simbólicos. De una parte, el usuario dispone de un dispositivo *personal*, cuya posesión y uso no difiere en mucho de los de otras tarjetas de plástico, donde puede llevar información sobre claves, algoritmos y otros datos que a él le afectan

y que tras su inserción en la unidad de lectura-escritura y la correspondiente autenticación le va a permitir trabajar directamente sobre la interfaz sin necesidad de aportar más datos.

Casi tan importante como la fiabilidad que adquieren los procedimientos telemáticos al contar con la garantía de que la clave secreta no abandona en ningún momento la tarjeta, es la sensación subjetiva de tranquilidad que acompaña al usuario cuando retira la tarjeta del sistema y *se la lleva puesta*. Es por esto por lo que, a la hora de establecer los procedimientos de securización, se puede aprovechar al máximo el simbolismo que la tarjeta aporta como «portadora» de la identidad de su titular.

La tarjeta constituye, así, una pieza clave en la configuración de interfaces amables o amigables (*friendly interfaces*) para el acceso a servicios telemáticos dotados de servicios de seguridad. Por todo ello, no es aventurado afirmar que, aunque la tarjeta no representa en sí misma un concepto novedoso en las técnicas de seguridad, su utilización combinada con interfaces amables puede resultar decisiva para la implantación masiva de los servicios de seguridad en redes.

2.5. UBICACIÓN DE LOS SERVICIOS DE SEGURIDAD

Con mucha frecuencia, en los escenarios de comunicación seguros que pueden presentarse intervendrán entidades comunicantes situadas en distintas redes interconectadas. En la inmensa mayoría de los casos, esas redes estarán configuradas conforme a arquitecturas telemáticas jerárquicas organizadas en niveles, bien sea basadas en el Modelo de Referencia OSI, bien en protocolos de la familia TCP/IP.

Existe una diferencia notable entre las distintas partes y componentes de una red. Si partimos de una concepción de las comunicaciones estructurada en niveles, sería muy diferente, tanto desde un punto de vista metodológico como de comportamiento funcional, implementar servicios de seguridad en, por ejemplo, el nivel de enlace que el hacerlo en el nivel de Aplicación. Cabría preguntarse, por tanto, cuál es la capa o nivel más adecuado para implantar servicios de seguridad.

En este apartado se exponen, de forma resumida, los planteamientos de los organismos internacionales de normalización acerca de este tema. Una respuesta de Perogrullo podría, no obstante, adelantarse: depende de a qué parte o a qué elementos de la red tenga acceso quien se plantea la instalación de estos servicios.

Así, si una empresa o una organización tiene repartidas sus dependencias entre distintos sitios del país, comunicándose a través de una red pública de conmutación de paquetes (basada, por ejemplo, en IP), no tiene mucho sentido que a la hora de plantearse la protección de sus comunicaciones tome la decisión de securizar el nivel de red, ya que la responsabilidad de esos protocolos cae bajo el dominio de los propietarios de la red pública y sólo podrá disponer de servicios de seguridad en ese nivel si los gestores de esa red han decidido implantarlos y ofrecérselos a sus usuarios. El responsable de comunicaciones en esta hipotética empresa u organización tendría que buscar otras soluciones si quiere implantar servicios de seguridad de forma autónoma. Asegurar los accesos a la red, hacerlo con las aplicaciones o conseguir un nivel de transporte seguro, serían posibles alternativas.

La Tabla 2.1 recoge una posible relación de los distintos servicios de seguridad que pueden ser proporcionados en cada uno de los siete niveles de una arquitectura telemática jerarquizada conforme el Modelo de Referencia OSI, según se especifica en la norma «matriz» ISO 7498-2 *Security Architecture*, a la que ya antes hemos hecho referencia. Como la propia norma hace constar, no se trata de una propuesta que deba

TIPO DE SERVICIO	NIVELES						
	1	2	3	4	5	6	7*
Autenticación de entidad Par	-	-	S	S	-	-	S
Autenticación del origen de los datos	-	-	S	S	-	-	S
Control de Acceso	-	-	S	S	-	-	S
Confidencialidad (en servicio orientado a la conexión)	S	S	S	S	-	S	S
Confidencialidad (en servicio no orientado a la conexión)	-	S	S	S	-	S	S
Confidencialidad en campos seleccionados	-	-	-	-	-	S	S
Confidencialidad en flujo de tráfico	S	-	S	-	-	-	S
Integridad (en servicio orientado a la conexión con recuperación de errores)	-	-	-	S	-	-	S
Integridad (en servicio orientado a la conexión sin recuperación de errores)	-	-	S	S	-	-	S
Integridad en campos seleccionados (en servicio orientado a la conexión)	-	-	-	-	-	-	S
Integridad (servicio no orientado a la conexión)	-	-	S	S	-	-	S
Integridad en campos seleccionados (servicio no orientado a la conexión)	-	-	-	-	-	-	S
No - Repudio de origen	-	-	-	-	-	-	S
No - Repudio de entrega	-	-	-	-	-	-	S

- Leyenda
- S Significa que sí debe ser incorporado este servicio como una opción en los estándares del nivel.
 - Significa que no debe proveerse el servicio.
 - * Cabe señalar que (respecto al nivel 7) los procesos de aplicación pueden, por sí mismos, proporcionar servicios de seguridad.

Tabla 2.1. Relación entre Niveles y Servicios de Seguridad en la Arquitectura OSI.

ser estrictamente cumplida, sino de una reflexión acerca de lo adecuado o no de implantar servicios de seguridad en determinados niveles.

En esta tabla, como la propia norma puntualiza, no se trata de equiparar en peso e importancia las distintas indicaciones acerca de qué servicios ha de haber en cada uno de los niveles. No obstante, nos va a servir para hacer unas consideraciones generales sobre la ubicación de los servicios de seguridad que es válida tanto para el modelo arquitectural OSI como para lo que, algo vaga e imprecisamente, podemos denominar «arquitectura Internet».

En primer lugar, se observa que, *grosso modo*, los servicios de seguridad están concentrados en tres niveles: red, transporte y aplicación. La sugerencia de colocar salvaguardas en los dos primeros niveles es meramente testimonial y se refiere sólo a la confidencialidad. Si se tiene en cuenta que en los entornos locales esos dos niveles se corresponden con los niveles físicos y de acceso al medio de las LANs, además de que en la arquitectura Internet estos niveles no están claramente especificados, sino que se dejan al arbitrio de cada subred, se puede llegar a la conclusión de considerar que va a tener muy poco interés el establecer normas internacionales para securizar los niveles físico y de enlace.

Gran importancia ha de tener, en cambio, la provisión de servicios de seguridad a nivel de red, y así se ha visto reflejado tanto en entornos X.25 como IP. La norma IPSec, que inicialmente fue desarrollada sólo para formar parte de la versión 6 de IP, está siendo incluida como una ampliación de la muy extendida IPv4, y comporta un completo conjunto de servicios de seguridad que operan en entornos PKI. Obviamente, la securización en este nivel cae bajo la responsabilidad de los proveedores del servicio de red, siendo ofrecidas las nuevas facilidades de seguridad como añadido a las específicas de encaminamiento de paquetes que le son propias.

Habida cuenta que el nivel de transporte es extremo a extremo (*end_to_end*), los servicios de seguridad a implantar aquí deben servir para completar los que el nivel de red proporcione o para suplir los que no proporcione. En la práctica, los niveles 3 y 4 trabajan de forma muy complementaria y conjunta, tanto en la Arquitectura Internet, donde TCP e IP aparecen siempre en forma de bloque, como en la Arquitectura OSI, donde en función del tipo (calidad) de red en que se apoye, así será la clase (TP0 a TP4) de protocolo de nivel 4 que se implante. Por ello, el fortalecimiento del nivel de red mediante un nutrido y robusto conjunto de servicios de seguridad permite bajar la guardia en lo que al nivel de transporte se refiere. Además, para asegurar servicios extremo a extremo, teniendo en cuenta las particularidades de los usuarios finales, se dispone también del nivel de Aplicación.

En lo que a los niveles altos se refiere, puede observarse en la Tabla 2.1 que en el nivel 5 del Modelo de Referencia OSI no se propone ningún servicio. Además, la inclusión de tres clases de servicios de confidencialidad en el nivel de presentación es un ejercicio puramente academicista, según se deduce de las explicaciones recogidas en la norma que propone dicha tabla. No podría ser de otra forma, ya que el nivel 5 se encarga de ayudar al de Aplicación en la regulación del diálogo y en la sincronización de los datos, mientras que el nivel 6 casi no tiene existencia como tal nivel, sino solamente como un conjunto de funciones auxiliares para la representación de los datos (aunque sumamente importantes de cara al funcionamiento global de la arquitectura).

(Como dato curioso, cabe señalar que cuando, a finales de los setenta y principio de los ochenta del pasado siglo, se definió el Modelo OSI, se pensaba que *seguridad* era casi sinónimo de *cifrado* y que el nivel encargado de esta función debería ser, en exclusiva, el nivel de Presentación. Esa es la razón por la que con frecuencia, en los tratados clásicos sobre redes de ordenadores, entre ellos el muy conocido texto de Tanenbaum [Tane89], los temas de seguridad y cifrado se abordaban en los capítulos dedicados al nivel 6. Posteriormente, debido de una parte al desarrollo de los distintos niveles de la arquitectura OSI y al refinamiento en cuanto a las funciones encomendadas a cada uno de ellos, y de otra a la conceptualización de la seguridad en redes llevada a cabo con la generación de la norma sobre *Security Architecture*, esta percepción primitiva se perdió por completo hasta el punto de considerar prácticamente excluido el nivel de Presentación de las tareas de protección de las comunicaciones.)

En la propuesta reflejada en la Tabla 2.1 podemos contemplar que, en los niveles altos, son las aplicaciones de nivel 7 las que concentran la mayor parte de los servicios de seguridad (todos los imaginables) que protegen al usuario final, estableciendo salvaguardas en la relación entre las distintas entidades comunicantes. Esta percepción es tanto más evidente en el Modelo Internet, donde las funcionalidades de los tres niveles altos del Modelo OSI están fundidas en un único nivel de Aplicación.

Puede observarse que el Servicio de Anonimato no aparece recogido en la Tabla 2.1. Ello es así porque cuando la norma ISO 7498-2 fue publicada, aún no se había detectado la necesidad de este servicio, que, por razones de complejidad, estará ubicado casi exclusivamente en el nivel de Aplicación.

Este reforzamiento de servicios de seguridad en las aplicaciones telemáticas puede contemplarse bajo dos situaciones:

1. Para disponer de servicios telemáticos seguros que se apoyen en redes inseguras, controlando desde las aplicaciones todas las salvaguardas que se implanten. Esta situación es la que se produce en multitud de casos y es la que seguirá apareciendo en tanto en cuanto no se doten de servicios de seguridad todas las redes intermediarias por las que han de circular los datos de una comunicación.
2. Para reforzar los servicios de seguridad ya establecidos en los niveles intermedios (por si acaso). Además, la ubicación de servicios de seguridad en el nivel de Aplicación permite al usuario final conservar un tanto por ciento importante del control de la seguridad, imprescindible para garantizar, en la mayoría de los casos, la privacidad en las comunicaciones.

2.6. CORTAFUEGOS

En el epígrafe anterior se ha estado analizando la ubicación de los servicios de seguridad en los distintos niveles en que se considera descompuesta una arquitectura telemática jerarquizada. Este planteamiento se corresponde con un paradigma basado en comunicaciones de todos con todos mediante sistemas abiertos. Es decir, en escenarios de comunicación constituidos por sistemas cuya estructura y diseño internos son privativos de cada uno de los entornos locales, pero que sí tienen concertada la forma de comunicarse entre sí (los protocolos telemáticos bajo los que operan) siguiendo las especificaciones de un determinado conjunto de normas. Las protecciones necesarias en el escenario de comunicación en el que participen (es decir, los servicios de seguridad) estarán implementadas en todos y cada uno de ellos conforme a lo establecido en la política de seguridad correspondiente.

Por contra, el escenario de comunicación en el que se definen los *cortafuegos* (*firewalls* en inglés) es totalmente distinto: se trata de proteger una red privada contra intrusiones desde el exterior concentrando la protección en los puntos de acceso. Es decir, dada una red privada que es necesario relacionar con el entorno exterior (pensemos, por extensión, en Internet), se trata de diseñar una puerta por donde debe pasar toda la información que entre o salga de la red para garantizar su adecuado funcionamiento. En realidad, más que de cortafuego habría que hablar de una muralla con una sola puerta de entrada bien guarnecida.

Para completar la panorámica sobre los conceptos y sistemas presentes en la seguridad en redes que se ha venido exponiendo en los distintos apartados de estos dos primeros capítulos, en los párrafos que siguen se pretende hacer una presentación muy resumida de las características y de las funciones propias de los cortafuegos (que, por otra parte, no serán estudiados de nuevo en los siguientes capítulos). No se pretende, por tanto, hacer una descripción detallada y diferenciada de las distintas arquitecturas existentes, sino una somera descripción que sirva para completar la idea de configuración de los distintos escenarios de seguridad posibles.

Al mismo tiempo, se pretende dejar bien sentado que para la inclusión en los cortafuegos de servicios de seguridad robustos serán necesarios los mismos mecanismos de seguridad que para el resto de las configuraciones. Al estudio de esos servicios, y de los mecanismos criptográficos en que se soportan, están dedicados los restantes capítulos que componen este texto.

FUNCIONES QUE CUMPLE UN CORTAFUEGO

Un cortafuego es un sistema o un conjunto de ellos que sirve para establecer una frontera y unos puntos de acceso entre una red privada que se pretende proteger y el entorno exterior de redes con las que se desea establecer comunicación. Puede tratarse de un simple encaminador (*router*) o de un servidor (*host*) o conjunto de servidores que sirven para proteger a la red de accesos indeseados desde el exterior y evitan que usuarios y entidades internas accedan a servicios y servidores externos que se consideren nocivos o inadecuados.

La función principal del cortafuego es obligar a que todas las conexiones que se establezcan entre el interior y el exterior de la red protegida pasen a través de él, que hará de intermediario, para ser inspeccionadas y permitidas o denegadas conforme a la política de seguridad establecida. A veces puede actuar también como un «escaparatista» o «embajador» de la red ante los usuarios ubicados en su exterior, de forma que muchas empresas u organizaciones usan también el cortafuego como un lugar donde almacenar información pública acerca de sus productos o cometidos.

No existe un único tipo de cortafuego, por lo que no es posible describir con precisión cuáles son las funciones que estos componentes llevan a cabo. No obstante, entre las principales funciones o tareas que estos dispositivos realizan, podemos destacar:

1. Protección contra ataques basados en el encaminamiento de paquetes IP. El cortafuego puede proporcionar protección contra varios tipos de ataques basados en paquetes encaminados en origen o en intentos de establecer nuevas rutas a través de redirectores ICMP. La existencia de un único punto de entrada facilita esta tarea y permite informar al administrador de estas incidencias.
2. Protección contra servicios indeseados. El cortafuego puede prohibir que determinados servicios, que se considera pueden introducir vulneraciones de la seguridad, entren o salgan de la red protegida, aunque sí estén permitidos de forma interna.
3. Control de acceso a los recursos de la red. Algunas máquinas o servidores de la red protegida podrían ser accesibles desde el exterior, mientras que otras quedarían aisladas. El cortafuego puede exigir la autenticación de los usuarios externos que deseen entrar en comunicación con alguno de los servidores o estaciones de la red, permitiendo o denegando el acceso de forma selectiva.
4. Monitorización de eventos relacionados con la seguridad de la red. El cortafuego puede realizar un registro pormenorizado de todos los eventos ocurridos, de forma que sean posteriormente analizados. De esta forma es posible conocer qué usuarios se han conectado, durante cuánto tiempo, intentos fallidos de penetración en el sistema, métodos de ataques, etc. En otros casos, puede estar dotado de un sistema de alarmas que avise al administrador de la red de que está siendo atacada o se está intentando acceder de forma no autorizada.
5. Centralización y control de la política de seguridad. Debido a su situación privilegiada, el cortafuego puede concentrar toda la aplicación de las normas de comportamiento reguladas en la política de seguridad, relevando al resto de entidades comunicantes de la red de la obligación de cumplirlas. En teoría, ello establece garantías de que tal política será cumplida, con lo que mejoraría la fiabilidad. No obstante, esto representa un empobrecimiento de las posibi-

lidades que ofrece el establecimiento de reglas de comportamiento y sólo sirve para solucionar casos muy simples en redes muy jerarquizadas.

6. Funciones no estrictamente relacionadas con la seguridad. Puede disponer de dispositivos de traducción o traslación de direcciones locales a direcciones Internet y realizar tareas que permitan la implantación de redes privadas virtuales. Al mismo tiempo, puede servir, como ya se ha dicho, de «escaparate», de «embajador» externo de la red protegida.

Por contra, hay tareas que el cortafuego no está capacitado para realizar que representan limitaciones notables en cuanto a la protección que puede proporcionar. Entre éstas podríamos destacar:

1. Ataques desde el interior de la red. Debido a su concepción de puerta fortificada, no puede impedir que los usuarios internos de la red hagan un mal uso de los recursos e incluso divulguen ante terceros contenidos nocivos.
2. Tráfico que puentea al cortafuego. Como los usuarios maliciosos son muy habilidosos, el cortafuego no es capaz de impedir que aquellos que estén dotados de un módem u otro dispositivo equivalente puedan establecer conexiones con otros usuarios externos o con sistemas proveedores de Internet (ISPs) de forma no autorizada.
3. Ataques mediante virus o programas maliciosos. Como a lo único que se limita es a regular el paso a través de la puerta de entrada, no tiene capacidad para analizar si los contenidos de lo que deja pasar son dañinos o perjudiciales. Intentar analizar y depurar todo lo que atraviesa el cortafuego representaría una carga de trabajo y un retraso en el tráfico que harían impracticable una solución de este tipo.
4. Restricción de acceso a servicios deseables. Esta es, quizás, la limitación más notable. Dada su estrategia de «cortar por lo sano», puede bloquear el acceso a algunos servicios que en determinado momento pueden ser de utilidad a los usuarios de la red. Será muy difícil encontrar un equilibrio entre limitaciones de seguridad y necesidades de los usuarios.

Estas limitaciones se pueden solucionar adoptando un modelo diferente, basado en la securización de todos y cada uno de los servidores y sistemas de la red, adoptando una política de seguridad que respete la independencia y las particularidades de cada uno de ellos e implantando los correspondientes mecanismos y protocolos de seguridad.

DIFERENTES CONFIGURACIONES DE CORTAFUEGOS

En todo este asunto de los cortafuegos hay una excesiva abundancia de nombres y sobrenombres, y sólo un reducido número de conceptos relevantes (mucho ruido y pocas nueces). Aun a riesgo de perder detalles, trataremos aquí de simplificar la descripción de los distintos elementos y configuraciones presentes en este tipo de sistemas de protección de redes. Además, en esta ocasión no nos esforzaremos en buscar nombre en español para las distintas configuraciones, por lo que reflejaremos directamente la denominación con la que se conocen en inglés. El lector interesado puede encontrar descripciones detalladas y completas tanto en textos especializados como a través de páginas web.

A la hora de diseñar las reglas mediante las cuales se prohíbe o se permite una determinada actuación se puede seguir una de las dos políticas siguientes:

- Lo que no está expresamente prohibido está permitido.
- Lo que no esté expresamente permitido está prohibido.

Los dos componentes básicos que están presentes en las distintas configuraciones de cortafuegos (aislados o de forma conjunta) son:

- Encaminadores (*routers*) con filtrado de paquetes.
- Servidores actuando como pasarelas a Nivel de Aplicación.

En la Figura 2.5 se representa una configuración de cortafuegos utilizando el primero de estos dos componentes, el cual permite el paso a ciertos paquetes mientras bloquea a otros, desde y hacia la red interna. Las decisiones de filtrado están basadas en el análisis de algunos de los campos presentes en la estructura del paquete. Si se trata de paquetes IP, se basan en los campos de la cabecera, como son la dirección IP origen y la dirección IP destino. Además, en algunos casos, el filtrado se realiza también basándose en la cabecera a nivel de transporte, analizando el puerto TCP o UDP origen y el puerto de destino, los cuales sirven, a su vez, para conocer el tipo de aplicación de que se trata (Telnet, SMTP, etc.).

A un cortafuegos configurado conforme a esta estructura (la representada en la Figura 2.5) se le suele denominar *screening router firewall*, ya que el elemento básico en que se apoya es un encaminador con inspección de direcciones para filtrado (*screening router*). Es una de las configuraciones más sencillas y económicas que pueden plantearse.

El segundo componente que puede estar presente (Figura 2.6) es un Servidor (*host*) que actúa como pasarela, realizando el control de paso en el Nivel de Aplicación. Este tipo de sistemas se conocen también como *servidor proxy* (contracción de

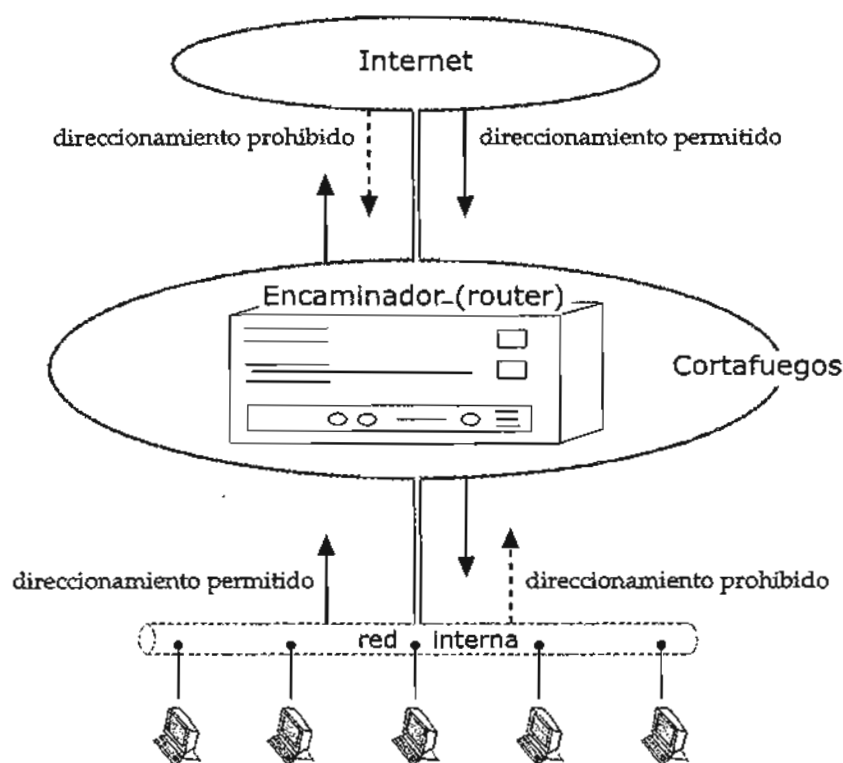


Figura 2.5. Filtrado de paquetes mediante un encaminador (*router*).

la palabra inglesa *procuracy*), que enfatiza el carácter que tiene de representante o sustituto de otro. El servidor proxy se coloca entre un usuario situado en el interior de la red protegida y un servidor situado en el exterior (o viceversa) y actúa a modo de repetidor, reenviando el tráfico recibido desde cada uno de los extremos de la comunicación, de tal manera que ofrece a ambos interlocutores la ilusión de estar conectados directamente (extremo a extremo). Es decir, el proxy presenta al usuario la ilusión de estar tratando directamente con el servidor real.

Por trabajar a nivel de aplicaciones, el proxy está en condiciones de interpretar el protocolo y realizar, de este modo, operaciones de filtrado mucho más refinadas que las que son posibles utilizando el filtrado de direcciones a nivel de red o de transporte. Así mismo, para la autenticación de usuarios puede utilizar mecanismos de seguridad más robustos y fiables.

A un cortafuegos configurado conforme a la estructura representada en la Figura 2.6 se le puede denominar *dual-homed host firewall*, debido al hecho de que el servidor presenta doble puerto, con dos interfaces, una para la conexión hacia el interior de la red y otra hacia el exterior.

El servidor (*host*) representado en la Figura 2.6 puede ser también un servidor directamente gestionado por el administrador de la red, en el cual se soporte principalmente la política de seguridad, dotado de un sistema operativo robusto y depurado, que, además, requiera de sus interlocutores una autenticación fiable y gobierne las funciones proxy de intermediación. A un servidor de este tipo se le denomina *bastion host*, que podríamos traducir por algo así como *servidor fortificado*.

Además de las dos configuraciones representadas en las figuras citadas, cada una de las cuales está basada en uno de los dos componentes básicos, el encaminador con filtrado (*screening router*) y el servidor proxy, existen otras configuraciones que se obtienen mediante la actuación simultánea y combinada de estos dos componentes y de otros elementos. Las configuraciones y arquitecturas existentes son muy variadas y numerosas y están ligadas muchas veces a soluciones y denominaciones comerciales. Por cuestión de oportunidad, no vamos a entrar aquí a describirlas todas ni a comentar sus ventajas e inconvenientes. De entre estas configuraciones solamente citaremos las más significativas. Cabe destacar:

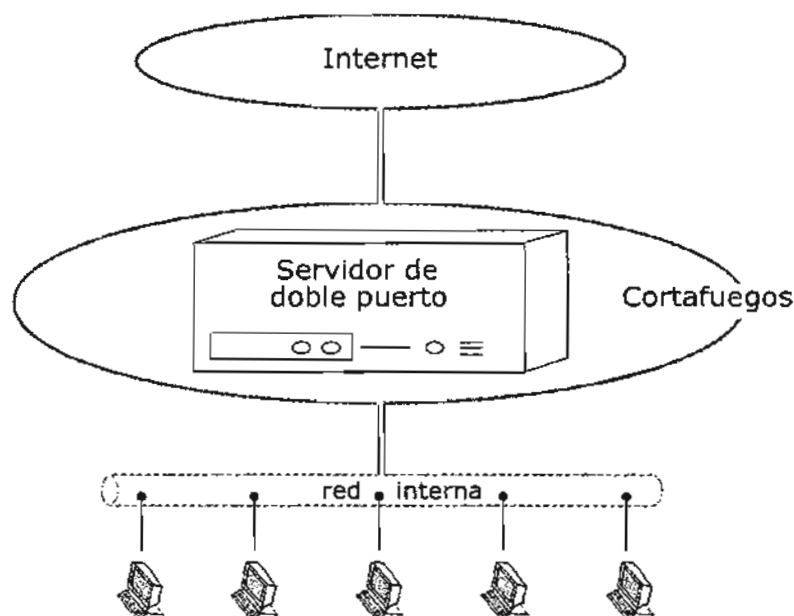


Figura 2.6. Servidor actuando como pasarela.

- *Screened host firewall*. Es un cortafuego formado por un servidor proxy, implementado con un *bastion host*, y un filtro, implementado con un *screening router*. De esta configuración se presentan, a su vez, dos variantes:
 - *Dual-homed bastion host*, cuando el servidor posee dos interfaces: una para conectarse al exterior y otra para conectarse a la red a proteger.
 - *Single-homed bastion host*, cuando el servidor posee solamente una interfaz con la red.
- *Screened subnet firewall*. Es un cortafuego formado por un servidor proxy, implementado con un *bastion host*, y dos filtros, implementados con dos *screening routers*, uno de ellos externo (del lado de Internet) y el otro interno (del lado de la red a proteger).

LIMITACIONES DE LOS CORTAFUEGOS

En realidad, esta forma de protección de una red privada, más que un cortafuego es, como decíamos antes, una muralla con una sola puerta de entrada bien guarnecida. Bajo este paradigma, las comunicaciones que se protegen son entre el cortafuego y el exterior de la red, ya que éste actuará como supervisor o intermediario de todas las comunicaciones que mantengan con el exterior los sistemas o usuarios pertenecientes a la red privada que se pretende defender.

Debido a estas características, los servicios de seguridad que se pueden implantar usando cortafuegos (que a veces son simples filtros de direcciones de red) son muy poco versátiles y muy difíciles de compaginar con la configuración y diseño de dominios de seguridad avanzados como los que cabe esperar que, de forma creciente, demanden los ciudadanos. Por la protección que aportan hay que pagar el precio de la falta de flexibilidad que representan.

No obstante, a pesar de las notables limitaciones de la seguridad que puede ofrecer un cortafuegos, estas soluciones suelen ser muy bien valoradas por los potenciales beneficiarios de técnicas de seguridad en redes, valoración que muchas veces está motivada más por la expectativa favorable que el nombre despierta que por una ajustada valoración de sus posibilidades reales. Además, el cortafuegos deja sin securizar las comunicaciones internas a la red protegida.

Por todo ello, para garantizar la implantación de servicios avanzados de seguridad sigue siendo mejor solución implantar protocolos de seguridad en todas y cada una de las entidades comunicantes que participan en un dominio de seguridad. Caso de que esa seguridad quiera reforzarse protegiendo una determinada red privada mediante un cortafuego, será necesario diseñar para éste una política de seguridad compatible con la securización particularizada de cada uno de los sistemas internos de la red que participen en comunicaciones seguras.

2.7. NORMAS Y ORGANISMOS DE NORMALIZACIÓN

En el mundo de las telecomunicaciones, desde sus inicios, se detectó la necesidad de establecer normas que regulasen el comportamiento de los sistemas para facilitar así la interoperabilidad entre distintos países. Con el advenimiento de la Telemática a finales de los setenta y principios de los ochenta, esta necesidad se hizo aún más visible. Para que sistemas telemáticos ubicados en distintas organizaciones puedan

intercambiar datos de forma eficaz es necesario que operen bajo protocolos previamente concertados.

Los protocolos y servicios de seguridad son parte integrante de las redes telemáticas y los estándares que regulan su comportamiento están siendo generados por los mismos organismos internacionales de normalización que se venían ocupando de estas redes.

A lo largo de los distintos capítulos que componen este libro se da una gran importancia a todo lo relacionado con las normas que rigen esos comportamientos. Ya en los primeros apartados del Capítulo 1 nos hicimos eco de la norma «matriz» de la Seguridad en Redes, la ISO/IEC 7498-2, y en lo sucesivo, cuando se aborden servicios y protocolos que hayan sido objeto de estandarización, trataremos de resumir los principales elementos de esas normas con el fin de ayudar al lector a entender lo más significativo de lo que representan, facilitando así su posterior análisis pormenorizado (acudiendo a las fuentes originales) cuando las circunstancias así lo requieran.

Algunas de estas normas sobre Seguridad en Redes en las que nos apoyaremos en lo sucesivo se encuentran en la actualidad bastante estabilizadas, debido a lo cual la interpretación y comentarios que sobre ellas hagamos pueden considerarse válidos durante un periodo de tiempo considerable. Otras, en cambio, aunque desde un punto de vista genérico y conceptual se encuentran también bastante estabilizadas, a nivel de detalles van siendo actualizadas con cierta frecuencia, por lo que el lector juicioso deberá comprobar si la referencia que aquí hacemos de esa norma se corresponde con la versión vigente en el momento de realizar la consulta. Las normas de ITU-T e ISO suelen conservar el mismo número de referencia durante bastante tiempo, pero cuando se trata de RFCs es habitual que una nueva versión reciba un nuevo número identificativo. Otras (pocas) normas que más adelante citaremos se encuentran, en el momento de redactar los comentarios a ellas referidos, en un estado bastante emergente, por lo que es de esperar que las formulaciones allí realizadas sufran, con el transcurso del tiempo, modificaciones significativas.

Esta necesidad de permanente actualización de las referencias normativas en las que es necesario apoyarse es el tributo que hay que pagar por dedicarse al estudio de ciertas tecnologías que, como la Seguridad en Redes Telemáticas, tienen un futuro esperanzador pero que, a causa precisamente del crecimiento que ello conlleva, se hallan sometidas a cambios significativos que exigen una permanente atención y actualización.

En la sección de bibliografía se presenta un recopilatorio de las normas de seguridad más significativas en las que nos hemos apoyado y en los apartados que siguen haremos un breve comentario acerca de los principales organismos internacionales de normalización en los que éstas han sido desarrolladas. Éstos son:

- ITU-T.
- ISO.
- IETF-IESG.
- ETSI.
- CEN-CENELEC.
- NIST.

ISO E ITU-T

ISO es el nombre*, no el acrónimo, de la Organización Internacional de Normalización (*International Organization for Standardization*) que fue fundada en 1946 y está integrada por más de 75 organismos generadores de normas nacionales. En España, el miembro correspondiente es AENOR, Asociación Española de Normalización y Certificación. ISO es la organización de normalización más importante a nivel internacional y se ocupa de facilitar el intercambio de bienes y servicios mediante la generación de estándares en ámbitos tan dispares como pueden ser el paso de rosca de los tornillos o los féretros para el transporte de cadáveres.

En el ámbito de la Telemática, su importancia radica en la creación del comité ISO/IEC JTC1 que desarrolló el famoso Modelo OSI para la Interconexión de Sistemas Abiertos (ISO/IEC 7498-1) y todo el conjunto de normas de él derivadas. En lo tocante a la Seguridad en Redes, ISO ha tenido un protagonismo importantísimo a partir de la publicación en 1988 de la segunda parte del Modelo OSI: la ISO/IEC 7498-2 *Security Architecture*, antes citada.

Las normas de ISO son elaboradas a través de comités técnicos y sometidas a diversas fases de votación hasta su aprobación final. Debido a la presencia de múltiples países, los procesos de aprobación son algo lentos, pero tienen la garantía de ser bastante democráticos y representativos por tener en cuenta los intereses de todos los países que la componen, incluso de aquellos con economías más débiles (la cuota de suscripción está en relación con el PIB de cada uno de sus miembros).

Antes de referirnos a las normas de ISO en las que nos apoyaremos en los capítulos sucesivos, comentaremos brevemente la composición y funciones de otro organismo internacional con el que ha colaborado estrechamente: ITU-T.

El ITU-T es el Sector de Normalización de las Telecomunicaciones de la UIT (Unión Internacional de Telecomunicaciones, ITU conforme a sus siglas en inglés), que es el organismo especializado de las Naciones Unidas que se encarga del estudio de aspectos técnicos y de explotación en el campo de las telecomunicaciones, y publica regularmente *Recomendaciones* con miras a la normalización a escala mundial. En lo que a nosotros nos interesa, desde 1993 ITU-T es el heredero del CCITT (Comité Consultivo Internacional de Telefonía y Telegrafía), que se venía ocupando desde muy antiguo (sus orígenes se remontan a 1865) de la normalización de esos dos servicios básicos de telecomunicación, ancestros de la Telemática. El CCITT publicaba sus *Recomendaciones* en varios idiomas de ámbito mundial, entre ellos el español (procedimiento que sigue utilizando el ITU-T).

En el CCITT participaban, entre otros organismos, las administraciones nacionales que explotaban en régimen de monopolio (o de organismo gubernamental) los servicios de Correos, Teléfonos y Telégrafos (conocidas como PTTs, conforme a sus siglas en inglés). Con el advenimiento de las comunicaciones de datos entre computadores, estas administraciones pretendieron proveer, también en régimen de exclusividad, estos nuevos servicios, por lo que el CCITT pasó a ocuparse de normalizarlos, iniciando una nueva serie de recomendaciones, conocida como *Serie X*, no porque contuviese escenas que pudiesen herir la sensibilidad de sus lectores (como las películas clasificadas con esa misma letra), sino porque todas esas normas se identifican mediante una X seguida de un punto y un número de serie.

* Se adoptó la palabra griega *iso*, que es utilizada en muchos idiomas (entre ellos el español) como elemento compositivo que entra en la formación de algunas palabras con el significado de *igual* o denotando *uniformidad* o *semejanza*.

El CCITT acostumbraba, además, a actualizar sus normas cada cuatro años editando lo que se denominaba «un libro», aunque en realidad estaba compuesto por una colección de éstos. Los últimos en ser editados fueron: en 1980, el *libro amarillo* (las cubiertas de los ejemplares eran de ese color), en el que se incluía una primera versión de X.25 (con los tres primeros niveles de la arquitectura OSI); en 1984, el *libro rojo*, que incluyó la primera versión del servicio de correo electrónico X.400; y en 1988, el *libro azul*, que incluía (además de una segunda versión de X.400) la primera versión del Servicio de Directorio X.500 y de la Recomendación X.509 (parte de la serie X.500), donde se definió por primera vez el Certificado de la Clave Pública al que aquí hacemos referencia una y mil veces en los distintos capítulos.

Durante estos primeros años (hasta 1988) tanto ISO como el CCITT habían estado trabajando sobre las mismas cosas y generando normas muy parecidas, pero no iguales, lo que representaba el consiguiente desasosiego por parte de quienes tenían que estar al tanto de sus formulaciones. A partir de 1988 esa situación cambió, unificando sus respectivas versiones tanto de correo como del Servicio de Directorio (la X.509 es la misma que la ISO 9598-8, aunque es más conocida por el primero de estos dos identificadores), continuando en lo sucesivo una colaboración muy estrecha entre ISO e ITU-T que ha dado como resultado el desarrollo de muchas normas conjuntas en las que nosotros nos apoyaremos en el desarrollo de los sucesivos capítulos. La norma «matriz» ISO/IEC 7498-2 a la que ya nos hemos referido se conoce en el ámbito de ITU-T como Recomendación X.800, *Arquitectura de seguridad de interconexión de sistemas abiertos para aplicaciones del CCITT*.

EL IETF Y LAS RFCs

En la década de los setenta, dentro de la red ARPANET se estuvieron desarrollando los primitivos protocolos de comunicación de paquetes que se consolidaron a principios-mediados de los ochenta con la aprobación, dentro de esa red, de los protocolos IP y TCP, además de otros protocolos del Nivel de Aplicación. La experiencia adquirida en estos trabajos fue determinante a la hora de definir el Modelo OSI a finales de los setenta y principios de los ochenta. En el entorno de ARPANET se justificaba la permanencia en el uso de estos estándares «de facto» (para distinguirlos de los estándares «de jure» aprobados en los organismos oficiales) por el hecho de la falta de madurez de los protocolos que se estaban desarrollando en el marco de ISO y del CCITT, y además se argumentaba que a medida que esa maduración y estabilización se fuese produciendo se iría promoviendo la «migración» de los protocolos de la pila TCP/IP a los correspondientes del modelo OSI. A finales de los ochenta (coincidiendo casualmente con la crisis del modelo multipolar vigente desde finales de la Segunda Guerra Mundial) se deja de hablar de esta migración hacia protocolos «de jure» y se inicia un viaje en sentido contrario tendente a desprestigiar los organismos oficiales internacionales y las formulaciones de ellos procedentes.

La cosa se estabiliza (más o menos) a mediados de los noventa con la creación de la *Internet Society*, que si bien no depende de ningún organismo oficial internacional de titularidad pública, sí está abierta a la cooperación de diseñadores de todo el mundo (aunque manteniendo el centro de gravedad en Estados Unidos). En este contexto operan el IETF (*Internet Engineering Task Force*) y el IESG (*Internet Engineering Steering Group*), que se encargan de los procesos de generación y aprobación de las normas desarrolladas en su seno.

Estas normas se conocen mediante el acrónimo RFC y un identificador numérico. RFC son las iniciales del muy relajado nombre de «Request For Comments», que trata de dar una imagen de lo abierto que es el proceso de generación de estándares para su evolución desde simples propuestas hasta estándares aprobados. (En la práctica las cosas están bastante condicionadas por los grupos de empresas que, obviamente, tratan de arrimar el ascua a su sardina.)

Dentro del IETF existen una serie de áreas y grupos de trabajo que abordan distintos temas organizativos y de desarrollo técnico. En lo que se refiere a la Seguridad en Redes, existe una actitud muy abierta por parte de estos grupos en cuanto a la colaboración y aceptación de formulaciones procedentes de los restantes organismos internacionales. En los capítulos dedicados a certificación y a infraestructuras de clave pública nos apoyaremos frecuentemente en las propuestas y RFCs generados por el grupo de trabajo *Public-Key Infrastructure (X.509)*, conocido como *PKIX Working Group*. Este grupo, como puede apreciarse en su mismo nombre, se basa en las formulaciones de la X.509 (de ISO e ITU-T), cosa que nos facilita la compatibilidad de formulaciones teóricas y prácticas entre las distintas normas en las que nosotros nos apoyamos.

ETSI Y OTROS

ETSI es el acrónimo de *European Telecommunication Standard Institute*, que es una organización sin ánimo de lucro promovida por las autoridades de la Unión Europea con el objetivo de la definición de normas y perfiles para su utilización en el ámbito europeo (y de aquellos países que deseen aceptarlos). Han trabajado sobre un amplio abanico de normas, de las cuales nosotros utilizaremos de forma detenida las correspondientes a la Firma Electrónica que abordamos en el Capítulo 8. Es de esperar que a medida que evolucione la normativa leg59al europea sobre firma y sobre otros servicios telemáticos seguros (recogiendo referencias expresas a estándares y perfiles técnicos) aumente también el protagonismo del ETSI en la generación de esas normas.

El CEN (Comité Europeo de Normalización) y el CENELEC (Comité Europeo de Normalización Electrotécnica) son organismos europeos bastante antiguos con una trayectoria no excesivamente brillante. En lo que a nosotros se refiere, solamente utilizaremos de pasada las normas que se han desarrollado en CEN-CENELEC como complemento a las propuestas de ETSI sobre Firma Electrónica.

En otro ámbito totalmente distinto, cabe destacar al NIST (*National Institute of Standards and Technology*), que es el organismo de Estados Unidos encargado de la homologación de normas y tecnologías. Nosotros haremos referencia a los estándares generados por el NIST, principalmente en los capítulos dedicados a criptografía, cuando se describan los distintos algoritmos en que se apoyan los mecanismos criptográficos.

CAPÍTULO

3

FUNDAMENTOS TEÓRICOS DE LA CRIPTOGRAFÍA

La criptografía moderna se apoya en una serie de estudios teóricos que permiten modelar matemáticamente el problema para obtener conclusiones acerca de la robustez de los distintos criptosistemas. En la descripción que sigue se ha procurado referir de forma sucinta, sin demostraciones ni especulaciones excesivas, los conceptos en que se basan los criptosistemas que más adelante se describen y que son, a su vez, los más frecuentemente utilizados en los mecanismos de seguridad en que se apoyan los servicios telemáticos.

Los objetivos perseguidos en estas notas se limitan, pues, a proporcionar a las personas interesadas en la Seguridad en Redes las referencias teóricas imprescindibles para entender adecuadamente los principios en que se basan estos sistemas criptográficos. Se ha procurado, para ajustarse a este presupuesto, referir sólo lo más relevante, evitando tanto el pasarse como el quedarse corto (al igual que en el juego de las siete y media).

El lector interesado puede encontrar bibliografía especializada donde se aborda en profundidad esta base matemática. En [FGHM00], [Rami98^{*}] y [Schn96] puede encontrarse una descripción más detallada de los conceptos utilizados en este capítulo y se hace, a su vez, referencia a fuentes más especializadas sobre el tema. Debido a la novedad e importancia que representa el criptosistema Rijndael, la aritmética en que se basa este algoritmo se explica aquí con algo más de detalle que en las referencias citadas.

3.1. TEORÍA DE LA INFORMACIÓN. CRIPTOSISTEMAS DE SECRETO PERFECTO

El paso de la criptografía clásica a la criptografía moderna viene dado, entre otros, por dos factores fundamentales: la plasmación de los algoritmos criptográficos en progra-

^{*} Gran parte de la información contenida en ese texto puede encontrarse también en: *Seguridad Informática. Libro electrónico en diapositivas de libre distribución en Internet*. Se trata de un conjunto de diapositivas regularmente actualizadas que pueden ser descargadas, entre otros sitios, desde la página personal de su autor, Jorge Ramió.

mas de computador y el tratamiento de estas técnicas desde el punto de vista de una disciplina científica.

Un primer aspecto de ese tratamiento teórico y científico consiste en analizar la naturaleza misma de la información en lo que se refiere a los cambios que se producen cuando se verifica la transformación que convierte un mensaje en claro en un mensaje cifrado.

INFORMACIÓN E INCERTIDUMBRE

Cuando se intenta cuantificar la información que aporta la recepción de un mensaje acerca de la ocurrencia de un suceso, nos encontramos con que esa posible medida debería estar relacionada con el grado de novedad o sorpresa que, para el receptor del mensaje, representa su conocimiento.

Antes de que el suceso haya ocurrido, el observador tan sólo puede especular sobre las expectativas acerca de que el suceso ocurra. Ello significa que el suceso ha de ser *verosímil*, lo que puede ser cuantificado mediante la *probabilidad* de su ocurrencia, que es una medida de la *incertidumbre* sobre la posibilidad de que el suceso ocurra. Lo más verosímil que puede ser un suceso es que haya certeza acerca de su ocurrencia (probabilidad de valor uno) y lo menos es que éste sea imposible (probabilidad cero). Para todos los demás casos, el grado de verosimilitud estará comprendido entre ambos valores y podría medirse por la probabilidad de su aparición.

Por contra, después de que el suceso haya ocurrido, el observador recibirá una *sorpresa*, sorpresa que será muy grande si la probabilidad de ocurrencia del suceso era muy pequeña. Si por ejemplo pensamos en algo que es casi imposible pero que, nunca se sabe, a lo mejor a la caída de la tarde va y sucede, pues nuestra sorpresa sería grandísima. Si tenemos certeza acerca del suceso (por ejemplo, «mañana será otro día») podremos cuantificar como cero la sorpresa, mientras que la ocurrencia de un suceso imposible (probabilidad cero) nos arrojaría una sorpresa infinita. Infinito y cero: esos son los límites de la sorpresa, y como medida de su ocurrencia se define la *cantidad de información*.

En 1948 Claude Shannon [Shan48] en su artículo «A Mathematical Theory of Communication» aportó las bases matemáticas de la teoría de la comunicación, que tan provechosa aplicación ha tenido en el mundo de la Telecomunicación. Un año después con su «Communication Theory of Secrecy Systems» [Shan49] formuló su aplicación a los sistemas criptográficos. Estos estudios proporcionan una base teórica importante para evaluar la seguridad de los *criptosistemas de clave secreta*, aunque en la práctica la robustez de estos sistemas depende en mucho de otras características inherentes al algoritmo, que escapan a las formulaciones genéricas de la Teoría de la Información.

Si $m_1, m_2, \dots, m_n$ es un conjunto de mensajes o estados (posibles sucesos) de una variable aleatoria y $p_1, p_2, \dots, p_n$ son sus respectivas probabilidades de ocurrencia

($\sum_{i=1}^n p_i = 1$), se define *cantidad de información* de un estado m_i como:

$$c_i = -\log_a p_i \text{ unidades de información} \quad (1)$$

Si la probabilidad de un suceso (estado) es 1, es decir, se trata de un hecho sobre el que hay total certeza, la cantidad de información será cero. En el caso contrario (probabilidad cero, suceso imposible) valdrá infinito. En los restantes casos será un número positivo comprendido entre cero e infinito.

La elección de a , base del logaritmo, es una cuestión de «escala», esto es, de determinación de las unidades de información. Si elige base 2, la cantidad de información que aporta la aparición de un estado en un fenómeno binario y equiprobable será la unidad:

$$\text{Probabilidad de un estado, } p_i = \frac{1}{2} \Rightarrow c = -\log_2 \frac{1}{2} = 1$$

En este caso, la unidad de información es el **bit**.

Entropía

Dada una variable aleatoria del tipo citado en párrafos anteriores, se define la **entropía** de dicha variable como:

$$H(m) = -\sum_{i=1}^n p_i \log_2 p_i \quad (2)$$

Es decir, es el valor medio ponderado de la cantidad de información de los diversos estados, y representa, por tanto, una medida de la incertidumbre media acerca de la variable aleatoria. La entropía se mide, por tanto, en bits/estado o en bits/símbolo emitido.

Si los n estados posibles son equiprobables, la posibilidad de ocurrencia de cada uno de ellos es $\frac{1}{n}$ y la entropía:

$$H(m) = \log_2 n \text{ bits/estado} \quad (3)$$

Por ejemplo, sea una variable de ocho estados o mensajes con valores comprendidos entre el 000 y el 111. Si pensamos en uno de los posibles mensajes, ¿cuántas preguntas tendría que hacernos un interlocutor para que mediante respuestas de *sí* o *no* pueda averiguar la combinación elegida? En realidad, puede interrogarnos de muy diversas formas. Si se realizan al azar, preguntando por un estado concreto, puede que se acierte a la primera, o puede que no se obtenga respuesta positiva hasta la octava pregunta. Pero si realiza una primera pregunta acerca de si es cero o uno el primer dígito, seguida de otras dos preguntas más acerca del segundo y el tercer dígito, al final podrá averiguar con certeza el mensaje de que se trate. Obsérvese que son tres las preguntas necesarias y que 3 es el $\log_2 8$ y 3 son los dígitos binarios necesarios para representar todos los estados de esa variable.

Codificador Óptimo

Puede demostrarse que la entropía $H(m)$, definida en (2), es máxima cuando todos los estados son equiprobables (o sea, cuando $H(m) = \log_2 n$) y que en este caso representa el *número de bits necesarios para representar todos los estados*. En este caso, la codificación sería óptima.

En general, la entropía representa el número medio de bits que serían necesarios para representar todos sus estados mediante dígitos binarios. Por definición, un **codi-**

ficador óptimo sería aquel que utiliza el mínimo número de bits para codificar los mensajes. Para ello debe atribuir menor número de bits a los mensajes más frecuentes y mayor número de bits a aquellos de menor probabilidad de aparición. Esta condición, plasmable de forma cuantitativamente exacta usando métodos matemáticos, fue intuita por Morse cuando asignó valores binarios (puntos y rayas) en su famoso código, asignando cadenas más cortas a las letras más frecuentes y reservando códigos más largos para los símbolos menos frecuentes en inglés.

REDUNDANCIA DE UN LENGUAJE Y CRIPTOANÁLISIS

Se define además la **entropía condicional** cuando existe alguna dependencia entre dos variables aleatorias, de modo que el conocimiento de una de ellas disminuye la incertidumbre sobre la otra. Así, dado el criptograma c , la entropía condicional determina la probabilidad de que éste se corresponda con determinado mensaje en claro.

Ello está relacionado con la redundancia de los lenguajes naturales, en los cuales determinadas letras son mucho más frecuentes que otras y en los que, en algunos casos, la aparición de una letra disminuye en mucho la incertidumbre sobre la que le sigue: existen digramas, trigramas, etc., de frecuente aparición. Este tipo de dependencias han sido usadas ampliamente y con éxito por los criptoanalistas para descifrar los mensajes.

Si en el idioma español la aparición de letras fuese equiprobable, considerando 27 letras (no incluidas como tal ni la *ch* ni la *ll*) el número de bits necesarios para representar cada una de ellas sería:

$$R = \log_2 27 = 4,8 \text{ bits/letra}$$

Esta magnitud, que coincide con la entropía máxima teórica, se denota por R , y se denomina **ratio absoluta del lenguaje**. En inglés sería $\log_2 26 = 4,7$ bits/letra.

Debido a que los lenguajes naturales son redundantes, la entropía de un mensaje de longitud N se aleja en mucho de la entropía máxima (caso de equiprobabilidad entre los símbolos). Se define la **ratio de un lenguaje** como la entropía real del mensaje. Por tanto, este valor, r , representa la cantidad de información media aportada por cada carácter, es decir, el número medio de bits/letra que serían necesarios para representar el mensaje.

Estadísticamente, se estima que el valor de r en español está comprendido entre 1,2 y 1,5. En inglés, se estima entre 1,3 y 1,5.

Se define como **redundancia de un mensaje**, y se representa por D , a la expresión:

$$D = R - r \quad (4)$$

Si además se utilizan códigos en sí mismos redundantes, como en el caso del código ASCII, que utiliza ocho bits por carácter, la redundancia para mensajes en español, compuestos solamente por letras, estaría comprendida entre 6,8 bits/letra y 6,5 bits/letra, información innecesaria que redundaría en facilidades para el criptoanalista a la hora de romper el secreto de un criptosistema.

De la expresión (4) se deduce que cuanto menor es la entropía, r , de un mensaje, tanto mayor es la redundancia, D , y más bazas tiene a su favor un criptoanalista que desee conocer el mensaje en claro a partir del criptograma.

SECRETO PERFECTO

Dado un criptograma, el sistema sería vulnerable si un criptoanalista es capaz de averiguar el mensaje en claro, o la clave, o ambos. Shannon se apoyó en el hecho de que la recepción de un criptograma aporta al criptoanalista ciertas informaciones probabilísticas acerca del mensaje en claro del que procede. Teóricamente, se mediría el **secreto de un criptosistema** como la incertidumbre acerca del mensaje en claro m , conocido el criptograma c . En general, el tamaño del espacio de claves da una medida de la entropía del criptosistema y, por tanto, de su seguridad. Cuanto mayor sea el número de bits de la clave tanto mayor será aquella.

Un criptosistema es considerado de **secreto perfecto** si el conocimiento del mensaje cifrado c *no aporta* al criptoanalista *información alguna* acerca del mensaje en claro m (excepto, acaso, su longitud).

Pongamos un ejemplo con connotaciones de espionajes y cosas parecidas: Dado un mensaje m_i de 34 caracteres que sea «la entrega del maletín a las 10:30», se cifra con un algoritmo y una clave determinada tales que se genera un criptograma c_j del mismo tamaño que el mensaje en claro. En un caso convencional, teniendo en cuenta las imperfecciones y deficiencias del criptosistema, al recibir el criptograma c_j , el criptoanalista es capaz de deducir cosas acerca del mensaje m_i , siendo unas soluciones más probables que otras. Si tuviese las herramientas y los conocimientos adecuados, podría incluso averiguar m_i . Sin embargo, si el criptosistema fuese de secreto perfecto, el haber recibido c_j sitúa al criptoanalista en tal situación que la probabilidad de que el mensaje sea «la entrega del maletín a las 10:30» es exactamente la misma que la probabilidad de que sea «pinto pinto gorgorito chinchin pun», o cualquier otra combinación de 34 caracteres. Es decir, que después de la captura del criptograma, sabe tanto del mensaje original como antes de capturarlo (excepto que tiene 34 caracteres).

En términos más rigurosos, cabe decir que, dado un espacio de posibles mensajes, de claves y de criptogramas, el sistema sería de **secreto perfecto** si la probabilidad de que el criptograma recibido, c_j , sea el resultado de haber cifrado el mensaje m_i , es igual a la probabilidad de haber recibido precisamente ese mensaje m_i . En otras palabras: las dudas del criptoanalista acerca de cuál será el mensaje transmitido no se ven alteradas por el hecho de haber recibido c_j . Ello significa que es *imposible* deducir m_i a partir de c_j .

Para que esto sea posible es necesario que el número de claves sea, al menos, tan grande como el número de posibles mensajes.

UN CRIPTOSISTEMA DE SECRETO PERFECTO

Un caso muy llamativo y curioso es el de un criptosistema de sencillísimo algoritmo y de aún más fácil implementación, pero que sin embargo puede ser calificado como de perfecto en cuanto al secreto que conlleva. Parece ser que era el usado por los espías soviéticos y que con los actuales conocimientos y recursos computacionales no es posible averiguar en qué consistía un mensaje secreto una vez captado el criptograma. Pero hay más: si el sistema es teóricamente perfecto, estamos en condiciones de afirmar que las generaciones futuras, con máquinas tan perfectas y dicharacheras como las que nos pintan los especialistas en ciencia-ficción, no serán capaces tampoco de averiguar el contenido de esos mensajes lanzados en su día por los espías en cuestión.

Más llamativo es, en una primera impresión, que siendo un sistema bueno, bonito y barato no sea de utilidad en la criptografía moderna y mucho menos aún en los

mecanismos usados en Seguridad en Redes Telemáticas. Veamos en qué consiste y el porqué de todas estas afirmaciones.

Podríamos denominarlo como criptosistema de *clave aleatoria y continua de un solo uso* (en inglés se conocen bajo la denominación **One-time pad**). Dado un mensaje compuesto sólo por letras (podemos considerar suprimidos los espacios en blanco) se trataría de obtener una clave mediante un procedimiento de azar puro (por ejemplo sacando letras de un bombo como el de la lotería, donde estuviesen contenidas bolas con todas las letras del abecedario (en español, 27 no considerando ni *ch* ni *ll*). Una vez obtenida la letra sería devuelta al bombo para igualar la probabilidad de salida en la siguiente extracción. Esta operación podría también realizarse utilizando varios dados (los hay de más de seis caras). Así se obtendría al azar una clave de igual tamaño que el mensaje original.

A continuación se procedería a sumar ambos textos letra a letra, módulo 27. Así se procede en el ejemplo que se muestra en la Figura 3.1: la suma de la quinta letra del abecedario (la *e*) con la vigésima primera (la *t*) da como resultado la vigésima sexta letra, que corresponde a la *y*; y así sucesivamente.

La interceptación del mensaje cifrado no aporta ninguna información acerca del mensaje en claro. El receptor del criptograma que no esté en posesión de la clave tiene tanta información sobre el mensaje en claro como tenía antes de recibirlo: sólo la probabilidad de que sea uno concreto entre todos los posibles de igual número de letras. Por ello se puede afirmar que el criptograma es de secreto perfecto.

Si en lugar de un método de azar puro se hubiese utilizado uno de los muchos programas de ordenador que proporcionan una solución semialeatoria, la robustez del criptosistema sería grandísima, pero no podría afirmarse que alcanza la cota de la perfección. Más débil aún sería si se usa el texto de un libro predefinido.

Para su hipotético uso en redes telemáticas, el problema que presenta es la generación y distribución de claves. Parece claro que los dos interlocutores tienen que haber concertado a priori cuáles van a ser las sucesivas claves que van a utilizar, ya que para alcanzar la fortaleza referida es imprescindible que las claves sean de *usar y tirar*; caso contrario, el criptoanalista encontraría las posibles redundancias que abrirían el camino al descubrimiento del mensaje original. Por ello, el procedimiento consistente en que uno de ellos genere la clave antes de cifrar y se la haga llegar al otro, por vía telemática, previo al envío del mensaje cifrado, es tarea ociosa, ya que los peligros de interceptación de la clave son similares a los existentes sobre la captura del mensaje.

Como se verá más adelante, en la tecnología de Seguridad en Redes, en la que de los *dominios de seguridad* pueden formar parte un elevadísimo número de interlocutores, uno de los problemas más necesarios de resolver es la generación y distribución de claves, por lo que un sistema del tipo de clave continua aleatoria es del todo

Mensaje en claro	estoesunmensajeindescifrable
Clave obtenida al azar	tbxkvzhsmqvjnfugdrysmajtcep
Criptograma (suma módulo 27)	yurzascgvyjcñoudthwrvugbuepu

Figura 3.1. Obtención del criptograma en un criptosistema de secreto perfecto.

impracticable. Dejémoslo, pues, como una curiosidad o como una alternativa para dominios restringidos y de limitadísimo acceso como puede ser el de los espías o el de los amantes.

DISTANCIA DE UNICIDAD

Es un concepto puramente teórico de difícil rentabilidad en la práctica. Se define como el número de caracteres cifrados necesarios para que la incertidumbre acerca de la clave, conocido el criptograma, sea cero o muy pequeña. Es decir, el número de caracteres cifrados que se necesitan para que el mensaje en claro sea perfectamente deducible.

Cuando el tamaño del mensaje es menor que la distancia de unicidad, existirán varias posibles soluciones que hagan corresponder claves y mensajes en claro. Cuanto mayor es la distancia de unicidad, tanto más robusto es el criptosistema con que ha sido cifrado el criptograma. Obviamente, cuanto mayor es el tamaño de un criptograma cifrado con una misma clave, tanto más se aproxima a la distancia de unicidad y más posibilidades tiene el criptoanalista de averiguar el mensaje en claro.

Cuando el sistema es de secreto perfecto, la distancia de unicidad es siempre mayor que la longitud del mensaje por muy largo que éste sea.

CONFUSIÓN Y DIFUSIÓN

Para los criptosistemas de **clave secreta** o simétricos, Shannon propuso dos técnicas tendentes a reducir la información subyacente en el mensaje cifrado a causa de las redundancias del lenguaje.

La *difusión* es una de ellas. Su objetivo es dispersar a lo largo del criptograma las propiedades estadísticas del mensaje en claro. Separando al máximo los caracteres que aparecen juntos en el mensaje, se dispersa la información relacionada con la aparición de digramas, trigramas y otras combinaciones de frecuente aparición. Ello se consigue mediante transposiciones y permutaciones de los bits o caracteres.

La otra técnica propuesta es la *confusión*, que consiste en establecer una dependencia funcional lo más compleja posible entre la clave y el mensaje cifrado. Para ello se usa la técnica de *sustitución* de un carácter del texto por otro u otros caracteres en el criptograma. Naturalmente, cuanto más elaborado sea este mecanismo de sustitución, tanto más eficaz será el efecto conseguido (el algoritmo de César es un ejemplo simplísimo de la técnica de sustitución).

3.2. TEORÍA DE NÚMEROS

Los *algoritmos de clave pública* se basan en problemas de difícil solución (véase apartado 3.3), de forma que tanto el emisor que cifra el mensaje como el receptor autorizado que lo descifra realizan operaciones matemáticas que, aun teniendo cierta complejidad, son asequibles (en tiempo, capacidad de proceso y requisitos de memoria) para los sistemas informáticos hoy día en uso. En cambio, un intruso que intente conocer el mensaje en claro a partir del criptograma se encontrará con que la operación matemática que para ello debería realizar es computacionalmente impracticable con los recursos de que dispone.

Habitualmente, en las operaciones aritméticas que pueden llevarse a cabo usando ordenadores, cuando se trata de números pequeños que necesitan para su exacta representación de un número elevado (o infinito) de decimales o cuando se trata de grandes números, se recurre a una representación aproximada que sirva para resolver con la fidelidad suficiente el problema planteado. Sin embargo, en el caso de la Criptografía los resultados han de ser exactos: no se trata de, dado un criptograma, recuperar el mensaje en claro con cierta aproximación, se trata de recuperarlo exactamente como ha sido concebido por su emisor.

UN CONJUNTO FINITO DE NÚMEROS CON LOS QUE PODER OPERAR

Ciertas operaciones, por ejemplo las de exponenciación, dan como resultados números muy grandes, que pueden crecer rápidamente y provocar desbordamientos en los elementos de cálculo del ordenador, máxime si se están barajando grandes números como es el caso de los criptosistemas de uso común. Por todo ello, se requiere moverse en un campo en el que se puedan realizar todas las operaciones aritméticas obteniendo resultados exactos (no valen las aproximaciones) sin desbordar la capacidad de representación de datos de los ordenadores. Este será el caso de la aritmética modular y, más generalmente, de los denominados *Campos de Galoise*.

Para poder llevar a cabo estas operaciones sería de interés poder operar sobre un conjunto finito de elementos con los que puedan realizarse las operaciones de suma y multiplicación, de tal modo que el resultado de cualquier operación fuese un elemento perteneciente a ese mismo conjunto finito de números. Para ello deberá estar definido el elemento neutro para cada una de las dos operaciones (el cero para la suma y el 1 para la multiplicación), cada elemento distinto de 0 ha de tener su correspondiente inverso respecto de la suma, y cada elemento distinto del 1 (excepto el 0) dispondrá de su correspondiente elemento inverso respecto de la multiplicación. Además, tanto la suma como la multiplicación han de cumplir las leyes conmutativa y asociativa y el producto ha de tener la propiedad distributiva respecto de la suma. La estructura algebraica de un conjunto de tales características sería la de *cuerpo conmutativo* o *campo*.

Las condiciones someramente expuestas en los párrafos anteriores distan mucho de ser triviales y fáciles de satisfacer. Por mor de la exactitud en su representación, no es posible el uso de números reales, y el conjunto $\mathbb{Z}$ de los números enteros no cumple las condiciones expuestas en el párrafo anterior en cuanto a la existencia de inverso respecto a la multiplicación. Veremos a continuación cómo con la aritmética modular, bajo determinadas condiciones, sí es posible encontrar conjuntos que cumplan con todas esas propiedades.

Aritmética modular

Dados a , r , k y n enteros $\neq 0$,

$$\text{Si} \quad a = k \cdot n + r \text{ (donde } r < n) \quad (5)$$

se dice que r es el resto de a módulo n o

$$r = a \bmod n$$

Dados a, b y $n \neq 0$ y enteros, si $a \bmod n = r$ y $b \bmod n = r$, se dice que a es congruente con b módulo n y se representa como:

$$\begin{array}{ll} & a \equiv_n b \\ \text{o también como} & a \equiv b \bmod n, \\ \text{o simplemente} & a = b \bmod n \end{array}$$

Se verifica que:

$$a - b = k \cdot n \text{ (para algún entero } k) \quad (6)$$

y se dice que a es un resto o residuo de b o que b es un resto o residuo de a .

Por ejemplo, puede afirmarse que 32 es congruente con -4 módulo 6, porque:

$$32 - (-4) = 36 = k \cdot 6 \text{ (donde } k = 6),$$

$$\text{o bien, según (5):} \quad 32 = 5 \cdot 6 + 2 \text{ y } -4 = 1 \cdot 6 + 2$$

De igual forma, 9 es congruente con 4 módulo 5 y 28 es congruente con 12 módulo 4.

Se llama *clase de equivalencia* definida por un número $a \bmod n$, y se representa por $[a]$, al conjunto de todos los números enteros que son congruentes con a módulo n . El conjunto (finito) de clases de equivalencia se denota por $\mathbb{Z}_n$.

Es fácil de demostrar que la relación de congruencia cumple con las propiedades *reflexiva, simétrica y transitiva*.

Asimismo, se pueden definir las operaciones suma y producto, y se demuestra que se cumplen las propiedades:

Asociativa respecto a la suma y respecto al producto:

$$\begin{aligned} a + (b + c) \bmod n &= (a + b) + c \bmod n \\ a \cdot (b \cdot c) \bmod n &= (a \cdot b) \cdot c \bmod n \end{aligned}$$

Conmutativa respecto a la suma y respecto al producto:

$$\begin{aligned} a + b \bmod n &= b + a \bmod n \\ a \cdot b \bmod n &= b \cdot a \bmod n \end{aligned}$$

Distributiva del producto respecto de la suma

$$a \cdot (b + c) \bmod n = (a \cdot b + a \cdot c) \bmod n$$

Es también inmediato comprobar la existencia de **elemento neutro** para la suma (el cero) y para la multiplicación (el 1).

Dado un número a , existe un único **elemento inverso** respecto a la suma ($-a$) tal que la suma de ambos sea igual al elemento neutro, el cero. No obstante, para el caso de la multiplicación, la existencia de un inverso único sólo se da en determinadas circunstancias. Más adelante veremos cuáles son.

PRINCIPIO DE LA ARITMÉTICA MODULAR

La relación de congruencia módulo n divide el conjunto infinito de números enteros en n clases de equivalencia, cada una de las cuales está «representada» por uno de los restos. La reducción módulo n supone un homomorfismo del anillo de los enteros al anillo de los enteros módulo n , constituido por un número finito de elementos.

Se define como **conjunto completo de restos** (mod n) y se denota como CCR al conjunto de enteros $\{r_1, r_2, \dots, r_1, \dots, r_n\}$ tales que se cumple que para todo $a \in \mathbb{Z}$ existe un elemento y sólo uno, r_i , tal que $a = r_i \pmod{n}$, donde $(1 \leq i \leq n)$. En otras palabras, cualquier entero estará «representado» por uno y sólo uno de los restos del conjunto. El conjunto $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$ es el **conjunto básico de restos** módulo n .

Se puede demostrar que dados dos enteros a y b y una determinada operación aritmética **op** (definida en base a las de suma, resta y multiplicación), se obtiene el mismo resultado realizando primero la operación completa y luego reduciendo módulo n que reduciendo los resultados intermedios módulo n y operando después [Rami98]. Esto es:

$$(a \text{ op } b) \pmod{n} = ((a \pmod{n}) \text{ op } (b \pmod{n})) \pmod{n} \quad (7)$$

En la Figura 3.2 se representa, de forma gráfica, esta propiedad. Como el resultado es el mismo, por lo general, el camino más sencillo será realizar primero la reducción de los operandos y proceder después a realizar la operación.

Aplicando este principio, se tendría:

$$(a + b) \pmod{n} = ((a \pmod{n}) + (b \pmod{n})) \pmod{n}$$

$$(a - b) \pmod{n} = ((a \pmod{n}) - (b \pmod{n})) \pmod{n}$$

$$(a \cdot b) \pmod{n} = ((a \pmod{n}) \cdot (b \pmod{n})) \pmod{n}$$

$$(a \cdot (b + c)) \pmod{n} = (((a \cdot b) \pmod{n}) + ((a \cdot c) \pmod{n})) \pmod{n}$$

En la Figura 3.3 se representan sobre una espiral los primeros números enteros de forma que puede observarse cómo todos los números que se representan sobre la misma línea vertical imaginaria pertenecen a una misma clase de equivalencia de las cinco en que se ha dividido el conjunto infinito de los enteros. Cuando, como resultado

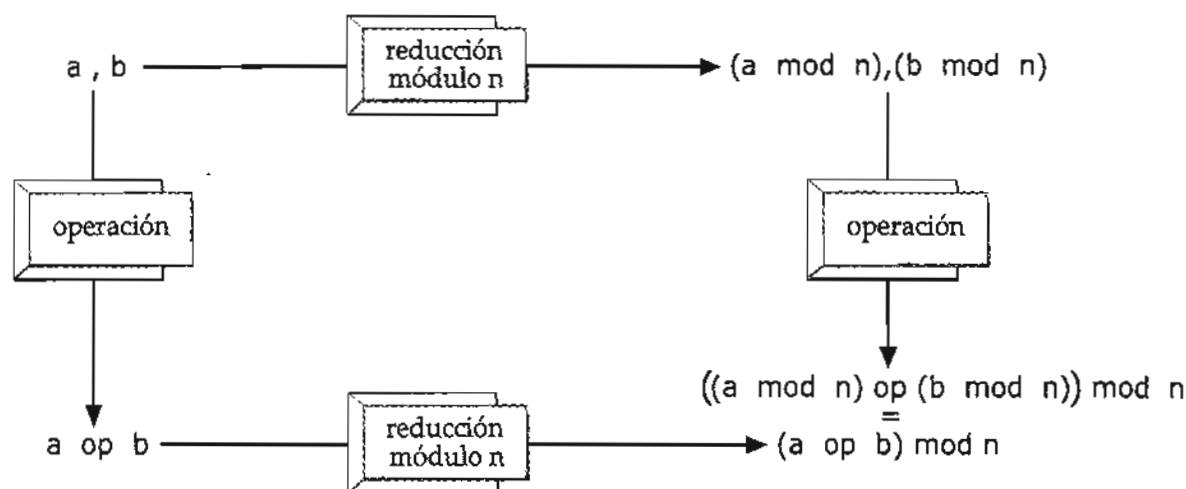


Figura 3.2. Principio de la aritmética modular.

de una operación, se genere un número relativamente elevado, por ejemplo 14, puede reducirse módulo 5, obtener el 4 y continuar operando.

En las **operaciones de exponenciación** pueden realizarse estas reducciones intermedias para simplificar los cálculos. Por ejemplo, para calcular $4^7 \bmod 5$, puede operarse:

a) Sin reducción:

$$\begin{aligned} 4^2 &= 4 \cdot 4 = 16 \\ 4^3 &= 16 \cdot 4 = 64 \\ 4^6 &= 64 \cdot 64 = 4.096 \\ 4^7 &= 4.096 \cdot 4 = 16.384 \end{aligned}$$

y, posteriormente:

$$4^7 \bmod 5 = 16.384 \bmod 5 = 4$$

b) Con reducción en los pasos intermedios:

$$\begin{aligned} 4^2 \bmod 5 &= 16 \bmod 5 = 1 \\ 4^3 \bmod 5 &= (4^2 \bmod 5 \cdot 4 \bmod 5) \bmod 5 = (1 \cdot 4) \bmod 5 = 4 \\ 4^6 \bmod 5 &= (4^3 \bmod 5)^2 \bmod 5 = 4^2 \bmod 5 = 1 \\ 4^7 \bmod 5 &= (4^6 \bmod 5 \cdot 4 \bmod 5) \bmod 5 = (1 \cdot 4) \bmod 5 = 4 \end{aligned}$$

En este elemental ejercicio se observa que el método de reducción de los resultados intermedios simplifica notablemente el cálculo. En los sistemas criptográficos será necesario realizar operaciones de exponenciación donde intervienen grandes números. Como más adelante se comentará, se manejan números de mil y dos mil dígitos binarios, que, caso de no poderse realizar reducciones en los pasos intermedios, arrojarían valores difícilmente manejables en un ordenador. Como ya se indicó, los resultados de las operaciones han de ser exactos (no valen aproximaciones), por lo que al

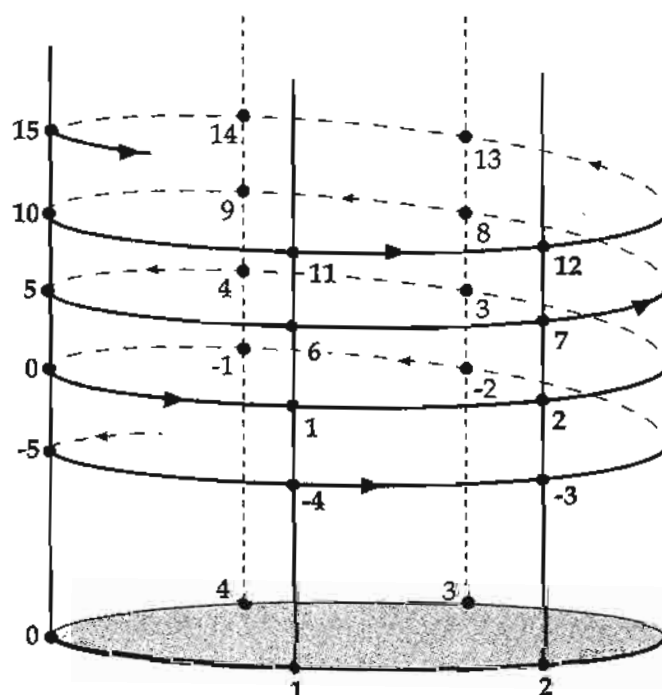


Figura 3.3. Representación gráfica de números enteros.

operar en aritmética modular, no sólo se simplifican los cálculos sino que se restringe el rango necesario para representar los resultados intermedios, facilitando así la implementación de los algoritmos.

La operación $a^x \bmod n$ puede llevarse a cabo mediante multiplicaciones reiteradas sin más que buscar la suma de potencias de 2 en que puede descomponerse x , lo cual no representa un problema complejo para ser abordado mediante un programa de ordenador. Sea, por ejemplo, $15^{23} \bmod 55$:

Expresado en binario, $23_{10} = 10.111_2 = 2^4 + 0 + 2^2 + 2^1 + 2^0 = 16 + 4 + 2 + 1$

Por tanto:

$$15^{23} \bmod 55 = 15 \cdot 15^2 \cdot 15^4 \cdot 15^{16} \bmod 55 = 15 \cdot 15^2 \cdot (15^2)^2 \cdot ((15^2)^2)^2 = 20$$

Hay algoritmos concretos, como el algoritmo de exponenciación rápida de Knuth [Schn96], que sistematizan el procedimiento para resolver este problema.

Logaritmos discretos

Como acabamos de ver, conocidos x y n , resulta asequible calcular

$$y = a^x \bmod n$$

Por contra, el problema inverso, esto es, conocidos n e y , calcular x en aritmética modular es inabordable computacionalmente y pertenece a los denominados *problemas intratables*. Si se pudiese plantear y resolver se expresaría como:

$$x = \log_a y \bmod n$$

Este tipo de problemas resulta de interés a la hora de diseñar algoritmos criptográficos asimétricos.

ELEMENTOS INVERSOS RESPECTO A LA MULTIPLICACIÓN

Para poder realizar divisiones exactas en aritmética modular (operaciones necesarias en los algoritmos criptográficos) es necesario el cálculo del inverso de un número respecto de la multiplicación. Dados a y n , se dice que x es inverso de a módulo n , y se representa por a^{-1} , si se cumple que

$$x \cdot a \bmod n = 1$$

$$\text{o bien} \quad a^{-1} \cdot a \bmod n = 1 \quad (8)$$

Como puede comprobarse en el desarrollo del ejemplo sobre el cálculo de $4^7 \bmod 5$, antes expuesto, 4 es inverso de 4 módulo 5.

No en todos los casos se da la circunstancia de que exista un elemento inverso y que éste sea único. Se puede demostrar el siguiente teorema:

Dado $a \in \mathbb{Z}$, $n \in \mathbb{Z}$ tal que $a \in \{0, \dots, n-1\}$ y $\text{mcd}(a, n) = 1$, entonces

$$\exists \text{ un único } x \in \{0, \dots, n-1\} \mid$$

$$a \cdot x \bmod n = 1 \quad \text{o} \quad x = a^{-1} \bmod n$$

Es decir, existe un único número en el conjunto $\{0, \dots, n-1\}$ que sea inverso de a . En caso de que a y n no sean primos entre sí, no existirá un único elemento inverso, o sea:

Existe elemento inverso de $a \bmod n$ si y sólo si $\text{mcd}(a, n) = 1$.

De otra forma: cada $(a \cdot i) \bmod n$, donde $0 \leq i \leq n-1$, es un resto mod n distinto. Es decir:

$$\{a \cdot i \bmod n\}_{i=0}^{i=n-1}$$

es una permutación completa del conjunto completo de restos $\mathbb{Z}_n\{0, 1, \dots, n-1\}$.

Decir que $\text{mcd}(a, n) = 1$ es decir que son primos relativos o que son primos entre sí. El algoritmo de Euclides permite encontrar, mediante un proceso iterativo, el mcd de dos números basándose en que si $a > b$, tal que $a = k \cdot b + r$, donde $0 < r < b$, se cumple que:

$$\text{mcd}(a, b) = \text{mcd}(b, r)$$

Mediante el algoritmo de Euclides es posible determinar si dos números son primos entre sí.

Ejemplo 1: Permutaciones del conjunto básico de restos para $a = 5$ y $n = 7$. Consideremos el conjunto completo de restos básico mod n : $\{0, 1, 2, 3, 4, 5, 6\}$. Se cumple que $\text{mcd}(5, 7) = 1$, por tanto se trata de números primos entre sí.

Se puede comprobar que si multiplicamos 5 por cada elemento del conjunto anterior obtendremos una permutación de él mismo (es decir, la *tabla de multiplicar* del 5).

$$\begin{array}{lll} 5 \cdot 0 = 0 \bmod 7 & 5 \cdot 1 = 5 \bmod 7 & 5 \cdot 2 = 3 \bmod 7 \\ 5 \cdot 3 = 1 \bmod 7 & 5 \cdot 4 = 6 \bmod 7 & 5 \cdot 5 = 4 \bmod 7 \\ 5 \cdot 6 = 2 \bmod 7 & & \end{array}$$

No sólo el 5, sino todos los restantes elementos del conjunto básico de restos tendrían, por ser 7 un número primo, un único elemento inverso: se dice que todos los elementos son *invertibles*.

Ejemplo 2: Sin embargo, si a y n no son primos entre sí, no se obtendrá una permutación del conjunto completo de restos básico. Así, en el caso de $a = 4$ y $n = 6$, se obtiene:

$$\begin{array}{lll} 4 \cdot 0 = 0 \bmod 6 & 4 \cdot 1 = 4 \bmod 6 & 4 \cdot 2 = 2 \bmod 6 \\ 4 \cdot 3 = 0 \bmod 6 & 4 \cdot 4 = 4 \bmod 6 & 4 \cdot 5 = 2 \bmod 6 \end{array}$$

Como se observa, no hay, en este caso, ningún elemento inverso del 4, y se dice que 4 no es *invertible*. En cambio, el 5 (primo de 6) sí es *invertible*: produciría una permutación completa y tendría un inverso único, el 5.

CÁLCULO DE INVERSOS. FUNCIÓN INDICADORA DE EULER

Se define como *conjunto reducido de restos* módulo n , y se denota por $\mathbb{Z}_n^*$, al subconjunto formado por los restos $\{1, \dots, n-1\}$ cuyos elementos son *invertibles*, es decir, son primos relativos con n . Si n es primo, el conjunto reducido de restos es el conjunto completo de restos, excepto el 0. Por ejemplo:

Conjunto reducido de restos mod 6: {1,5}

Conjunto reducido de restos mod 7: {1,2,...,6}

La Función indicadora de Euler es igual al cardinal del conjunto reducido de restos y se denota como $\Phi(n)$.

Si p es primo, como ha podido comprobarse en párrafos anteriores, se cumple que:

$$\Phi(p) = p-1 \quad (9)$$

Si p y q son primos y $n = p \cdot q$ es fácil de demostrar que:

$$\Phi(n) = \Phi(p \cdot q) = (p-1) \cdot (q-1) = \Phi(p) \cdot \Phi(q) \quad (10)$$

En general:

$$\text{Si } n = \prod_{i=1}^k p_i^{e_i} \text{ donde:} \quad (11)$$

p_i representa a los distintos factores primos,

k = número de factores primos,

e_i = exponente con que aparece cada factor,

se puede demostrar (aunque no de forma tan sencilla) que la función de Euler es:

$$\Phi(n) = \prod_{i=1}^k p_i^{e_i-1} \cdot (p_i - 1) \quad (12)$$

Ejemplo 1: Sea:

Como $21 = 3 \cdot 7$, ambos primos, se tiene:

$$\Phi(21) = (3 - 1) \cdot (7 - 1) = 12$$

Ejemplo 2: Calcular $\Phi(72)$:

Descomponiendo 72 en sus factores primos quedaría:

$$72 = 2^3 \cdot 3^2$$

Por lo que:

$$\Phi(72) = 2^2 (2 - 1) \cdot 3^1 (3 - 1) = 4 \cdot 6 = 24$$

El problema de la factorización es computacionalmente complejo debido a la ausencia de algoritmos eficaces, ya que el método del tanteo (que ya usábamos con éxito en la enseñanza primaria para obtener estos factores) se convierte en algo impracticable cuando se trata de grandes números. Es relativamente sencillo elegir varios números primos grandes, elevarlos a una determinada potencia y multiplicarlos entre sí. Pero el problema inverso es sumamente difícil de resolver. Por eso se usa en determinados algoritmos criptográficos.

Teorema de Fermat

Si p es primo, para todo a perteneciente al conjunto reducido de restos de p , se cumple que $\text{mcd}(a, p) = 1$. En este caso, se demuestra que:

$$a^{p-1} \bmod p = 1 \quad (13)$$

Ejemplo: Comprobamos que $4^{5-1} \bmod 5 = 1$

En la tabla de potencias de 4 módulo 5 vistas en un ejemplo anterior, se observa que:

$$4^4 \bmod 5 = 1$$

Teorema de Euler

El teorema de Fermat es un caso particular del teorema de Euler que se enuncia así: Para todo par de enteros a, n tal que $\text{mcd}(a, n) = 1$, se cumple:

$$a^{\Phi(n)} \bmod n = 1 \quad (14)$$

Se pueden **calcular inversos** usando este teorema. Por definición:

$$a^{-1} \cdot a \bmod n = 1 \quad (15)$$

Por el teorema de Euler:

$$a \cdot a^{-1} \bmod n = 1 = a^{\Phi(n)} \bmod n$$

De donde se deduce que el inverso de a , a^{-1} , es:

$$a^{-1} = a^{\Phi(n)-1} \bmod n \quad (16)$$

Si n es un número primo, se tiene que $\Phi(n) = n - 1$, por lo que:

$$a^{-1} = a^{n-2} \bmod n \quad (17)$$

Ejemplo 1: Dados $a = 4$ y $n = 7$, primos entre sí, calcúlese el inverso de 4 módulo 7. Por ser 7 un número primo:

$$\Phi(n) = 7 - 1 = 6$$

Para hallar $x = 4^{-1} \bmod 7$, aplicando la fórmula deducida del teorema de Euler:

$$x = 4^{6-1} \bmod 7 = 4^5 \bmod 7 = 2 = x$$

Ejemplo 2: Calcúlese el inverso de 4 mod 15 ($a = 4$, $n = 15$).

Aunque n no es primo, es primo relativo con 4, por lo que existirá un único inverso, x . Descomponiendo en factores primos, se tiene:

$$\begin{aligned} n &= 15 = 3 \cdot 5 \\ \Phi(n) &= (3-1) \cdot (5-1) = 8 \end{aligned}$$

Por lo que:

$$x = 4^{8-1} \bmod 15 = 4^7 \bmod 15 = 4 = x$$

Se pueden calcular también los inversos mediante una variante del algoritmo de Euclides para el cálculo del máximo común divisor. Este procedimiento, debido a

Knuth es conocido como *Algoritmo Extendido de Euclides*; en [Schn96] puede encontrarse su descripción y un programa en C++ que lo implementa.

TEOREMA CHINO DE LOS RESTOS

Llamado así por haber sido demostrado por un matemático chino del siglo I de nuestra era, sirve para resolver ecuaciones del tipo:

$$a \cdot x \pmod{n} = b \quad (18)$$

Donde a , b y n son conocidos y tal que a y n son primos entre sí. Ello equivale a resolver un sistema de ecuaciones, ya que el enunciado del teorema, de fácil demostración, dice que siendo $n = d_1 \cdot d_2 \dots d_k$ tal que $d_1 \dots d_k$ sean primos entre sí dos a dos, el sistema

$$\begin{array}{l} x \pmod{d_1} = x_1 \\ x \pmod{d_2} = x_2 \\ \dots \dots \dots \\ x \pmod{d_k} = x_k \end{array}$$

tiene solución común en el rango $\{0, n-1\}$ y ésta es:

$$x = \sum_{i=1}^k \frac{n}{d_i} \cdot y_i \cdot x_i \pmod{n} \quad (19)$$

Donde y_i es el inverso de n/d_i módulo d_i , y x_i se obtiene de resolver:

$$\begin{array}{l} a \cdot x_1 \pmod{d_1} = b \pmod{d_1} \\ a \cdot x_2 \pmod{d_2} = b \pmod{d_2} \\ \dots \dots \dots \\ a \cdot x_k \pmod{d_k} = b \pmod{d_k} \end{array}$$

Ejemplo: Calcúlese el valor de x que cumpla con

$$7 \cdot x \pmod{30} = 17$$

Como $30 = 2 \cdot 3 \cdot 5$, resultarán las ecuaciones:

$$\begin{array}{ll} 7 \cdot x_1 \pmod{2} = 17 \pmod{2} = 1, & \text{cuya solución es: } x_1 = 7^{\Phi(2)-1} \pmod{2} = 1 \\ 7 \cdot x_2 \pmod{3} = 17 \pmod{3} = 2, & \text{cuya solución es: } x_2 = 2 (7^{\Phi(3)-1} \pmod{3}) = 2 \\ 7 \cdot x_3 \pmod{5} = 17 \pmod{5} = 2, & \text{cuya solución es: } x_3 = 2 (7^{\Phi(5)-1} \pmod{5}) = 1 \end{array}$$

Para los valores y_i (inversos de $n/d_i \pmod{d_i}$) se tiene:

$$\frac{30}{2} y_1 \pmod{2} = 1 \quad \Rightarrow \quad y_1 = 15^{\Phi(2)-1} \pmod{2} = 1$$

$$\frac{30}{3} y_2 \pmod{3} = 1 \quad \Rightarrow \quad y_2 = 10^{\Phi(3)-1} \pmod{3} = 1$$

$$\frac{30}{5} y_3 \pmod{5} = 1 \quad \Rightarrow \quad y_3 = 6^{\Phi(5)-1} \pmod{5} = 1$$

Aplicando la ecuación (19), el valor de x resulta ser:

$$x = \left(\frac{30}{2} + \frac{30}{3} \cdot 2 + \frac{30}{5} \right) \bmod 30 = 11$$

En efecto, puede comprobarse que sustituyendo este valor de x en la ecuación inicial que planteaba el problema, se obtiene:

$$7 \cdot 11 \bmod 30 = 17$$

CAMPOS DE GALOIS DEL TIPO $\text{GF}(q^n)$

Como hemos venido exponiendo en los apartados precedentes, cuando p es un número primo, el conjunto de números enteros comprendidos entre 1 y $p-1$, resultante de reducir módulo p todos los elementos del conjunto $\mathbb{Z}$, presenta la peculiaridad de que, definidas en él las operaciones de suma y multiplicación, los elementos resultantes de realizar dichas operaciones pertenecen también al mismo conjunto $[0, p-1]$. Además, siendo 0 y 1 los elementos neutros para, respectivamente, la suma y la multiplicación, cada elemento tiene su inverso respecto a la suma dentro del conjunto y, también, cada elemento distinto de cero tiene su inverso respecto a la multiplicación. Asimismo, se cumplen las leyes conmutativa, asociativa y distributiva, por lo que estos conjuntos tienen, como ya se ha dicho, la estructura algebraica de un *cuerpo conmutativo* o *campo*. Estos campos definidos módulo un número primo p reciben el nombre de *Campos de Galois* en honor de un matemático francés que vivió a principios del siglo XIX y que antes de morir violentamente a la edad de veinte años tuvo tiempo de realizar múltiples trabajos sobre este tipo de conjuntos. Abreviadamente, estos conjuntos se denominan $\text{CG}(p)$ o más frecuentemente, derivado de su nombre en inglés, $\text{GF}(p)$.

En algunos algoritmos criptográficos (más concretamente en el criptosistema Rijndael [DaRi99] que estudiaremos en el próximo capítulo) se hace uso de otro tipo de conjuntos que también tienen un número finito de elementos y estructura de campo. Se trata de los denominados $\text{GF}(q^n)$ cuyas características son:

- q es un número primo y n denota el grado de los polinomios que constituyen el conjunto.
- Los elementos $a(x)$ del conjunto son polinomios de grado $n-1$ o menor, del tipo:

$$a(x) = a_{n-1} \cdot x^{n-1} + a_{n-2} \cdot x^{n-2} + \dots + a_1 \cdot x + a_0$$

- Los coeficientes a_i son enteros q módulo q , es decir: $a_i \in [0, q-1]$.
- El número de elementos del conjunto es finito (lo que posibilita la obtención de valores exactos en las operaciones realizadas en un computador).
- Cada elemento $a(x)$ es un *resto* módulo $p(x)$, donde $p(x)$ es un polinomio *irreducible* de grado n .
- Decir que $p(x)$ es un polinomio irreducible significa que no puede descomponerse en el producto de otros polinomios de grado inferior. Es decir, no tiene otros divisores más que él mismo y la unidad.

Esto es, el mismo papel que juega un número primo p (que no puede expresarse como el producto de otros números enteros menores que él) en la determinación de un $\text{GF}(p)$ lo juega en los $\text{GF}(q^n)$ el polinomio irreducible $p(x)$.

Para los fines aquí perseguidos, dentro de estos campos $GF(q^n)$, los que tienen interés son los campos del tipo $GF(2^n)$, es decir, aquellos en los que los coeficientes de los términos de los polinomios se hallan comprendidos dentro del conjunto $\{0,1\}$. (a veces también se denotan como $GF(2^n)$ para abreviar su representación).

En estos casos, en las operaciones de **suma** entre elementos del conjunto, el coeficiente de cada uno de los términos del polinomio resultante se obtendrá aplicando la suma módulo 2 de los correspondientes coeficientes de los términos de los sumandos con igual exponente. Esto es, si se trata de sumar los elementos:

$$\begin{aligned} a &= a_{n-1} \cdot x^{n-1} + a_{n-2} \cdot x^{n-2} + \dots + a_1 \cdot x + a_0 \\ b &= b_{n-1} \cdot x^{n-1} + b_{n-2} \cdot x^{n-2} + \dots + b_1 \cdot x + b_0 \end{aligned}$$

donde todos los coeficientes pertenecen al conjunto $\{0,1\}$, la suma $(a_i + b_i \bmod 2)$ de los coeficientes de los términos de exponente i , será 0 o 1, conforme a las reglas del OR exclusivo. El elemento neutro respecto a la suma es el cero (un polinomio en el que todos sus coeficientes valen cero), lo que conlleva que todo elemento es inverso de sí mismo respecto a la suma (el OR exclusivo de un número binario consigo mismo es cero). Es decir, en este conjunto la suma y la sustracción son la misma operación.

En cuanto a la **multiplicación**, la reducción módulo el polinomio irreducible $p(x)$ se realizará, obviamente, multiplicando de forma convencional los polinomios y obteniendo el resto de la operación de dividir por $p(x)$ el polinomio resultante.

El caso más sencillo que se presenta es el de $GF(2^2)$, en el que los cuatro polinomios elementos del conjunto son los representados por los vectores 11, 10, 01 y 00, es decir: $x + 1$, x , 1 y 0.

La suma de dos de estos elementos, por ejemplo $a = x + 1$ y $b = x$, será:

$$a \oplus b = (x + 1) + x = 2x + 1 = 1$$

Que se podría haber obtenido directamente realizando la operación OR exclusivo entre los vectores 11 y 01.

En cuanto al producto, si $p(x) = x^2 + x + 1$, la multiplicación de esos dos mismos elementos a y b será:

$$a \cdot b = (x + 1) \cdot x = x^2 + x$$

Para reducir este polinomio módulo $p(x)$ puede sustituirse x^2 por su residuo respecto a $p(x)$, es decir: $x^2 = x + 1$. Aplicando esta reducción por exponente, resulta:

$$a \cdot b = x^2 + x = x + 1 + x = 1$$

En estas estructuras algebraicas también está garantizada la existencia de **elemento inverso** respecto a la multiplicación para todos los elementos excepto para el cero. En el ejemplo anterior, a y b son inversos multiplicativos. Se puede demostrar que los teoremas antes expuestos para el cálculo de elementos inversos en aritmética modular son también válidos en los campos $GF(2^n)$, por lo que el cálculo de inversos puede llevarse a cabo aplicando el algoritmo extendido de Euclides o el teorema de Euler. Así, la expresión (16) adopta aquí la forma:

$$a^{-1} = a^{\Phi(p(x)) - 1} \bmod p(x) \quad (20)$$

El valor de $\Phi(p(x))$ es el cardinal del subconjunto formado por todos los polinomios de $GF(2^n)$ que sean «primos» con $p(x)$, es decir, que no se puedan descomponer

en polinomios comunes con aquellos en que se pueda descomponer $p(x)$. Como quiera que $p(x)$ es irreducible, ese subconjunto está formado por todos los polinomios excepto el cero, por lo que:

$$\Phi(p(x)) = 2^n - 1 \quad (21)$$

Ejemplo: En el conjunto $GF(2^3)$, donde $p(x) = x^3 + x + 1$, calcúlese el inverso del polinomio representado por 100_2 , es decir, $a(x) = x^2$.

Aplicando (21) resulta:

$$\Phi(p(x)) = 2^3 - 1 = 7$$

Aplicando (20) resulta:

$$a^{-1} = a^6 \bmod p(x) = a^4 \cdot a^2 \bmod p(x)$$

Elevando al cuadrado $a(x)$ y aplicando reducción por exponente (sustituyendo x^3 por $x + 1$) resulta el polinomio:

$$a^2 = x^4 = x \cdot (x^3) = x \cdot (x + 1) = x^2 + x$$

Operando de forma similar (y aplicando también reducción de coeficientes) resulta:

$$a^4 = (x^2 + x)^2 = x^4 + 2x^3 + x^2 = (x^2 + x) + x^2 = x$$

Por tanto, el elemento inverso de $a(x)$ que se proponía calcular será:

$$a^{-1} = a^6 = a^4 \cdot a^2 = x \cdot (x^2 + x) = x^3 + x^2 = x^2 + x + 1 = a^{-1}$$

Estas mismas operaciones pueden llevarse a cabo utilizando los vectores o números binarios que representan a los distintos polinomios del conjunto finito $GF(2^3)$. Así, la última operación que aparece en el ejemplo anterior, el producto de $a^4 = x$ por $a^2 = x^2 + x$, se puede llevar a cabo multiplicando sus representantes binarios 010_2 y 110_2 , lo que consiste en sumar módulo 2 el multiplicando consigo mismo desplazado tantos lugares a la izquierda como lugar ocupe el correspondiente «1» del multiplicador. Es decir:

Multiplicando:	010
Multiplicador:	• 110
	$\begin{array}{r} 010 \\ 010 \\ 010 \\ \hline 01100 \end{array}$
Producto:	01100

Para reducir el polinomio resultante módulo $p(x)$, de forma que no tenga ningún término con potencia de x superior a 2, se realiza un OR exclusivo con 1011 , correspondiente al polinomio $p(x)$ (debido a que la operación de resta es la misma que de la suma). Es decir:

Resultado parcial:	1100
$p(x)$:	$\oplus 1011$
Resultado:	0111

que, obviamente, se corresponde con el resultado antes obtenido ($a^{-1} = x^2 + x + 1$).

En el algoritmo **Rijndael** (que estudiaremos en el siguiente capítulo) se realizan operaciones con octetos donde los elementos pertenecen al campo $GF(2^8)$ [DaRi99]. Las operaciones de suma y multiplicación de esos elementos módulo $p(x) = x^8 + x^4 + x^3 + x + 1$ se llevan a cabo de forma equivalente a como hemos visto en los ejemplos (más sencillos) precedentes.

En las **operaciones de suma** se aplica la reducción de coeficientes módulo 2.

Ejemplo: La suma de los elementos $a = 57_{16} = 0101\ 0111_2$ y $b = 83_{16} = 1000\ 0011_2$ puede realizarse, en notación binaria, mediante una simple operación OR exclusivo, dando como resultado $1101\ 0100_2 = D4_{16}$. En notación polinomial sería:

$$a \oplus b = (x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = (x^7 + x^6 + x^4 + x^2 + 2x + 2) = x^7 + x^6 + x^4 + x^2$$

Las **operaciones de multiplicación** se realizan mediante la reducción de exponentes módulo un polinomio irreducible de grado 8 cuyo valor es $p(x) = x^8 + x^4 + x^3 + x + 1 = 1\ 0001\ 1011_2 = 11B_{16}$. A diferencia de lo que ocurre con la suma, la multiplicación no puede resolverse mediante una simple operación lógica a nivel de octetos.

La **multiplicación por x** es una operación relativamente sencilla. De forma genérica, sea $a(x)$ el polinomio que se desea multiplicar por x :

$$a(x) = a_7 x^7 + a_6 x^6 + a_5 x^5 + a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0$$

El polinomio $b(x)$ resultante de esa multiplicación será:

$$b(x) = a_7 x^8 + a_6 x^7 + a_5 x^6 + a_4 x^5 + a_3 x^4 + a_2 x^3 + a_1 x^2 + a_0 x$$

Para reducir módulo $p(x)$ el anterior polinomio sólo pueden darse dos situaciones:

1. $a_7 = 0$, en cuyo caso $b(x)$ no rebasa el grado 7 y la reducción no es necesaria, quedando el polinomio resultante tal y como aparece en la expresión anterior, que en notación binaria consiste en un desplazamiento a la izquierda del representante binario de $a(x)$.
2. $a_7 = 1$, en cuyo caso debe restarse $p(x)$, es decir, realizar una operación OR exclusivo con $11B_{16}$. En este caso el bit correspondiente al coeficiente de x^8 siempre resultará cero, por lo que la operación de multiplicación por x consistirá en un desplazamiento a la izquierda del representante binario de $a(x)$ y un XOR con $1B_{16}$.

Esta operación de multiplicar por x (02_{16} en hexadecimal) se denota como $b = \text{xtime}(a)$.

Ejemplo: La multiplicación por x de $a = 57_{16}$ es: $b = \text{xtime}(57) = AE_{16}$.

La **multiplicación entre dos polinomios** puede abordarse multiplicándolos directamente término a término y realizando luego la reducción por exponente, según hemos visto en ejemplos anteriores sobre $GF(2^2)$ y $GF(2^3)$.

Ejemplo: Calcúlese el producto de $a = 57_{16} = 0101\ 0111_2$ y $b = 83_{16} = 1000\ 0011_2$. En notación polinomial sería:

$$\begin{aligned} a &= 0101\ 0111_2 = x^6 + x^4 + x^2 + x + 1 \\ b &= 1000\ 0011_2 = x^7 + x + 1 \end{aligned}$$

El resultado que se obtiene de multiplicar término a término ambos polinomios y realizar la posterior reducción de coeficientes es:

$$a \cdot b = x^{13} + x^{11} + x^9 + x^8 + 2x^7 + x^6 + x^5 + x^4 + x^3 + 2x^2 + 2x + 1 \quad (22)$$

Los términos con exponente igual o superior a 8 deben ser sustituidos por su correspondiente expresión reducida módulo $p(x)$. Así, tras operaciones algo engorrosas, resulta:

$$x^8 = x^4 + x^3 + x + 1$$

$$x^9 = x \cdot x^8 = x \cdot (x^4 + x^3 + x + 1) = x^5 + x^4 + x^2 + x$$

$$x^{11} = x^3 \cdot x^8 = x^3 \cdot (x^4 + x^3 + x + 1) = x^7 + x^6 + x^4 + x^3$$

$$x^{13} = x^5 \cdot x^8 = x^5 \cdot (x^4 + x^3 + x + 1) = x^9 + x^8 + x^6 + x^5 = x^6 + x^3 + x^2 + 1$$

Que sustituidos en la expresión (22) dan como resultado:

$$a \cdot b = x^7 + x^6 + 1 = 1100\ 0001_2 = C1_{16}$$

Bastante más sencillo resulta realizar la operación de multiplicación utilizando la notación binaria de forma similar a como hemos visto en ejemplos anteriores (Figura 3.4). La reducción de términos con exponente igual o superior a 8 se realiza mediante la correspondiente operación OR exclusivo con 100011011_2 (representante del polinomio irreducible) desplazado a la izquierda los lugares necesarios para hacer coincidir el bit de mayor peso con el «1» correspondiente que representa un exponente igual o superior a 8.

Puede observarse cómo el producto $a \cdot b$ que aparece en la sexta línea de la Figura 3.4 se corresponde con la expresión (22), aunque con este procedimiento se ha obtenido de forma mucho más sencilla. Obviamente, el resultado final coincide con el valor $1100\ 0001_2 = C1_{16}$ obtenido mediante el otro procedimiento.

Para su plasmación en un algoritmo, otra forma de abordar la multiplicación es mediante la aplicación reiterada de la operación *xtime*. En efecto, dado el multiplicando $a = 57_{16}$, se pueden obtener los resultados de multiplicarlo por las distintas potencias de x mediante la aplicación reiterada de la operación *xtime*. Es decir:

$$57_{16} \cdot 01_{16} = 57_{16}$$

$$57_{16} \cdot 02_{16} = \text{xtime}(57) = AE_{16}$$

a:	01010111
b:	• 10000011
	01010111
	01010111
	01010111
	01010111
Producto a • b :	10101101111001
P(x) desplazado :	⊕ 100011011
Resultado parcial :	100000011001
P(x) desplazado :	⊕ 100011011
Resultado final :	11000001

Figura 3.4. Multiplicación de dos polinomios.

$$\begin{aligned}
57_{16} \cdot 04_{16} &= \text{xtime}(AE) = 47_{16} \\
57_{16} \cdot 08_{16} &= \text{xtime}(47) = 8E_{16} \\
57_{16} \cdot 10_{16} &= \text{xtime}(8E) = 07_{16} \\
57_{16} \cdot 20_{16} &= \text{xtime}(07) = 0E_{16} \\
57_{16} \cdot 40_{16} &= \text{xtime}(0E) = 1C_{16} \\
57_{16} \cdot 80_{16} &= \text{xtime}(1C) = 38_{16}
\end{aligned}$$

Para obtener el producto de 57_{16} por $b = 83_{16} = 1000\ 0011_2$ sólo queda realizar la suma (OR exclusivo) de los términos que se corresponden con la existencia de «unos» en el multiplicador. Esto es:

$$57_{16} \cdot 83_{16} = 57_{16} \cdot (01_{16} \oplus 02_{16} \oplus 80_{16}) = 57_{16} \oplus AE_{16} \oplus 38_{16} = C1_{16}$$

El elemento neutro respecto a la multiplicación es el 01_{16} y cada elemento del conjunto tiene un **inverso multiplicativo** único que puede obtenerse, entre otros métodos, mediante el algoritmo extendido de Euclides.

3.3. PROBLEMAS DE DIFÍCIL SOLUCIÓN

Los sistemas criptográficos asimétricos o de clave pública suelen estar basados en algoritmos de muy difícil solución para aquella persona (un intruso) que no esté en posesión de determinada información (la clave de descifrado) con cuyo concurso sí es relativamente sencillo encontrar la solución del problema (mensaje en claro).

Así por ejemplo, si el usuario **A**, emisor del mensaje, quiere enviar un mensaje cifrado a un receptor concreto **B** cifrará dicho mensaje usando la clave pública de **B**. Esta operación de cifrado consistirá en una operación matemática de cierta complejidad, pero computacionalmente viable, que será llevada a cabo usando unos recursos informáticos determinados (memoria, capacidad de proceso, etc.) y que se realizará en un tiempo razonable. Como tanto la clave pública de **B** como el algoritmo matemático utilizado son perfectamente conocidos en todo el dominio de seguridad, cualquier otro usuario malicioso **M** podría descifrar el mensaje si es capaz de resolver la operación inversa a aquella con que fue cifrado. Un criptosistema será muy robusto y muy válido para ser utilizado en servicios de seguridad si, computacionalmente, a **M** le resulta prácticamente imposible la resolución de esta tarea.

COMPLEJIDAD DE ALGORITMOS

Obsérvese que decir «prácticamente imposible» es hacer referencia a la incapacidad que un sujeto, el usuario malicioso **M**, tiene para llevar a cabo una determinada operación con los medios informáticos que se le suponen y en un tiempo *razonable*. Conceptos todos estos muy subjetivos y alejados de lo que cabe esperar de una formulación teórica y rigurosamente matemática. Para abordar el problema con el rigor que corresponde se ha elaborado una teoría de la *Complejidad de Algoritmos* que trata de clasificar los problemas y evaluar el tiempo que hipotéticamente conllevaría su resolución.

Como ya se ha dicho, se trata en estos apartados de describir de forma resumida los fundamentos teóricos en que se basan los sistemas criptográficos en que se apoyan los protocolos y servicios de seguridad, principal objeto de estudio de este libro.

Escapa, por tanto, a estos objetivos abordar aquí con un mínimo de rigor los planteamientos y conclusiones de esta teoría de la complejidad. Baste decir que este tipo de problemas, que ha sido estudiado de forma numerosa en este siglo y en los precedentes, se clasifican en *decidibles* e *indecidibles*. Estos últimos serían, en contraposición a los primeros, aquellos para los que no existe un algoritmo que los resuelva. El hecho de que un problema sea *decidible* no significa que sea *factible*, es decir, que se pueda resolver en un tiempo razonable.

Para establecer una medida del tiempo de ejecución de los algoritmos independientemente de la velocidad de computación de la máquina, se ha establecido una referencia en lo que se denomina *operación bit*, que es el proceso necesario para sumar dos bits (cualquiera que sea su valor) teniendo en cuenta el acarreo procedente de una operación anterior. Es fácil demostrar que para sumar dos números de k bits son necesarias k operaciones bit, y que para el producto de un número de k bits por otro de h bits ($k \leq h$) son necesarias del orden (algo menos) de $2 \cdot k \cdot h$ operaciones bit.

Para reflejar el número de operaciones bit que son necesarias para ejecutar un algoritmo se utiliza la notación $O(g(n))$, *Orden de Magnitud*, donde n es el tamaño en bits de la entrada y g es la función, ligada al algoritmo, que se quiere evaluar. Se dice que un algoritmo tiene *complejidad polinomial* si su ejecución depende polinómicamente del tamaño de la entrada y *complejidad exponencial* si el número de operaciones bit necesarias depende exponencialmente del tamaño de la entrada.

Basándose en esto, los problemas decidibles se clasifican en *tratables* e *intratables*. Cuando la complejidad es función de una potencia de n , el número de operaciones a realizar aumenta notablemente cuando se incrementa el tamaño de n , pero cuando la complejidad depende de una expresión exponencial del tipo 2^n , el número de operaciones necesarias, y por tanto el tiempo que debe emplear un ordenador en resolverlas, se dispara hacia cifras astronómicas en cuanto se produce un crecimiento de n .

Otra ordenación en base a la cual encontramos a veces calificados los algoritmos criptográficos es la de considerarlos como *clase P* o *clase NP*. Los primeros serían aquellos algoritmos cuyo tiempo de ejecución puede expresarse de forma polinómica, en tanto que los segundos, que pertenecen a la tribu de los intratables, son *no determinísticos*, es decir, tras cada paso de computación no queda determinado de forma única cuál es el siguiente paso a realizar. Los problemas *NP* (Non Deterministic Polynomial Problems) tienen solución mediante algoritmos cuyo tiempo de resolución es de tipo exponencial. Por último, se denominan problemas *NP-Complejos* a aquellos en los que la determinación de un algoritmo determinista de tipo polinómico para uno de ellos permitiría resolver también el resto de ellos en tiempo polinómico.

ALGUNOS PROBLEMAS DE INTERÉS

A modo de conclusión, para lo que aquí nos interesa, podemos insistir en que existen determinados algoritmos que podemos denominar de *una sola dirección* que son fácilmente computables en la fase de cifrado, mientras que en sentido contrario, al tratarlos de resolver mediante criptoanálisis, se convierten en computacionalmente intratables. Veamos a continuación tres de ellos que son usados en los criptosistemas de clave pública que más adelante se describen.

Problema de factorización

Un problema sencillo es encontrar dos números enteros grandes p y q (generalmente interesará que sean primos) y calcular N tal que $N = p \cdot q$. En cambio, un problema de muy difícil solución, tipo NP, es el siguiente: «dado N , encontrar dos números enteros p y q , mayores que la unidad, que sean primos, tal que $N = p \cdot q$ ». En esa dificultad se apoya la fortaleza de RSA y de otros algoritmos criptográficos. A lo largo del tiempo han sido propuestos y analizados diversos algoritmos de factorización. Cuando N es pequeño, resulta relativamente sencillo encontrar los factores primos, pero cuando N es del orden de miles de bits, resulta una operación sumamente costosa aun utilizando los algoritmos de factorización más avanzados.

Problema de la mochila

Puede enunciarse como:

Dado un conjunto de objetos, cada uno con un peso diferente, ¿es posible seleccionar un subconjunto de ellos para meterlos en una mochila de forma que el peso final de ésta sea exactamente uno previamente determinado? Es un problema del tipo NP-completo y se usó en los primeros algoritmos asimétricos, aunque luego ha sido demostrada su vulnerabilidad.

Problema de algoritmo discreto

Como ya se estudió, en aritmética modular es relativamente sencillo resolver el problema: dado dos enteros a y b menores que n , encontrar x tal que:

$$x = a^b \pmod{n}$$

Sin embargo, el problema inverso, es decir, conocidos x y a menores que n , calcular b de forma que:

$$b = \log_a x \pmod{n}$$

es un problema computacionalmente no abordable.

Multitud de algoritmos de cifrado, tanto para obtener confidencialidad como para conseguir el mecanismo de firma electrónica, se apoyan en esta circunstancia, por lo que, caso de encontrarse algún algoritmo factible que resolviese el problema del cálculo del logaritmo discreto, sería necesario desechar todos ellos y encontrar nuevos algoritmos seguros.

Existen estudios que determinan que el grado de dificultad de los métodos de cálculo del logaritmo discreto es muy similar al que presenta el problema de la factorización, siendo además muy similares los algoritmos que, hasta la fecha, ofrecen más rapidez (digamos mejor, menos lentitud) a la hora de abordar la resolución de ambos problemas.

Otros problemas, como el de la extracción de raíces cuadradas, se encuentran también dentro de este conjunto de algoritmos de una sola dirección y son también usados en los sistemas criptográficos.

CAPÍTULO

4

CRIPTOGRAFÍA SIMÉTRICA, ESTEGANOGRAFÍA Y MARCAS DE AGUA

En contra de lo que el título podría dar a entender, la casi totalidad de este capítulo está dedicada al estudio de los criptosistemas simétricos o de clave secreta. Decimos «casi» porque en la última parte del capítulo se ha incluido el tema de las **Marcas de Agua** y de la **Esteganografía** (este último ya fue comentado en el Capítulo 1) debido a que estos sistemas comparten con aquellos ciertas similitudes en el esquema de simetría del proceso de transformación de datos. Además, a veces, unos y otros pueden ser usados de forma combinada, por lo que con frecuencia suelen producirse confusiones. Por ello, se tratará de dejar constancia de las características que les son comunes y de las grandes diferencias existentes en su concepción.

Como ya quedó dicho antes, la Criptografía es una herramienta básica para la configuración de los Mecanismos de Seguridad en que se apoyan los Servicios de Seguridad. También en los capítulos anteriores se discutió acerca de los conocimientos criptográficos (no excesivamente profundos) que resultan necesarios e imprescindibles para abordar tareas de diseño e instalación de protocolos y sistemas que añadan servicios de seguridad a un determinado servicio telemático.

En la descripción que sigue se aborda, de forma muy resumida, la estructura y funcionamiento de los criptosistemas simétricos que tan relevante papel juegan en el desarrollo de estos protocolos de seguridad. Esta descripción será suficiente para entender el tipo de transformaciones que se realizan y las características principales que de ellas se derivan, pero no se encontrará aquí la especificación completa y detallada necesaria para una hipotética implementación de estos algoritmos (que puede encontrarse en libros y artículos especializados en Criptografía, o en las especificaciones que los definen).

4.1. DE LA CRIPTOGRAFÍA CLÁSICA A LA CRIPTOGRAFÍA MODERNA

Múltiples y bien fundados estudios han puesto de manifiesto que en las distintas organizaciones sociales que han ido surgiendo a lo largo de la Historia, cuando el

nivel de complejidad de las relaciones en ellas establecidas hacía necesario el intercambio de mensajes escritos, también aparecía con frecuencia la necesidad de que algunos de estos mensajes, por las razones que fuese, sólo pudieran ser entendidos por personas concretas y en circunstancias concretas.

Podríamos decir que empieza a hacerse uso de la criptografía siempre que se realiza una transformación del mensaje original de forma que lo convierte en incomprendible para cualquier persona que no esté en posesión de una *información secreta* que es necesaria para devolverlo a su situación original. Esta información secreta es la *clave*, la llave, para abrir esa cerradura que nos da acceso al mensaje en claro.

El estudio de los distintos procedimientos (criptosistemas y algoritmos, podríamos decir) que han ido surgiendo a lo largo de la Historia no es sólo un interesantísimo ejercicio de reflexión sobre el alcance del ingenio humano, sino que, para quien esté interesado en adentrarse en el detenido estudio de la criptografía, le resultará de suma utilidad para entender las bases sobre las que ésta se asienta. Por razones de oportunidad, este estudio no va a ser aquí ni siquiera bosquejado, remitiendo al lector interesado a la consulta de tratados especializados.

Desde el principio, la *sustitución* de unas cosas por otras, caracteres o grupos de caracteres, ha sido una constante en las técnicas criptográficas. La más elemental y simple es la sustitución monoalfabética, que consiste en la sustitución de un carácter o signo por otro. El ejemplo más simple lo constituye el cifrador de César, que consistía en sustituir cada letra por aquella situada tres lugares más adelante dentro de la ordenación marcada por el alfabeto. Otros tipos de sustitución más evolucionados son la *sustitución homofónica*, en la que se establece más de una correspondencia, y la *polialfabética*, en la que cada letra del texto en claro es sustituida por un carácter distinto dependiendo de una o varias claves. Otra técnica también utilizada sistemáticamente en criptografía es la *transposición*, que consiste en alterar el orden de los caracteres en el mensaje.

Aparte de los muchos precedentes existentes en la antigüedad clásica, tanto greco-romana como de otras culturas, fue en el Renacimiento cuando recibió un importante impulso teórico tanto en lo relativo a la forma de realizar el cifrado como en los procedimientos de criptoanálisis. El más conocido de todos es el atribuido a Vigenère (aunque parece que fueron otros sus creadores) basado en el cifrado polialfabético, el cual sirvió de base para las máquinas que se desarrollaron en la primera parte del siglo xx, que estaban constituidas por una serie de rotores que permitían varios pasos consecutivos de cifrado. A este tipo de máquinas pertenecen las que se usaron en las dos guerras mundiales con que fue castigada la población en el pasado siglo.

Sea como fuere, estos ingenios bélicos, además de la estética de sus orígenes, tienen una característica distintiva que los hace poco interesantes para los fines y los procedimientos presentes en la seguridad cívica: el algoritmo es secreto. El que el algoritmo sea secreto es una característica fundamental presente en toda la criptografía clásica, que debiera de constituir un signo diferencial respecto a la criptografía moderna usada para los intercambios de información en que están interesados los ciudadanos. En los sistemas militares, gubernativos y de espionaje se sigue prefiriendo el uso de criptosistemas secretos (pero esta clase de criptografía y de seguridad no va a ser aquí objeto de estudio).

Fue Kerckhoffs, en 1883, quien implantó las bases para lo que hoy podemos denominar **criptografía moderna**: el criptosistema y el algoritmo de cifrado deben ser de conocimiento público, y sólo la clave debe permanecer secreta. La seguridad del sistema reside en la robustez del algoritmo y en la adecuada custodia de la clave para que sólo sea conocida por las personas autorizadas para ello.

Jugando con el ejemplo de la llave y la cerradura, podríamos decir [DiLa98] que el sistema en que esté basada la cerradura puede ser complejo y fácilmente analizable por los especialistas (que de camino pueden hacer una evaluación acerca de su fiabilidad). Por el contrario, la llave debe ser una pieza sencilla, mucho más barata, cuya silueta de dientes y hendiduras es secreta y de la que, además, resulta fácil obtener un duplicado. Además, si bien el funcionamiento interno de la cerradura puede ser complicado, el proceso de abrir y cerrar debe ser rápido y sencillo de ejecutar. Este es el símil que puede servir para los algoritmos y las claves en la criptografía moderna.

Aun a riesgo de abusar del ejemplito de las llaves y las cerraduras, he aquí otro caso que puede servir para reforzar la idea de lo que aquí entendemos por criptografía para la seguridad cívica [Schn96]. Si tomamos una carta y, para evitar que alguien pueda leerla, la metemos en una caja fuerte y escondemos ésta en un sitio complicado y de difícil acceso, eso no es seguridad. Si, por el contrario, la guardamos en la caja fuerte y facilitamos las características constructivas de la caja y permitimos que esta caja y otras cajas iguales de la misma serie de fabricación puedan ser examinadas por expertos de confianza que nos garantizan que sólo puede ser abierta conociendo previamente la clave secreta de apertura y, además, al entregarle la clave al destinatario autorizado éste puede abrir la caja y leer la carta, entonces sí estamos hablando de seguridad.

Con estos precedentes y apoyándose en el desarrollo de los sistemas electrónicos y de la informática, fue concebido el criptosistema DES, que más adelante se comenta con cierto nivel de detalle. Al igual que sus precedentes, este primer criptosistema obedece al esquema de lo que conocemos como *criptografía simétrica* o *criptografía de clave secreta*. Al estudio de estos criptosistemas está dedicado este capítulo.

CARACTERÍSTICAS PRINCIPALES DE LA CRIPTOGRAFÍA SIMÉTRICA

Recordemos que, en este caso, la clave de cifrado y la usada para el descifrado son la misma (o la una puede deducirse fácilmente de la otra). La Figura 4.1 refleja el esquema básico que ya se planteó en el Capítulo 1. Con este presupuesto, el proceso de obtención del mensaje cifrado c es:

$$c = E_k(m)$$

Donde k es la clave secreta, E_k representa la operación de cifrado con esa clave y m es el mensaje en claro.

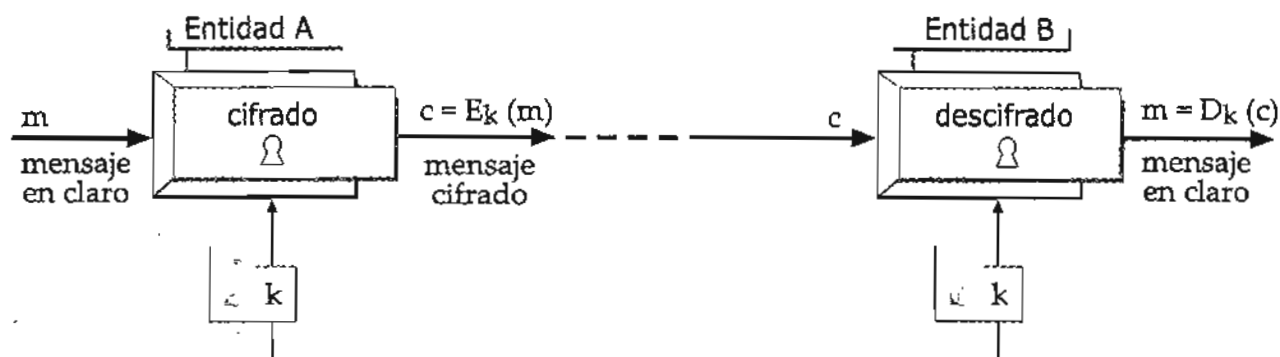


Figura 4.1. Esquema de la criptografía simétrica.

Simétricamente, el proceso de recuperación del mensaje original, mediante la operación de descifrado, D_k , en el que se utiliza la misma clave secreta k , se representa como:

$$m = D_k (c)$$

La simetría del proceso se basa, por tanto, en que este tipo de cifradores se comporta tal que:

$$D_k [E_k (m)] = m$$

En contraposición a los algoritmos de clave pública, que se basan en operaciones matemáticas de cierta complejidad, los algoritmos de clave secreta se apoyan en operaciones lógicas o algebraicas muy simples, lo que conlleva que el tiempo necesario para realizar el proceso completo de cifrado es muy pequeño. Esta característica hace que este tipo de cifrado pueda utilizarse para proporcionar confidencialidad en aplicaciones telemáticas de banda ancha, en las que se requiere gran velocidad en el flujo de datos, circunstancia esta que no se puede soportar utilizando criptosistemas de clave pública.

Como contrapartida, la filosofía de funcionamiento de la criptografía simétrica implica que, para establecer comunicación segura entre dos usuarios, es necesario que éstos conozcan y compartan una clave simétrica concreta. Si fuesen tres los participantes en el entorno de comunicación y se quisiera salvaguardar la confidencialidad dos a dos, cada usuario necesitaría conocer y custodiar dos claves distintas, siendo tres el número total de claves puestas en juego en el dominio. Cuando el número de usuarios crece y crece (como es el caso de los que tienen acceso a Internet y pueden formar parte de un hipotético dominio de seguridad) el número de claves necesarias sería enorme e inmanejable.

Para complicar más las cosas, la fortaleza de los criptosistemas de clave secreta es limitada, siendo mucho más vulnerables a los ataques que los criptosistemas de clave pública. No sólo mediante los ataques por fuerza bruta sino usando los denominados *criptoanálisis diferencial* y *criptoanálisis lineal* es susceptible de ser vulnerada la seguridad de estos criptosistemas. Esto determina que para garantizar la seguridad del conjunto es necesario renovar de cuando en cuando las claves, para así dificultar la labor de los posibles atacantes. Si, además de ser muchísimas las claves puestas en juego, hay que gestionar su reparto y su actualización implantando una adecuada gestión de claves (*key management* en inglés), el uso de estos criptosistemas se convierte en prácticamente imposible, al menos en sistemas abiertos en los que pueda ser muy elevado el número de los hipotéticos participantes en una comunicación y en el que, además, éstos puedan pertenecer a organizaciones y entornos distintos y heterogéneos. A esto hay que añadir que las claves secretas habrá que enviarlas por canales seguros (posiblemente extratelemáticos), ya que si son interceptadas en el proceso de distribución, todo el tinglado de seguridad en que se basa el sistema global se vendría abajo.

En resumen, de cara a los requisitos necesarios para la seguridad en redes telemáticas, podemos señalar que:

- a) Debido a la velocidad de ejecución, los criptosistemas simétricos son **imprescindibles** para proporcionar servicios de confidencialidad en las aplicaciones telemáticas de alta velocidad.
- b) Debido a su principio de funcionamiento, el número de claves necesarias para proporcionar confidencialidad en un entorno, en el que participen un número

considerable de entidades comunicantes, es tan elevado que hace prácticamente **inaplicable** el uso exclusivo de criptosistemas simétricos en estos escenarios.

- c) La mayor vulnerabilidad de estos criptosistemas hace necesaria la renovación periódica de las claves, lo que añadido a lo dicho en el párrafo anterior, hace **impracticable** una adecuada política de distribución y renovación de claves.

Este «ni contigo ni sin ti tienen mis males remedio» se resuelve, como ya se apuntó en el Capítulo 1 y se desarrollará más adelante, con el uso combinado de ambos tipos de criptografía (simétrica y de clave pública) a la hora de configurar las PDUs que se intercambien en los protocolos de seguridad que se apliquen.

CIFRADORES DE FLUJO Y CIFRADORES DE BLOQUE

Dado un mensaje en claro m , más o menos largo, representado mediante un número variable de octetos, a la hora de proceder a su cifrado para convertirlo en un criptograma c existen dos tipos básicos de cifradores, diferenciados en cuanto a cómo dividir el mensaje en claro para abordar la tarea de transformación. Estos dos procedimientos son:

- Procedimiento de cifrado en flujo.
- Procedimiento de cifrado en bloque.

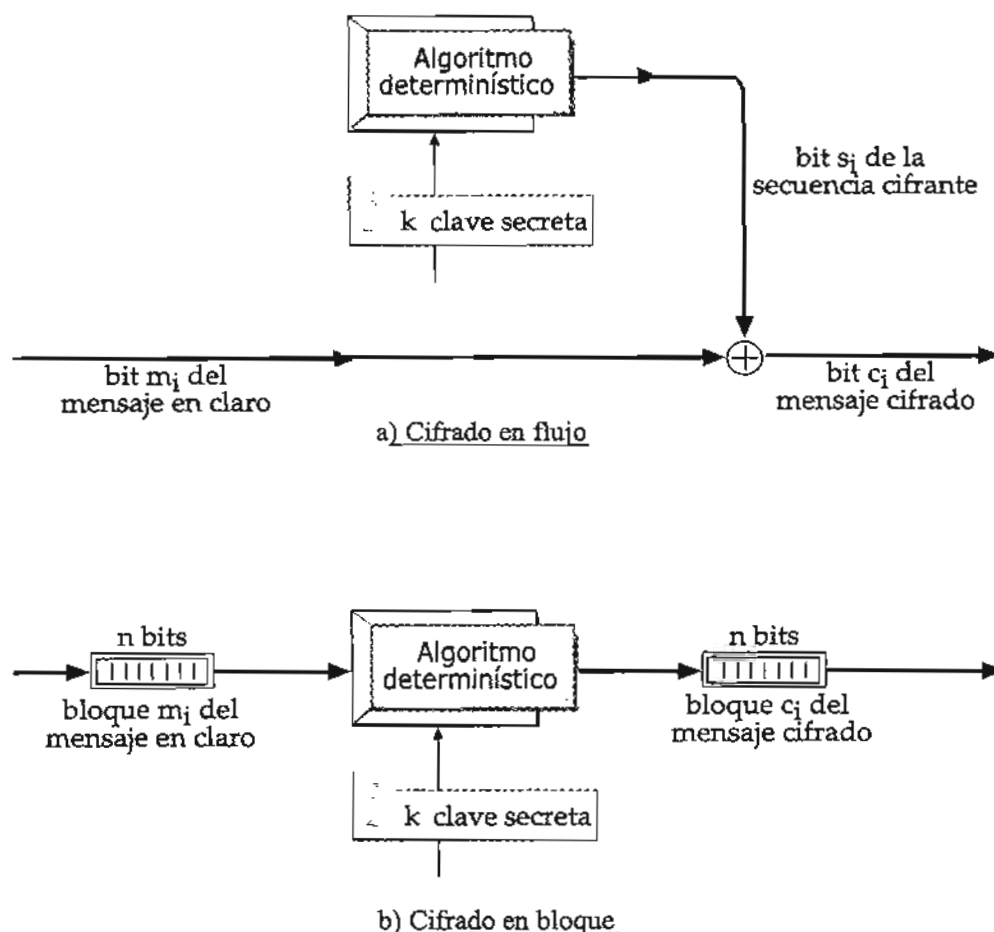


Figura 4.2. Procedimientos de cifrado.

El primero de ellos, Figura 4.2 a), está directamente entroncado con los cifradores clásicos y es por eso que parece como más «natural». Consiste en cifrar bit a bit, mediante una operación de suma módulo 2 (XOR), los bits del mensaje m con los correspondientes de una *secuencia cifrante* de bits generada por un sistema que opera mediante un *algoritmo determinístico* conforme al valor de una *clave secreta* de tamaño predeterminado y no muy larga. El procedimiento de descifrado y las características de estos cifradores se abordan en el epígrafe 4.8.

El segundo tipo, el procedimiento de cifrado en bloque, se representa en la Figura 4.2 b). En este caso, se divide el mensaje en claro en bloques de n bits cada uno (en los cifradores más «clásicos» los bloques constan de 64 bits). La característica principal de este tipo de cifradores consiste en que cada bloque se cifra de igual forma, independientemente del lugar que ocupe en la cadena, de manera que todos los bits del bloque se cifran conjuntamente, participando en operaciones que tratan de oscurecer las posibles relaciones que tuviesen en el mensaje original.

En el siguiente apartado se estudia la forma en que estos bloques de entrada y de salida se interrelacionan para constituir el mensaje cifrado final (son los denominados *modos* de funcionamiento) y en los siguientes apartados se aborda la descripción de los algoritmos más representativos.

4.2. CIFRADO EN BLOQUE: MODOS DE OPERACIÓN

Continuando con el esquema representado en la Figura 4.2 b), cabe preguntarse cómo organizar los bloques de salida c_i para constituir el criptograma final c , dicho de otro modo, de qué forma pueden relacionarse entre sí los bloques de entrada y de salida para la obtención del mensaje cifrado. Para ello se han definido cuatro modos distintos:

- Libro Electrónico de Códigos, ECB (*Electronic Code Book*).
- Encadenamiento de Bloques Cifrados, CBC (*Cipher Block Chaining*).
- Realimentación de Mensaje Cifrado, CFB (*Cipher Feedback*).
- Realimentación de Salida, OFB (*Output Feedback*).

Modo ECB

Es el modo de procedimiento más simple y el más fácil de implementar. También es el menos seguro. Consiste en que los $m_1, m_2, \dots, m_m$ bloques en que se ha dividido el mensaje se van pasando, uno detrás de otro, como entrada del cifrador y los $c_1, c_2, \dots, c_m$ resultantes se concatenan uno detrás de otro, y en el mismo orden, para constituir el mensaje cifrado final, c . Para conseguir un número m exacto de bloques hay que recurrir al rellenado (*padding*) con cadenas de bits predeterminadas.

El nombre de Libro Electrónico de Códigos tiene que ver con el uso de libros de códigos que se hacía en la criptografía clásica. Siendo n el número de bits del bloque, para cada clave podría hacerse, en teoría, un libro electrónico donde figurasen todos los mensajes posibles, 2^n , y sus correspondientes criptogramas. Esta especie de trabajo previo habría que hacerlo, además, para cada clave. Si tenemos en cuenta que en los algoritmos más implantados (como DES e IDEA) $n = 64$ y que las claves constan de un número elevado de bits (56 en DES y 128 en IDEA) este planteamiento es un desatino. Por ello, se recomienda olvidarse de su nombre completo y recordarlo solamente como ECB, o como método simple.

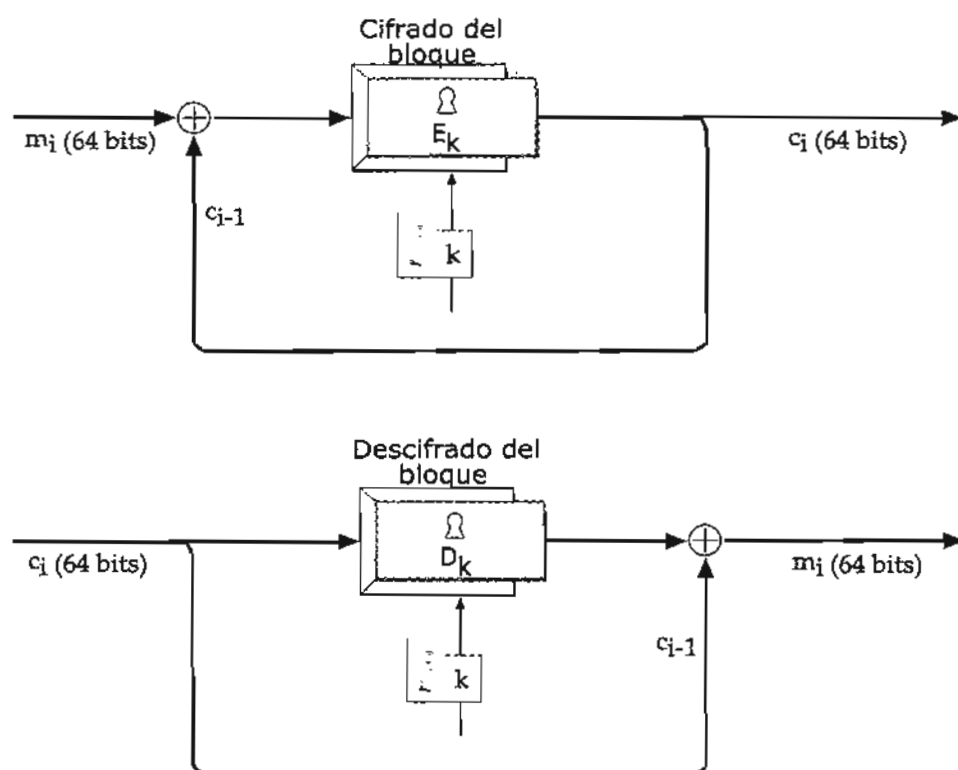


Figura 4.3. Modo CBC.

Como cada bloque está cifrado independientemente, el descifrado también puede realizarse separadamente para cada bloque c_i , con lo que se reconstruye fácilmente el mensaje original. Es el más fácil de atacar, sobre todo si hay cierta información previa sobre el contenido de los bloques. Por eso se recurre a los otros *modos* que ofrecen mayor resistencia al criptoanálisis.

Modo CBC

Para robustecer la seguridad del cifrado, el modo de encadenamiento de bloques cifrados es tal que el resultado c_1 de cifrar el primer bloque se suma módulo 2 con el bloque m_2 . El resultado de esta operación XOR constituye la entrada de la operación de cifrado que da lugar al bloque c_2 , y así sucesivamente, como se representa (de forma genérica) en la Figura 4.3. En esta figura se ha representado un bloque de 64 bits que se corresponde con el caso de los algoritmos más antiguos (como pueden ser DES e IDEA), pero el modo CBC puede aplicarse igualmente a otros algoritmos más avanzados en los que el tamaño del bloque sea bastante mayor que 64 bits.

Se puede demostrar, con poca dificultad, que el esquema de descifrado, que también se recoge en dicha figura, reconstruye exactamente el mensaje en claro $m_1, m_2, \dots, m_m$. Algo más complejo es demostrar el aumento de seguridad conseguido, que evita alguno de los riesgos inherentes al modo ECB.

Modo CFB

En la Figura 4.4 se recoge el esquema de funcionamiento del proceso de cifrado. (El esquema de descifrado presenta una total simetría con el representado en esta figura.)

El procedimiento sería igualmente válido para cualquier número n de bits del bloque (distinto de los 64 que aparecen en la figura). El tamaño p del segmento del criptograma que se realimenta puede, asimismo, ser distinto de un octeto. No obstante, en la figura se han reflejado exactamente esas cifras para mayor sencillez de interpretación y porque son las utilizadas en los algoritmos más representativos.

Inicialmente, es necesario cargar el registro de desplazamiento con un valor que debe ser lo más aleatorio posible, aunque no es necesario que sea secreto. (En el modo antes descrito, el CBC, también es necesaria una inicialización semejante del registro auxiliar que sirve para realimentar la entrada al cifrador.)

El mensaje en claro se divide en octetos, m_i , que dan lugar al correspondiente octeto del mensaje cifrado, c_i , tras sumarse, módulo 2, con los ocho bits más significativos del registro de 64 bits resultante del cifrado del contenido del registro de desplazamiento. En cada paso, el octeto c_i de salida pasa a ocupar los ocho bits menos significativos del registro, mediante un desplazamiento de éste de ocho lugares a la izquierda. De esta forma el contenido del octeto de salida formará parte del proceso de cifrado del siguiente octeto del mensaje.

Quizás la característica más llamativa de este modo de cifrado podría ser que, bajo el modo CFB, el cifrador se comporta de forma parecida a un cifrador de flujo, con la particularidad de que la generación del mensaje cifrado se realiza octeto a octeto. Si en lugar de un octeto se utiliza un número menor de bits, todavía se acentúa más esta similitud con los cifradores de flujo. Si se emplea en una aplicación en la que el resultado del cifrado se esté transmitiendo directamente a través de una red de comunicación de datos, este modo de cifrado reduce a un octeto el retardo en la generación de símbolos a ser transmitidos. De aquí suele deducirse a veces que este modo de cifrado es particularmente idóneo para aplicaciones de seguridad en redes telemáticas, aunque la realidad es que (como puede comprobarse en los sucesivos capítulos) los problemas y las soluciones que se presentan en este tipo de aplicaciones tienen más que ver con la complejidad de los elementos y de los agentes de comunicación en ellas presentes que con la forma concreta de realizar un cifrado con claves secretas.

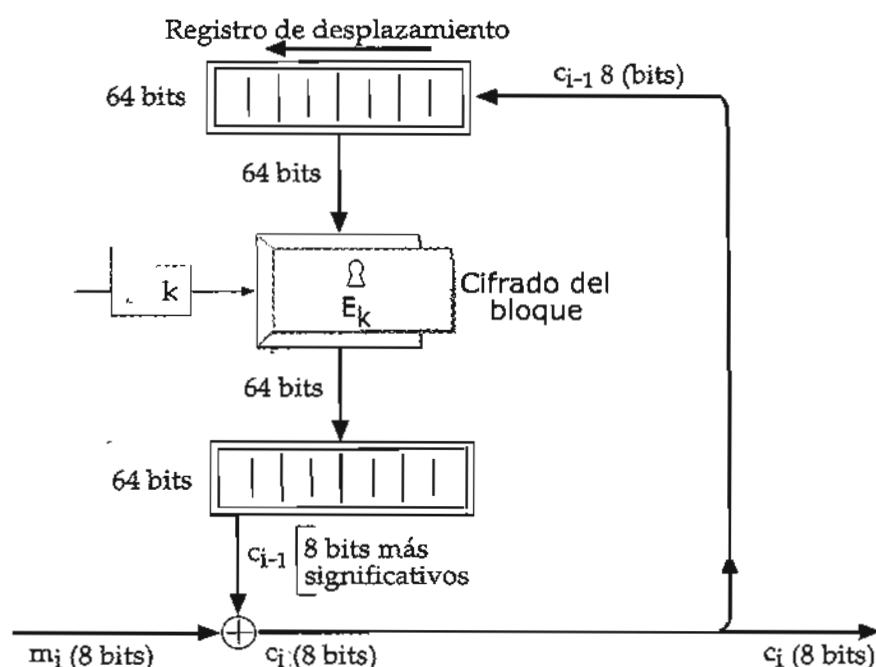


Figura 4.4. Modo CFB.

Modo OFB

Su esquema de funcionamiento (Figura 4.5) es muy parecido al de CFB antes descrito. La diferencia consiste en que el octeto que realimenta al registro de desplazamiento se obtiene directamente de la salida del cifrador de bloque, antes de proceder a la suma módulo 2 con el octeto de entrada, m_i , del mensaje en claro. Puede observarse que los nombres con que se designan estos dos modos tan semejantes (*realimentación de mensaje cifrado* para el CFB y *realimentación de salida* para el OFB) no son excesivamente clarificadores.

En ambos modos, un mismo mensaje en claro puede dar lugar a mensajes cifrados distintos, aun usando la misma clave k , si se cambia el valor inicial del registro de desplazamiento (otra característica típica de los cifradores de flujo). Este valor, que no es necesario que sea secreto, sí debe ser tan aleatorio como sea posible.

Caso de que el mensaje cifrado vaya a ser transmitido por una línea de comunicación sin recuperación de errores y apareciesen errores de transmisión, cada bit recibido erróneamente afectará, en el proceso de descifrado, a varios octetos consecutivos (lo que se denomina la *propagación del error*). Esta situación afecta a todos los modos de cifrado en los que, de una u otra forma, se produce realimentación hacia la entrada de parte del criptograma de salida. Es fácil de demostrar que en el modo de encadenamiento de bloques (CBC) este error produce 16 octetos erróneos consecutivos, mientras que en el CFB afecta a sólo ocho octetos consecutivos y en el OFB el error se limita al bit afectado.

Esta constatación podría inducir a pensar que el modo OFB es el que mejor se adapta a las aplicaciones de transmisión de datos. No obstante, esa forma de plantear las cosas queda desdibujada si pensamos en comunicaciones telemáticas mediante protocolos en los que los datos viajen cifrados. En estos casos, no cabe hablar de propagación de error porque esos datos formarán parte de las PDUs de un protocolo seguro encastrado en una arquitectura jerarquizada y organizada en niveles, donde las

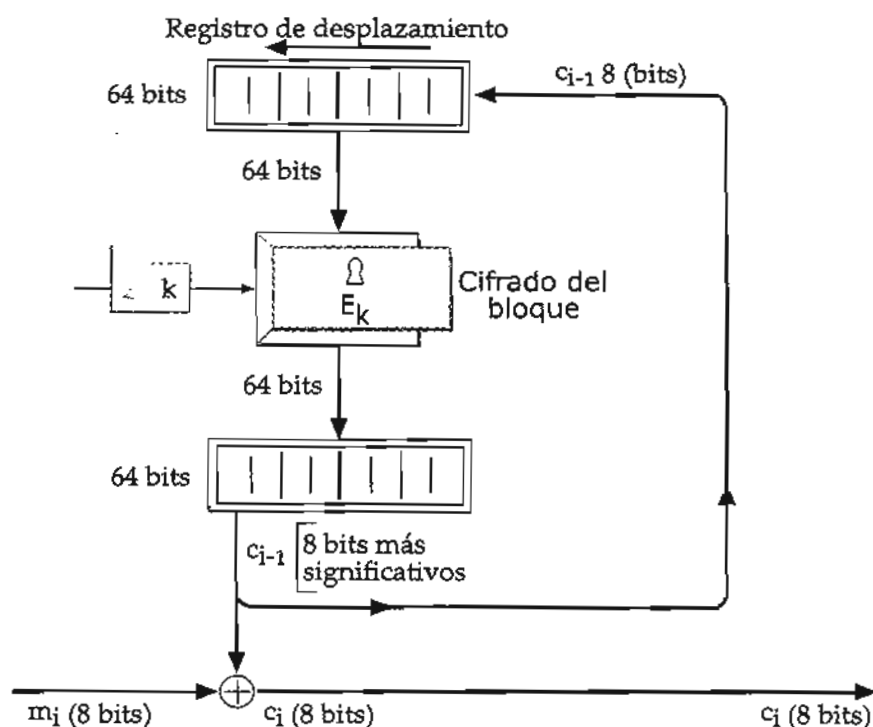


Figura 4.5. Modo OFB.

distintas funciones de comunicación (incluidas las de detección y corrección de errores) estarán atribuidas y serán llevadas a cabo por protocolos de nivel especializados en cada una de ellas.

4.3. CRIPTOSISTEMA DES

El DES, *Data Encryption Standard*, puede ser considerado como el primer algoritmo de la criptografía moderna usado masivamente en procedimientos civiles, tanto personales como comerciales. También conocido como DEA (*Data Encryption Algorithm*) por ANSI y normalizado por ISO como DEA-1, ha venido desempeñando desde 1977 un papel relevante en la implantación de la criptografía, en general, y, muy especialmente, de los servicios de seguridad en las redes telemáticas. Aunque actualmente puede ser considerado como un algoritmo anticuado que ha sido vulnerado en múltiples ocasiones, usado en determinadas condiciones, aún puede ser de utilidad en muchas aplicaciones.

Basado en un diseño original de IBM, surgió como resultado de una propuesta de los organismos de normalización norteamericanos que buscaban un algoritmo con alto nivel de seguridad, que estuviese completamente especificado, susceptible de ser validado, cuya seguridad residiese sólo en la clave, fácil de entender, disponible para todos los usuarios, económicamente asequible y fácil de implementar. Es decir, las condiciones que ha de tener un algoritmo útil para lo que aquí hemos denominado seguridad cívica, con el añadido de bueno, bonito y barato.

La NSA (*National Security Agency*), organismo norteamericano encargado de la seguridad nacional (de protección contra los enemigos y esas cosas), que mantenía la idea de considerar la criptografía como un arma de valor militar y estratégico, no apta para dejarla en manos de civiles, intervino sobre la propuesta inicial de IBM acortando el tamaño de la clave, que pasó de los 112 bits que tenía en el diseño original a sólo 64, y modificando la estructura de la parte más sensible del algoritmo, las denominadas «cajas S» que luego comentaremos. Esto generó una gran controversia ante el temor de que el acortamiento de la clave facilitase la ruptura por parte de una organización con muchos recursos, como es el caso de la NSA. También existía la sospecha de que al cambiar parte del algoritmo hubiesen introducido una puerta falsa, una *trapdoor*.

Sea como fuere, en 1977 el estándar [ANSI X3.92] vio la luz y, aunque con las limitaciones reseñadas, empezó su andadura con notable éxito, hasta el punto de que el periodo de validez que como estándar tenía se vio reiteradamente prorrogado.

FUNCIONAMIENTO DE DES

El texto original, o texto en claro que se desea cifrar, se divide en bloques de 64 bits cada uno:

$$m = m_1, m_2, \dots, m_n$$

La clave k de cifrado es de 64 bits, es decir, ocho octetos, de los cuales ocho bits (el LSB de cada octeto) son para control de paridad, por lo que el tamaño real de la clave a efectos criptográficos es de 56 bits.

Se basa en la combinación de las dos técnicas básicas de la criptografía simétrica: confusión y difusión. La primera de ellas consiste en establecer una dependencia

funcional lo más compleja posible entre la clave y el mensaje en claro, y se consigue mediante la *sustitución* de unos símbolos por otros. La segunda consiste en procurar dispersar a lo largo del criptograma las propiedades estadísticas del mensaje en claro, y en DES se consigue mediante la *transposición* (permutación de los bits).

En la Figura 4.6 se representa el esquema general del algoritmo. Las operaciones que se realizan utilizan una aritmética binaria convencional y unas operaciones lógicas con números de 64 bits, como máximo, de manera que fue fácilmente implementable con la tecnología electrónica existente a finales de los setenta, para lo que se desarro-

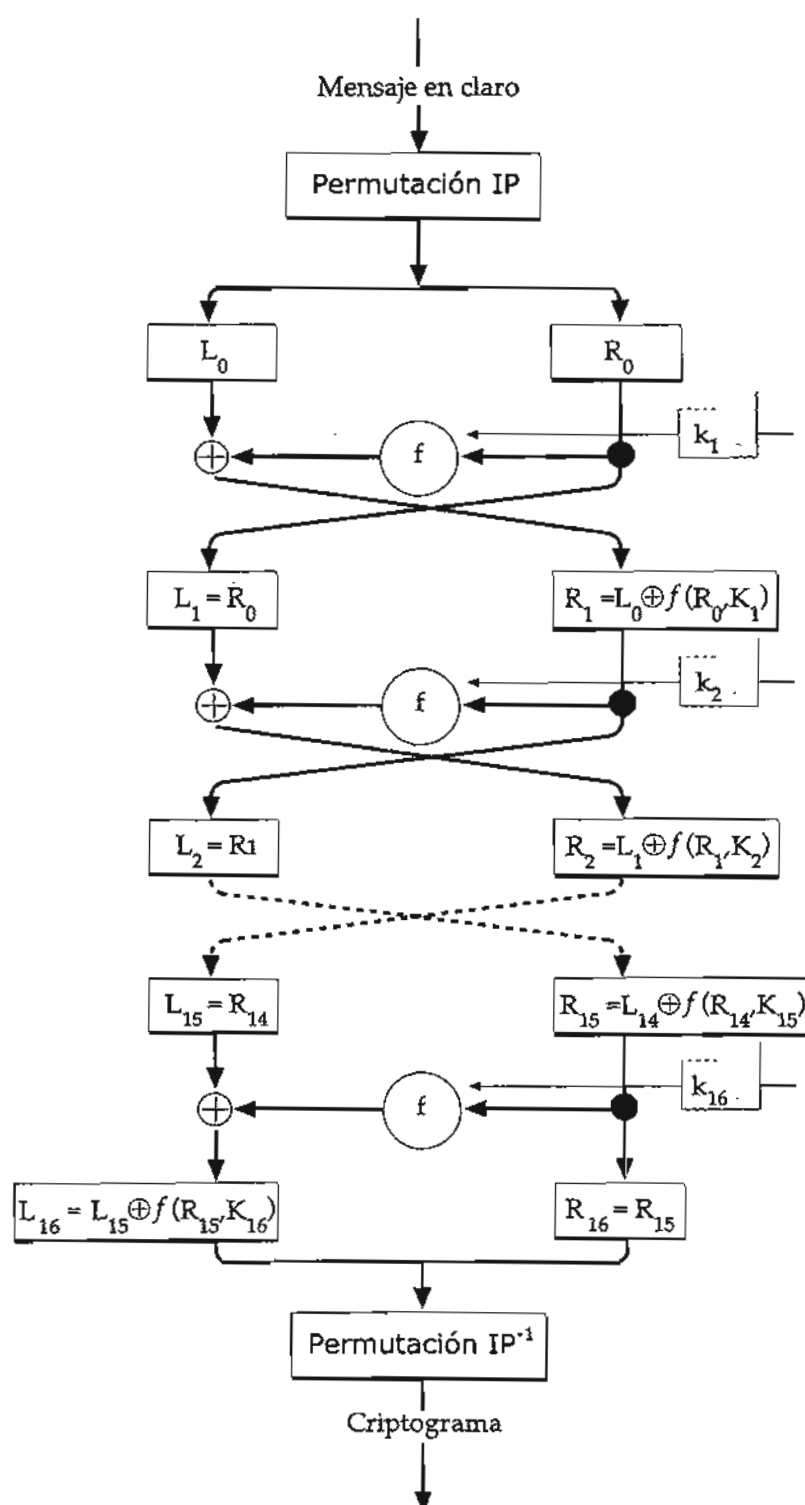


Figura 4.6. Esquema de funcionamiento del algoritmo DES.

llaron circuitos integrados de propósito específico. Es también muy sencillo de implementar mediante un programa, sin más que seguir punto por punto sus especificaciones. Como puede observarse, consta de una permutación inicial, seguida de 16 ciclos o rondas (*rounds*) en los que actúan otras tantas subclaves, para terminar con una permutación inversa de la inicial. Esta permutación se realiza conforme a una tabla preestablecida (que se puede encontrar en cualquiera de los tratados de criptografía que antes se han referenciado) y, según parece, no aporta seguridad al procedimiento, con lo que bien pudiera suprimirse sin que ello signifique disminución de la robustez del algoritmo, siendo una incógnita la razón por la cual se incluyó.

La **estructura de un ciclo**, según se recoge en la Figura 4.7, es tal que en cada uno de ellos el bloque m_i se subdivide en dos mitades de 32 bits cada una. Al bloque R_{i-1} se le somete a una permutación extendida conforme a una tabla, de forma que algunos bits se repiten, resultando una salida de 48 bits. Por otro lado, los 56 bits de la clave se parten en dos mitades, cada una de las cuales es desplazada a la izquierda uno o dos lugares según el ordinal del ciclo. De esos 56 bits se obtiene una *subclave* de 48 bits mediante una permutación con compresión, resultando que cada subclave usa un subconjunto diferente de bits de la clave inicial.

Esta subclave de 48 bits se suma, mediante una operación XOR, con los 48 bits resultantes de la permutación con expansión con que se transformó la parte derecha del bloque resultante del ciclo anterior. Posteriormente, a esta pieza de información se la somete a una transformación, mediante una operación de sustitución (definida mediante tablas), en las denominadas **Cajas-S** (*S-boxes, substitutions boxes*). Para ello, los 48 bits se dividen en ocho grupos de seis bits cada uno. Cada grupo de seis bits es la entrada de cada una de las ocho Cajas-S existentes, de forma que cada una de ellas tiene seis bits de entrada y cuatro bits de salida, resultando, por tanto, a la salida,

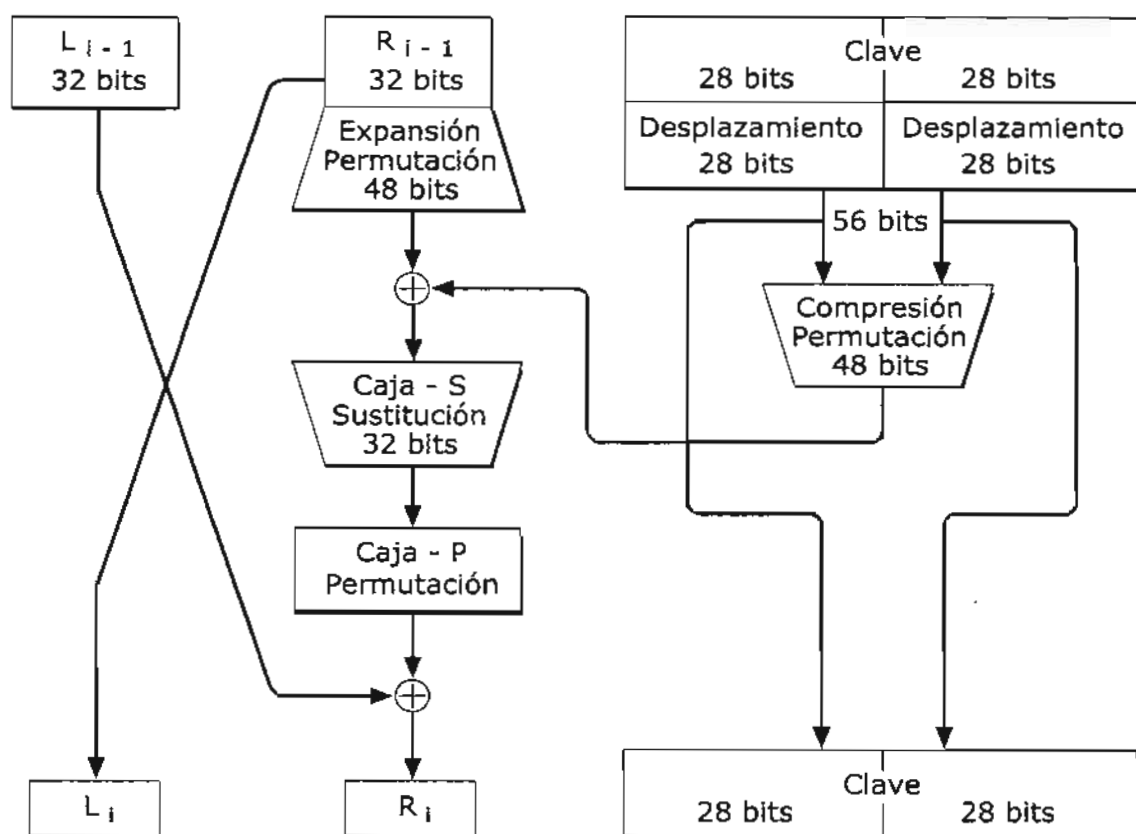


Figura 4.7. Estructura de un ciclo DES.

una pieza de 32 bits, que es la que, tras una nueva permutación simple, produce el resultado final de las cuatro operaciones que en la Figura 4.6 se resumen como *Función f*. La salida de la *Función f* se combina, mediante una operación XOR, con la mitad izquierda del bloque proveniente del ciclo anterior. El resultado de estas operaciones se convierte en la nueva mitad derecha, mientras que la antigua mitad derecha pasa a ser la nueva mitad izquierda.

Si L_i y R_i son las mitades izquierda y derecha del bloque resultante de la iteración i -ésima, K_i es la clave de 48 bits que se aplica en el ciclo i , y f es la función que representa las citadas sustituciones, permutaciones y XOR, el resultado del ciclo es:

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1}, K_i) \end{aligned}$$

Para los criptoanalistas, es en estas Cajas-S donde reside la robustez del algoritmo, ya que, si bien las otras operaciones son lineales y fáciles de analizar, las transformaciones que se realizan en Cajas-S son no-lineales y son las que verdaderamente proporcionan la seguridad de DES.

Estas operaciones se repiten 16 veces, constituyendo los 16 ciclos de DES. Después del último ciclo, las mitades derecha e izquierda no se intercambian. El proceso termina aplicando al bloque concatenado $R_{16}L_{16}$ una permutación final inversa de la inicial.

LAS CLAVES Y EL DESCIFRADO

Para el descifrado se usa el mismo algoritmo y la misma clave. La única diferencia consiste en que las subclaves han de usarse en orden inverso. Esto es, si las subclaves utilizadas en el proceso de cifrado son $k_1, k_2, k_3, \dots, k_{16}$, entonces, para el proceso de descifrado deberán usarse, en este orden, las subclaves $k_{16}, k_{15}, k_{14}, \dots, k_1$. Por eso, aunque la clave de descifrado no es *exactamente* la misma que la de cifrado, es directamente extraíble de ésta, que es la condición exigible a un algoritmo simétrico.

Debido a la forma en que se extraen las subclaves que se aplican en los distintos ciclos del algoritmo, ciertas claves no aportan seguridad suficiente al procedimiento, por lo que son llamadas *claves débiles*. Si todos los bits de cada mitad son 0 o 1, la clave utilizada para cada valor del algoritmo es la misma en todos los ciclos. Esto conlleva a que, de todas las combinaciones posibles que pueden obtenerse con los 56 bits de la clave, hay cuatro de ellas totalmente inservibles.

Del mismo modo, es sencillo comprobar que algunas parejas de claves, al cifrar el texto en claro dan lugar a idéntico criptograma, lo que significa que una de las claves de ese par puede servir para descifrar mensajes encriptados con la otra clave del par. Esto da lugar a la existencia de 12 claves denominadas *claves semidébiles*. Además existen otras 48 claves que, debido al tipo de desplazamientos utilizados, sólo generan cuatro subclaves diferentes. Todas estas combinaciones constituyen un conjunto de 64 subclaves no hábiles (cuyo listado puede encontrarse en cualquier tratado de criptografía) que deben excluirse a la hora de seleccionar una clave al azar. Esta merma en el número de claves posibles es absolutamente insignificante comparada con la cifra que resulta de calcular 2^{56} .

Por último, es fácil de demostrar que, debido al uso de operaciones XOR, si se toma el complementario de una clave, es decir, se sustituyen los 0 por 1 y los 1 por 0 y se cifra con ella el complementario del texto en claro se obtiene:

$$\begin{aligned} E_k(P) &= C \\ E_{k'}(P') &= C' \end{aligned}$$

Ello significa que en un ataque del tipo *texto en claro elegido* sólo es necesario probar la mitad de las posibles claves, es decir, 2^{55} en lugar de 2^{56} . Además de esta posible deficiencia, se han encontrado muchas otras debilidades de DES y se han realizado múltiples estudios acerca de los posibles ataques a este algoritmo. Aunque, a primera vista, el número de claves posibles pudiera parecer muy elevado (más de 72 seguido de 15 ceros, en decimal) en la actualidad el número de bits de la clave resulta manifiestamente insuficiente para proporcionar un nivel de seguridad adecuado. Otros algoritmos más modernos, como el IDEA que se describe en el siguiente apartado, lo superan con creces tanto en este como en otros aspectos directamente relacionados con su estructura interna.

No obstante, como ya se ha apuntado en el Capítulo 1, y se justificará más adelante, en casi todos los protocolos de seguridad se aplica el uso combinado de criptografía de clave pública y criptografía simétrica con *claves de un solo uso*, por lo que las vulnerabilidades que pueda presentar el algoritmo de clave secreta quedan cubiertas por la robustez del procedimiento global, lo cual significa que para multitud de usos civiles de la Seguridad en Redes se podrá seguir usando el DES durante bastante tiempo con un nivel de seguridad aceptable. Además, la fortaleza del criptograma se puede aumentar de forma notable, según se describe más adelante, repitiendo varias veces el proceso de cifrado.

4.4. ALGORITMO IDEA

Es un cifrador europeo que fue presentado por sus autores, X. Lai y J. Massey, en 1990 bajo el nombre de PEES (*Proposed European Encryption Standard*) y que, tras algunas mejoras, en 1992 pasó a denominarse con el nombre de IDEA (*International Data Encryption Standard*). Es un algoritmo muy robusto que se hizo popular por haber sido adoptado por PGP, una aplicación de correo electrónico seguro de uso muy extendido.

Se trata de un cifrador simétrico en el que tanto el mensaje en claro como el criptograma se encuentran divididos en bloques de 64 bits, como en el DES, pero utilizando una clave de 128 bits, lo que representa un incremento notabilísimo del espacio de claves, que pasa a ser de $2^{128} = 3,4 \cdot 10^{38}$. Esta es ya una cifra muy elevada, lo que conlleva un aumento considerable de su seguridad si se lo compara con DES, dificultando mucho un ataque exitoso por fuerza bruta.

El algoritmo [LaMa91] consiste en dividir el bloque m_i del mensaje que se desea cifrar en cuatro subbloques de 16 bits cada uno. A continuación se aplican una serie de operaciones aritméticas de grupos algebraicos diferentes, que además son fácilmente implementables tanto en hardware como mediante programas escritos en lenguaje de máquina o escritos usando lenguajes de alto nivel. El hecho de que no exista nada equivalente a las misteriosas Cajas-S de DES y que las operaciones sean todas puramente aritméticas, sin permutaciones ni compresiones de registros conforme a lo establecido en tablas, hace que el algoritmo sea más rápido y eficaz, al tiempo que el criptoanálisis puede ser enfocado de forma más rigurosa.

Las operaciones a realizar son:

1. Suma módulo 2, OR exclusivo, de X e Y. Lo representamos como $X \oplus Y$.

2. Suma módulo 2^{16} de X e Y . Lo representaremos como $X \oplus Y$.
3. Multiplicación módulo $2^{16} + 1$ de X e Y . Lo representamos como $X \otimes Y$.

Todas estas operaciones son invertibles en sí mismas, lo que posibilita que se pueda usar el mismo algoritmo para cifrar y para descifrar. En efecto, además de la ya bien conocida operación XOR, la suma módulo 2^{16} , que es la que se consigue sumando los contenidos de registros de 16 bits despreciando los acarreo, son operaciones en las que están definidos y son únicos los inversos respecto a la suma y existe el correspondiente elemento neutro. De semejante manera, con la multiplicación módulo $2^{16} + 1 = 65.537$, por ser éste un número primo, se asegura (como vimos en el capítulo anterior) la existencia de inversos en dicho grupo multiplicativo.

En la Figura 4.8 se representa el comportamiento del algoritmo. Cada bloque m_i de 64 bits se divide en cuatro subbloques $m_{i1} \dots m_{i4}$ de 16 bits cada uno. En el primer

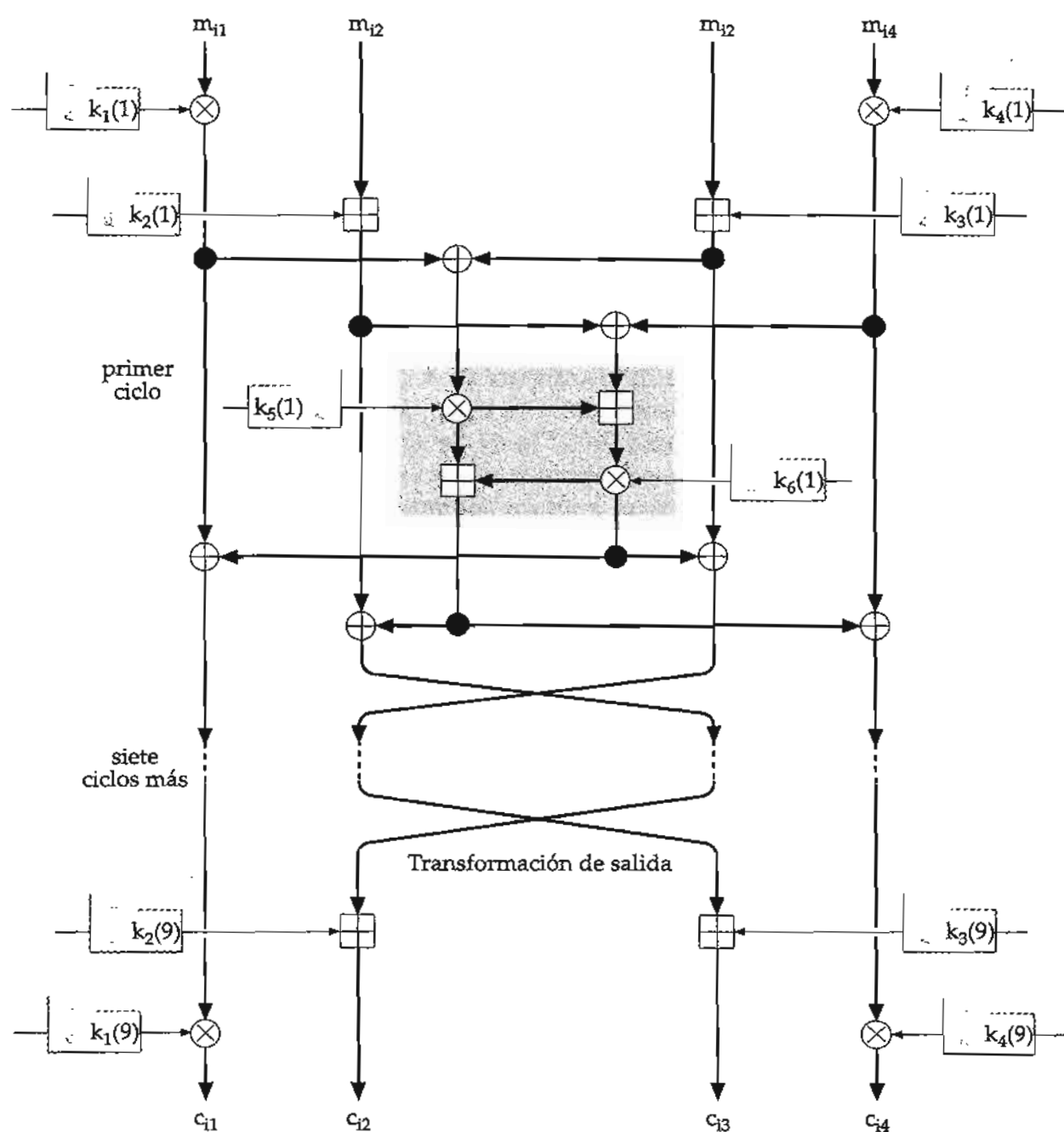


Figura 4.8. Esquema de funcionamiento del algoritmo IDEA.

ciclo actúan seis subclaves de 16 bits, obtenidas a partir de la clave k de 128 bits según un procedimiento que se comenta más abajo. En la figura se describen las operaciones que se realizan a partir de estos subbloques hasta que se generan las cuatro salidas del primer ciclo.

Esta misma operación se repite otras siete veces más utilizando la salida de un ciclo como entrada del siguiente y aplicando en cada uno de ellos un nuevo juego de seis subclaves. Al final del séptimo ciclo se aplica la *transformación de salida* indicada en la figura, dando como resultado los cuatro subbloques del criptograma c_i .

Al igual que en DES, las técnicas utilizadas en IDEA para oscurecer el mensaje son la confusión y la difusión, aunque son aquí de efectos más contundentes que en aquél, resultando ser muy resistente frente a ataques por análisis diferencial. Se ha determinado que el módulo central (delimitado por la zona sombreada en la Figura 4.8) es el que provoca la mayor difusión de los bits, de forma tal que cada bit de salida depende de todos los bits de entrada y de las subclaves.

Las subclaves se obtienen, como queda dicho, a partir de la clave global de 128 bits. Como son necesarias seis subclaves para cada ciclo, se necesitan 48 subclaves para los ocho ciclos, a las que hay que añadir cuatro subclaves más para la transformación de salida, lo que representa un total de 52 subclaves para todo el proceso. El esquema a seguir es muy sencillo: a partir de la clave original de 128 bits se obtienen, por segmentación, exactamente seis subclaves de 16 bits. Luego se gira la clave 25 bits a la izquierda y se segmenta de nuevo, repitiendo el proceso hasta obtener las subclaves necesarias. Este procedimiento de obtención de subclaves da lugar a un conjunto muy variado de claves de 16 bits, con una gran dispersión de los bits de la clave que son utilizados en cada una de ellas, lo cual conduce a una mejora en la seguridad del algoritmo IDEA.

El descifrado se realiza usando el mismo algoritmo que el cifrado, sólo que en este proceso las subclaves de descifrado, $k'_1, k'_2, \dots, k'_{52}$, se obtienen, en unos casos, como inversos multiplicativos (módulo $2^{16} + 1$) de otras tantas subclaves de cifrado, y en otros casos se obtienen como inversos aditivos (módulo 2^{16}) de otras determinadas subclaves de cifrado, conforme a un procedimiento fácil de describir que puede encontrarse en los tratados de criptografía (aquí no se recoge por los motivos anteriormente aducidos).

En cuanto a la **velocidad de ejecución**, el algoritmo IDEA es, en las implementaciones existentes, aproximadamente el doble de rápido que el DES, obteniéndose velocidades suficientes para su uso en la securización de aplicaciones telemáticas de elevado flujo de datos, como son las que soportan la videoconferencia y otros servicios multimedia.

La seguridad de IDEA es también muy elevada. El espacio de claves, 2^{128} , es superior incluso al triple* DES. En las condiciones tecnológicas actuales el ataque por fuerza bruta es impracticable. Si a eso añadimos que puede usarse bajo los cuatro modos de funcionamiento de los cifradores de bloque (ECB, CBC, etc.) y que también se presta al cifrado múltiple (triple IDEA, por ejemplo) se llega a la conclusión de que para su utilización en los intercambios de datos que se producen en el ámbito de la seguridad cívica, este algoritmo ofrece unos resultados más que satisfactorios.

Además, aunque para fines comerciales es necesario adquirir licencias, es de uso libre en entornos académicos y sin ánimo de lucro. Como se ve, todo son detalles a su favor.

* El cifrado múltiple se estudia en el epígrafe 4.7.

4.5. OTROS CIFRADORES DE BLOQUE

Existen multitud de propuestas de cifradores de bloque, es decir, aquellos en los que se opera de forma conjunta sobre el grupo de bits que constituyen cada uno de los bloques en que se ha dividido el mensaje en claro. En los apartados anteriores se han descrito los dos algoritmos más usados en las distintas soluciones de seguridad. A continuación se refieren, resumidamente, otros cuatro que, por diversas circunstancias, también ofrecen cierto interés.

RC2

El algoritmo RC2 fue diseñado por Ron Rivest para la firma RSA Data Security y pretendía, como tantos otros, desbancar al DES. Al igual que el algoritmo de cifrado de flujo RC4, del mismo autor, no ha sido publicado de forma detallada. No obstante, ha sido utilizado en algunos paquetes de correo electrónico seguro (S-MIME) y en otros productos de seguridad. Este criptosistema es propiedad de RSA Data Security y está protegido como secreto comercial.

Esta ausencia de publicidad hace que su interés quede muy reducido, porque los usuarios que lo utilicen han de fiarse de la robustez que RSA proclama que tiene, no pudiendo tener la opinión de criptoanalistas autorizados: se tienen que fiar de la empresa en el sentido de que no existen en él trampillas o puertas falsas. Por si fuera poca la desinformación, tanto de RC2 como de RC4 (algoritmo que se comentará más adelante) se han difundido detalles a través de Internet, de forma anónima, por alguien que dice conocer el código. Es decir, más confusión. Su autor, Ron Rivest, ha publicado* en Internet un documento que informa acerca de la estructura del algoritmo pero que no significa la especificación completa de éste.

De lo que se conoce de él, puede decirse que se trata de un algoritmo de cifrado en bloque de 64 bits (como DES e IDEA), que admite una clave de tamaño variable (desde 8 a 1.024 bits) y que su velocidad de cifrado es independiente del tamaño de la clave y es tres veces más rápido que DES. Debido a su característica de clave de longitud variable, el espacio de claves distintas asciende a la astronómica cifra de 2^{1024} , lo que lo hace muy resistente a los ataques por fuerza bruta.

Según parece, se apoya en tres algoritmos diferentes:

- Expansión de clave. A partir de una clave de entrada de longitud variable genera una clave expandida de 64 palabras de 16 bits. Lo que pretende conseguir es establecer una relación compleja entre cada bit de la clave expandida y los bits de la clave de entrada, lo que refuerza su seguridad.
- Algoritmo de cifrado. No usa nada parecido a las Cajas-S de DES, basando el oscurecimiento del mensaje en dos operaciones denominadas «mix» y «mash», algo así como «mezclar» y «triturar». Cada ciclo de una de estas operaciones consta a su vez de cuatro etapas diferenciadas que actúan sobre cada una de las cuatro palabras de 16 bits en que se divide el bloque de 64 bits de entrada. No es iterativo y en total se realizan 16 ciclos del tipo «mixing» entremezclado con otros dos del tipo «mashing».
- Algoritmo de descifrado. Como en otros algoritmos de esta clase, se compone de operaciones elementales que tratan de deshacer la mezcla conseguida con

* Rivest, R., *A description of the RC(r) Encryption Algorithm*, Internet Draft. Junio 1997.

las operaciones de cifrado. En este caso las operaciones se denominan «*r-mix*» y «*r-mash*». El proceso de descifrado cuenta, por tanto, con el mismo número de ciclos, y en el mismo orden, que el algoritmo de cifrado, sólo que aplicando operaciones inversas.

RC5

Fue desarrollado por el mismo autor del RC2 y ha sido también incorporado por RSA Data Security en muchos de sus productos de seguridad. En realidad se trata de una familia de algoritmos criptográficos caracterizados por una serie de parámetros que determinan el tamaño de la clave (variable), el número de ciclos y el tamaño de los bloques. Según los datos dados a conocer por la empresa RSA DS, el RC5 es muy resistente a todo tipo de ataques.

SKIPJACK

El interés de este algoritmo viene de la mano de haber sido el elegido para soportar el cifrado en el esquema del *Clipper chip* que se comentó en el Capítulo 2, y sobre el que se volverá a insistir más adelante cuando se aborde el problema de la gestión de claves. Como corresponde a todo lo que rodea a esa intentona de dejar la confidencialidad de las comunicaciones pendiente del *buen hacer* de determinadas agencias gubernamentales, este algoritmo fue desarrollado en el ámbito de la Agencia Nacional de Seguridad, NSA, de los Estados Unidos. Es un algoritmo *secreto* y su estructura no ha sido publicada. Además, sólo se pensaba permitir su implementación en determinados sistemas hardware que estuviesen contruidos a prueba de ataques contra la revelación de su contenido (*tamper proof*).

Es probable que sea robusto y es posible que no tenga puertas falsas por las que se puedan colar quienes la NSA decida, pero es también posible todo lo contrario porque no disponemos de la opinión de criptoanalistas independientes que lo hayan estudiado. Se permitió un acceso parcial a los detalles constructivos del Skipjack a algunos especialistas de confianza (de la NSA), por lo que existen informes acerca de su robustez y de su estructura*.

Según parece, se trata de un cifrador de bloque que trabaja con los cuatro modos presentes en estos criptosistemas y que en el modo CFB puede trabajar con distintos tamaños de c_i (incluso hasta de un bit), lo que puede convertirlo, de hecho, en un cifrador de flujo. (No se olvide que el deseo de sus progenitores era colocárnoslo en todos los equipos, teléfonos incluidos.)

Su estructura se cree que es similar a la de DES, con bloques de 64 bits, Cajas-S, una clave de 80 bits y un comportamiento iterativo basado en 32 ciclos.

GOST simétrico

Fue publicado como estándar en la extinta Unión Soviética, y pertenece a un grupo de algoritmos que abarca distintas funciones de seguridad, incluidas las de firma electró-

* E. F. Brickel, D. Denning et al. «SKIPJACK Review: Interim Report», en «Building in the Big Brother» [Hoff95], págs. 119 y ss.

nica. Se trata, por tanto, de un algoritmo público (como debe ser) que ha podido ser sometido a análisis por parte de los especialistas.

Su estructura es muy similar a la de DES. Al igual que éste, es iterativo y parte también de bloques de 64 bits que divide en dos mitades y que se intercambian entre sí en cada ciclo conforme a la expresión:

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1}, K_i) \end{aligned}$$

La función f es aquí el resultado de sumar, módulo 2^{32} , R_{i-1} con la subclave correspondiente al ciclo i , a continuación, pasarlo por unas Cajas-S donde se realizan operaciones de sustitución.

Una primera gran diferencia con DES es que la clave es aquí muchísimo mayor, 256 bits, lo que representa un incremento notable de su robustez. La otra diferencia es que consta de 32 ciclos y la subclave que actúa en cada uno de ellos es de 32 bits, de tal forma que el procedimiento de obtención de estas subclaves a partir de la clave original de 256 bits es más rápido y sencillo que el utilizado en DES.

4.6. UN ALGORITMO CRIPTOGRÁFICO PARA EL SIGLO XXI

Como ya se ha comentado, aunque el algoritmo DES resulta aún de utilidad para muchas aplicaciones, el hecho es que el avance que han experimentado los ordenadores y microprocesadores, tanto en velocidad como en potencia de cálculo, hace que sea cada vez menos difícil el ataque por fuerza bruta contra un algoritmo con tamaño de bloques de 64 bits y una clave de 56 bits. Más robustos son otros algoritmos de primera generación (caso de IDEA), pero resultan manifiestamente insuficientes para las previsiones existentes acerca de la capacidad de cálculo de los ordenadores en los próximos veinte o treinta años.

Numerosas han sido las iniciativas e intentonas de crear un algoritmo suficientemente robusto y fiable como para desbancar al DES de su posición de privilegio.

HACIA UN ESTÁNDAR DE CIFRADO AVANZADO (AES)

La iniciativa más solvente de cara a la sustitución de DES y otros algoritmos de primera generación es la llevada a cabo por el NIST, organismo de normalización de los Estados Unidos, con la convocatoria pública de propuestas para la adopción de un nuevo Estándar de Cifrado Avanzado (*Advanced Encryption Standard*, AES). En septiembre de 1997 realizó una convocatoria, a nivel mundial, para recibir propuestas para un nuevo algoritmo que sirviese para proteger las comunicaciones «no clasificadas» de sus distintos departamentos gubernamentales, pero que pudiera servir también para comunicaciones personales y comerciales.

En esta iniciativa se procuró solamente la adopción de un nuevo estándar para criptografía simétrica, no abordándose en ese momento la problemática de los algoritmos de clave pública que, aunque mucho más robustos que los de clave secreta, también requerirán de revisión de cara a afrontar las condiciones de computación que se esperan para las primeras décadas de este nuevo siglo.

Además, el algoritmo buscado debía responder a las características de los cifradores de bloque. Debido a que los modos de funcionamiento realimentado le otorgan a

los cifradores de bloque la posibilidad de comportarse también como cifradores de flujo cuando las circunstancias lo requieran, el algoritmo buscado, que debía ser mucho más rápido que los convencionales, podrá satisfacer los requisitos de ambos esquemas. En principio, se requería que los algoritmos dispusiesen de tres versiones en las que, tanto la clave como el tamaño de los bloques, fuese de 128, 192 y 256 bits. Este requisito se modificó posteriormente para que las distintas versiones mantuviesen, al menos, esos tres tamaños de clave pero con un solo tamaño de bloques de 128 bits.

Lejos del oscurantismo y de las trampas que rodeaban todo lo relacionado con el *Clipper Chip*, el algoritmo seleccionado debería ser público, disponible a nivel mundial, libre de derechos de autor, implementable tanto en hardware como en software, y no requerir para su adecuada implantación de ningún tipo de dispositivos o procedimientos patentados.

LA COMPUTACIÓN CUÁNTICA

Aunque en la actualidad no pasa de ser una formulación teórica con reducidas demostraciones prácticas, es posible que dentro de no muchos años se desarrolle un tipo de computadores basados en la dualidad de materia y onda presente en la física cuántica. Ello representará una drástica disminución en el volumen de los componentes y en el crecimiento de la velocidad de cálculo [Beth00].

La complejidad de los planteamientos de los principios de funcionamiento de los computadores cuánticos es tal que para su adecuada comprensión sería necesario adentrarse en conceptos y conocimientos que van más allá de los objetivos perseguidos en este libro. Baste aquí solamente señalar que en los ordenadores cuánticos se procesan los datos codificados en formas de onda que son tratadas apoyándose en determinadas propiedades físicas de esta forma de energía (se habla de *q-bits*, bits cuánticos), lo que permite diseñar algoritmos (criptografía cuántica) que puedan resolver en un tiempo razonable problemas numéricos actualmente inabordables.

Algunos problemas matemáticos de difícil solución que usando métodos actuales requieren un número de operaciones bit con un orden de magnitud* $O(n \cdot 2^n)$, en su versión cuántica el orden de magnitud se reduce a $O(n^2)$. En general, se espera que el tiempo que tardan esas nuevas máquinas en resolver los mismos problemas sea del orden de la raíz cuadrada del tiempo que tardan los ordenadores actuales.

Volviendo a las condiciones sobre tamaños impuestas por la iniciativa AES, cabe apuntar en primer lugar que al fijar en 128 el número de bits del bloque, la dificultad de construir un diccionario de todos los mensajes posibles en cada bloque para proceder luego a ensayar todas y cada una de las claves es del orden de 2^{128} , cuya raíz cuadrada es 2^{64} . Es decir, la dificultad de manejo de tal diccionario por un ordenador cuántico sería equivalente a la dificultad de llevar a cabo una operación similar usando un ordenador actual y un tamaño de bloques de 64 bits.

A ello hay que añadir el hecho de que las dimensiones del espacio de claves aumentan considerablemente, y que al existir varios tamaños posibles de claves, se puede adaptar la protección elegida a los requisitos de seguridad de cada aplicación concreta. Si pensamos en un ataque por fuerza bruta, el tiempo que tardaría un ordenador cuántico en realizar una búsqueda exhaustiva en un espacio de claves en el que ésta tenga 256 bits sería, en términos comparativos, proporcional a la raíz cuadrada de 2^{256} , es decir, 2^{128} , que será equivalente al necesario para realizar esa misma ope-

* Véase el apartado *Complejidad de los algoritmos* del epígrafe 3.3.

ración por un ordenador actual sobre un espacio de claves en el que ésta tenga 128 bits (el cual es, por ejemplo, el caso de IDEA).

Es decir, usando los mayores tamaños de clave disponibles en los criptosistemas conformes con AES, los ataques por fuerza bruta seguirán siendo impracticables incluso en un futuro, cuando esté operativo este tipo avanzado de máquinas de computación. Todo esto conduce a pensar que a un estándar de estas características puede augurársele larga vida, al menos veinte o treinta años de vigencia. (Los más optimistas hablan incluso de cincuenta o sesenta. Cabría decir, con D. Juan, «¡Cuán largo me lo fiáis!».)

RIJNDAEL: EL HEREDERO DE DES

Después de una primera ronda se seleccionaron quince de los algoritmos presentados a la convocatoria AES. En una segunda conferencia en 1999 se seleccionaron cinco de ellos en base a las numerosas pruebas realizadas por criptoanalistas y especialistas de todo el mundo. Estos algoritmos seleccionados fueron:

- MARS, de IBM.
- RC6, de RSA Laboratory.
- SERPENT, de R. Anderson, E. Biham y L. Knudsen.
- TWOFISH, de B. Schneier y otros.
- RIJNDAEL, de los belgas flamencos Joan Daemen y Vincent Rijmen.

Es menester alabar aquí el procedimiento llevado a cabo para la evaluación de los algoritmos candidatos: ha estado presidido por una gran apertura hacia toda la comunidad científica.

Al fin, en octubre de 2000, el NIST anunció la elección del Rijndael como algoritmo propuesto para el AES, con lo cual se convertirá en el nuevo estándar de Estados Unidos para las próximas décadas. Dada la apertura existente en su definición y especificación, es de esperar que sea adoptado también por otros países y otras organizaciones.

El algoritmo ofrece mayores posibilidades que las exigidas en la convocatoria final, ya que tanto la clave como el tamaño del bloque se pueden elegir entre 128, 192 y 256 bits, aunque las opciones de mayor tamaño de bloque no estarán incluidas en el estándar debido a que el estudio criptoanalítico realizado durante el proceso de selección se basó en bloques de 128 bits.

Los criterios bajo los que se desarrolló el algoritmo [DaRi99] fueron los siguientes:

- Resistencia ante todos los ataques conocidos.
- Velocidad y código compacto en su implementación sobre una amplia gama de plataformas.
- Simplicidad de diseño.

El algoritmo Rijndael no es excesivamente complejo ni en cuanto al tipo de operaciones básicas que realiza ni en lo que se refiere a las combinaciones que con ellas lleva a cabo. El esquema de funcionamiento del algoritmo se aparta de la denominada Estructura Feistel, que es la que posee DES, consistente en la partición del bloque que se cifra en dos mitades y la realización de operaciones con la clave sólo sobre una de ellas, de forma que se permutan a cada vuelta o ronda. En su lugar se define un *Estado*

intermedio que consiste en una determinada organización de los octetos que constituyen el bloque al que se está realizando el cifrado. A medida que se va realizando la transformación de los datos ese estado se va modificando y sirve de entrada a la siguiente etapa.

El estado puede ser considerado como una organización o *array* rectangular de octetos organizados en cuatro filas, de forma que el número total de columnas N_b depende del número de bits del bloque. Otro tanto cabe decir de la organización de los octetos de la clave: se agrupan en cuatro filas y N_k columnas, donde N_k depende del número de octetos de la clave. En la Figura 4.9 se representa esta organización de octetos para un bloque de 128 bits (16 octetos) y una clave de 192 bits (24 octetos).

Tanto los bloques como las claves usadas en Rijndael son consideradas como una ristra unidimensional de octetos. Dependiendo del tamaño, estos octetos se numeran (en decimal) dentro de los rangos 0...15 (si la longitud es de 16 octetos), 0...23 (si es de 24 octetos) o 0...31 (si se trata de piezas de 32 octetos). La correspondencia entre esta disposición unidimensional y la rectangular mostrada en la Figura 4.9 se lleva a cabo de modo que se trasladan los primeros cuatro octetos sobre la primera columna, los cuatro siguientes sobre la segunda y así sucesivamente. En un bloque de 128 bits (16 octetos) como el representado en la Figura 4.9 a) la correspondencia de esos octetos con la numeración decimal de su organización unidimensional sería:

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

El algoritmo se compone de varias rondas o ciclos cuyo número está especificado en una tabla y depende del tamaño del bloque y del tamaño de la clave. Así, en el caso de bloques de 128 bits, el número de rondas será 10, 12 o 14, dependiendo de si el tamaño de la clave es 128, 192 o 256 bits, respectivamente. Cada ronda está compuesta por cuatro operaciones basadas en transformaciones uniformes e invertibles que reciben el nombre de capas (*layers*). Lo de uniforme significa que todos los bits del estado son tratados de forma similar. El que sean invertibles significa que será factible

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

a) Estado con
 $N_b = 4$ (128 bits)

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$	$k_{0,4}$	$k_{0,5}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$	$k_{1,4}$	$k_{1,5}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$	$k_{2,4}$	$k_{2,5}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$	$k_{3,4}$	$k_{3,5}$

b) Estado con
 $N_k = 6$ (192 bits)

Figura 4.9. Ejemplo de *Estado* y clave de cifrado.

realizar la operación en sentido contrario durante la operación de descifrado. Estas capas han sido diseñadas para resistir ante los criptoanálisis lineal y diferencial y cada una de ellas tiene su propia funcionalidad:

- **Capa de mezcla lineal** (*linear mixing layer*): Garantiza un alto nivel de difusión a lo largo de las múltiples rondas.
- **Capa no lineal** (*non-linear layer*): Consiste en la aplicación de Cajas-S en paralelo que tienen propiedades óptimas de no linealidad.
- **Capa de adición de clave** (*key addition layer*): Se trata de una simple operación OR exclusivo entre el Estado intermedio y la clave propia de cada ronda.

EL CICLO BÁSICO DE RIJNDAEL

La ronda o ciclo básico cuenta con cuatro pasos o etapas:

a) Sustitución de byte (*ByteSub*)

En esta operación, al igual que en las restantes, este algoritmo trabaja a nivel de octetos, en lugar de operar a nivel de bits como los algoritmos tradicionales. La transformación *ByteSub* es una sustitución no lineal que se aplica a todos y cada uno de los octetos que configuran el Estado intermedio. Se construye en base a dos transformaciones:

1. Cada octeto es considerado como un elemento del conjunto $GF(2^8)$ que estudiamos en el apartado *Campos de Galois del tipo $GF(q^n)$* del epígrafe 3.2. Cada octeto es sustituido por su inverso multiplicativo. Como el elemento cero carece de inverso, el octeto de valor 00 se sustituye por sí mismo.
2. Al resultado de la sustitución anterior se le aplica la transformación definida por la siguiente operación matricial:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

donde $y_0, y_1, \dots, y_7$ representa el octeto resultante de la operación y $x_0, x_1, \dots, x_7$ representa la entrada.

Como en realidad se trata de operaciones sobre valores hexadecimales, no es necesario realizar estas operaciones «en vivo y en directo» en el momento de realizar la transformación sobre un octeto concreto del Estado intermedio, sino que es posible tener construida una tabla bidimensional (Figuras 4.10 y 4.11) donde se reflejen los resultados de esta transformación para todos los octetos comprendidos entre 00_{16} y FF_{16} . Esta tabla tiene 16 columnas y 16 filas que determinan la sustitución corres-

		y															
x		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	0c	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	0a	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	80	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	0e	55	28	df
	f	8c	a1	89	0d	bf	E6	42	68	41	99	2d	0f	b0	54	bb	16

Figura 4.10. Tabla que determina la operación *ByteSub*.

pondiente a cada octeto genérico xy_{16} . Puede observarse que lo que se suma al resultado de la multiplicación matricial recogida en el punto 2 de la transformación *ByteSub* es el valor hexadecimal 63_{16} . Por ello, en la tabla mencionada el valor correspondiente a 00_{16} (valor que resulta inalterado de las dos anteriores operaciones) es, precisamente, 63_{16} . Por comparación con DES^1 , a esta tabla se la denomina como Caja-S.

La operación inversa a *ByteSub* (necesaria en el proceso de descifrado) puede llevarse a cabo aplicando la sustitución que se refleja en otra tabla, que se representa en la Figura 4.11, inversa de la anterior, construida de forma sencilla a partir de ella sin más que tomar como entrada los valores que antes resultaban como salida y viceversa. O sea, en esta tabla al valor 63_{16} le corresponde el 00_{16} .

		y															
x		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	de	a1	66	28	d9	24	b2	76	5b	a4	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	4d	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	3c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	0e	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Figura 4.11. Tabla que determina la operación inversa de *ByteSub*.

b) Desplazamiento de filas (ShiftRow)

En esta transformación lo que se hace es girar cíclicamente las filas del Estado intermedio varios lugares a la izquierda dependiendo de la posición de la fila y del tamaño del bloque de cifrado. Cuando se trata de bloques de 128 o 196 bits, la primera fila (fila 0) no se desplaza, la fila 1 se desplaza cíclicamente un lugar, la fila 2 dos lugares y la fila 3 tres lugares. Cuando los bloques son de 256 bits estos desplazamientos son ligeramente diferentes: cero, uno, tres y cuatro lugares, respectivamente.

En la Figura 4.12 se representan los efectos de esta transformación en el caso de tratarse de un bloque de 128 bits. A la izquierda se muestran los 16 octetos del Estado intermedio procedentes de la anterior transformación, numerados (en decimal) del cero al quince, y a la derecha se representa la nueva posición que ocuparán esos octetos dentro de la matriz una vez se haya realizado la transformación.

c) Mezcla de columnas (MixColumn)

En esta transformación, los cuatro octetos que constituyen cada una de las columnas del estado intermedio procedente de la etapa anterior son consideradas como polinomios sobre $GF(2^8)$. En realidad se trata de una operación matemática que va un paso más allá de las que se describieron en el epígrafe 3.2. En este caso, se trata de un polinomio de grado inferior a cuatro cuyos coeficientes son, a su vez, polinomios en $GF(2^8)$. Es decir, los cuatro coeficientes son los cuatro octetos de la columna.

Si partimos de dos de estos polinomios:

$$p(x) = p_3 x^3 + p_2 x^2 + p_1 x + p_0$$

$$q(x) = q_3 x^3 + q_2 x^2 + q_1 x + q_0$$

la suma de estos dos vectores sobre $GF(2^8)$ se realiza mediante una simple operación OR exclusivo.

La multiplicación resulta algo más compleja. Si multiplicamos estos dos polinomios el resultado será un polinomio de grado seis que se obtendrá multiplicando miembro a miembro los polinomios de partida. Esto es:

$$r(x) = r_6 x^6 + r_5 x^5 + r_4 x^4 + r_3 x^3 + r_2 x^2 + r_1 x + r_0$$

Donde:

$$r_0 = p_0 \cdot q_0$$

$$r_1 = p_1 \cdot q_0 \oplus p_0 \cdot q_1$$

$$r_2 = p_2 \cdot q_0 \oplus p_1 \cdot q_1 \oplus p_0 \cdot q_2$$

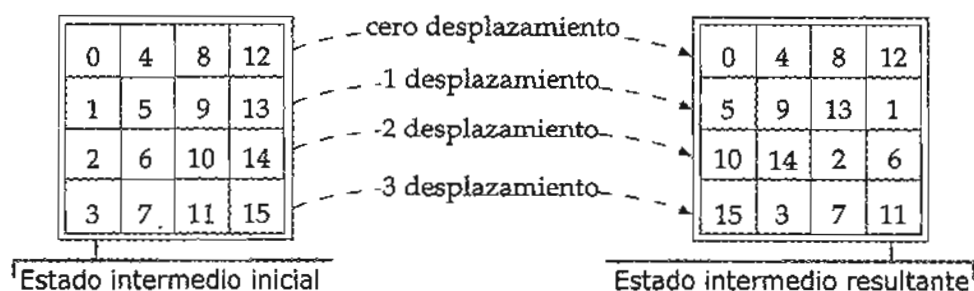


Figura 4.12. Efecto de la transformación ShiftRow.

$$\begin{aligned}
r_3 &= p_3 \cdot q_0 \oplus p_2 \cdot q_1 \oplus p_1 \cdot q_2 \oplus p_0 \cdot q_3 \\
r_4 &= p_3 \cdot q_1 \oplus p_2 \cdot q_2 \oplus p_1 \cdot q_3 \\
r_5 &= p_3 \cdot q_2 \oplus p_2 \cdot q_3 \\
r_6 &= p_3 \cdot q_3
\end{aligned}$$

Como $r(x)$ tiene que ser representable mediante un vector de cuatro octetos, se realiza una reducción módulo un polinomio de grado cuatro. El polinomio elegido para llevar a cabo esta reducción es $M(x) = x^4 + 1$. Como $x^j \bmod (x^4 + 1) = x^{j \bmod 4}$, el producto modular de $p(x)$ y $q(x)$, que denotaremos como $p(x) \otimes q(x)$, será:

$$s(x) = p(x) \otimes q(x) = s_3 x^3 + s_2 x^2 + s_1 x + s_0 \quad \text{donde:}$$

$$\begin{aligned}
s_0 &= p_0 \cdot q_0 \oplus p_3 \cdot q_1 \oplus p_2 \cdot q_2 \oplus p_1 \cdot q_3 \\
s_1 &= p_1 \cdot q_0 \oplus p_0 \cdot q_1 \oplus p_3 \cdot q_2 \oplus p_2 \cdot q_3 \\
s_2 &= p_2 \cdot q_0 \oplus p_1 \cdot q_1 \oplus p_0 \cdot q_2 \oplus p_3 \cdot q_3 \\
s_3 &= p_3 \cdot q_0 \oplus p_2 \cdot q_1 \oplus p_1 \cdot q_2 \oplus p_0 \cdot q_3
\end{aligned}$$

que puede ser expresado como un producto matricial de la forma:

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} p_0 & p_3 & p_2 & p_1 \\ p_1 & p_0 & p_3 & p_2 \\ p_2 & p_1 & p_0 & p_3 \\ p_3 & p_2 & p_1 & p_0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}$$

Es menester hacer notar que aunque se trate de operaciones con polinomios de grado inferior a cuatro no se está trabajando sobre $\text{GF}(2^4)$, sino sobre $\text{GF}(2^8)$. Además, el polinomio $M(x) = x^4 + 1$ no es un polinomio irreducible sobre $\text{GF}(2^8)$, por lo que la operación de multiplicación de polinomios aquí descrita no es necesariamente invertible.

Concretando, la operación **MixColumn** de Rijndael consiste en considerar los cuatro octetos que constituyen cada una de las columnas del Estado intermedio procedente de la etapa anterior como polinomios sobre $\text{GF}(2^8)$ y multiplicarlos por un polinomio fijo $c(x)$ módulo $x^4 + 1$. Este polinomio predeterminado es:

$$c(x) = 03_{16} x^3 + 01_{16} x^2 + 01_{16} x + 02_{16}$$

Si llamamos $a(x)$ al polinomio determinado por una columna del Estado de entrada y $b(x)$ al correspondiente con la columna del Estado de salida resultante de haberle aplicado esta transformación, tendremos:

$$b(x) = c(x) \otimes a(x)$$

que expresado en forma matricial conforme a lo deducido más arriba para los polinomios genéricos $s(x)$, $p(x)$ y $q(x)$, resulta:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

Para la operación de descifrado habrá que aplicar la transformación inversa. Aunque, como decíamos más arriba, por no ser $M(x) = x^4 + 1$ un polinomio irreducible sobre $GF(2^8)$, este tipo de operaciones no son necesariamente invertibles, en el caso de Rijndael esta situación se resuelve gracias a que el polinomio fijo $c(x)$ elegido sí tiene un inverso, al que denominaremos $d(x)$. Es decir:

$$c(x) \otimes d(x) = 1$$

Este polinomio resulta ser:

$$d(x) = 0B_{16} x^3 + 0D_{16} x^2 + 09_{16} x + 0E_{16}$$

Por todo ello, la operación de descifrado se realiza mediante una operación matricial en la que los coeficientes de la matriz se obtienen a partir de los coeficientes de $d(x)$ de forma similar a lo descrito para la operación de cifrado.

d) Adición de la clave de ciclo (*AddRoundKey*)

Este cuarto y último paso consiste simplemente en realizar una operación OR exclusivo entre el Estado intermedio procedente de la transformación anterior y la subclave correspondiente al ciclo o ronda de que se trate. El Estado resultante de esta transformación será la entrada del siguiente ciclo o, si se trata de la última ronda, constituirá el dato de salida del algoritmo.

En resumen, las cuatro etapas que constituyen el ciclo básico de Rijndael son:

ByteSub, ShiftRow, MixColumn y AddRoundKey

ESQUEMA DE CIFRADO Y DESCIFRADO

Para cifrar un bloque, el algoritmo realiza en primer lugar (véase Figura 4.13) un paso *AddRoundKey* de adición con la primera subclave.

A continuación se llevan a cabo varias rondas o ciclos como el antes descrito, cuyo número, n , es:

- 9 si tanto el bloque como la clave son 128 bits.
- 11 si la clave o el bloque son 192 bits y ninguno de ellos supera esta longitud.
- 13 si la clave o el bloque son de 256 bits.

Como colofón, realiza un ciclo extra compuesto por las transformaciones *ByteSub*, *ShiftRow* y *AddRoundKey*, es decir, no aparece en él el paso *MixColumn* de mezcla de columnas.

En el **proceso de descifrado** se procederá justo en orden inverso, realizando también en cada paso la operación inversa. La transformación *AddRoundKey*, por estar constituida por una simple operación XOR, es inversa de sí misma, pero en el caso de las otras tres será necesario utilizar la transformación inversa, según se ha ido comentando cuando se ha descrito su funcionamiento.

Para obtener **las subclaves** a aplicar en cada ciclo se procede a generar una serie de expansiones de la clave original, cada una del mismo tamaño que ésta. Las expansiones se realizan mediante un procedimiento descrito en pseudo C en la especificación del algoritmo, que consiste en añadir palabras de cuatro octetos obtenidas mediante

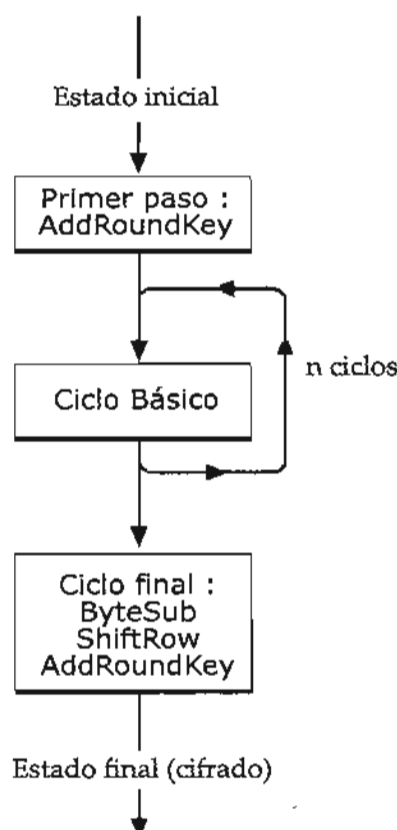


Figura 4.13. Proceso de cifrado de Rijndael.

operaciones XOR entre otras partes prefijadas de la expansión anterior. Para obtener la primera palabra de cuatro octetos e iniciar la expansión, se hace uso de la transformación mediante Cajas-S que opera en el paso *ByteSub*, y de una operación XOR con una constante específica predeterminada en función del ordinal del ciclo.

La generación de subclaves para el proceso de descifrado se lleva a cabo de forma idéntica a la seguida en el proceso de cifrado, sólo que en este caso el orden de utilización de las subclaves será también el inverso, es decir, se comenzará por la última y se finalizará aplicando la primera.

LA UTILIZACIÓN DE RIJNDAEL

Siguiendo las especificaciones de Rijndael, la implementación de un criptosistema que opere conforme a este algoritmo no resulta excesivamente compleja. En los párrafos precedentes hemos recogido la descripción de las operaciones que lo constituyen tratando de aproximarnos a los fundamentos matemáticos en que se sustentan. Pero esta complejidad no se traslada a la especificación escueta de su comportamiento; antes bien, para desarrollar un programa conforme con este algoritmo solamente será necesario saber utilizar las tablas prefijadas en que se basa la transformación *ByteSub*, la matriz prefijada que determina la operación *MixColumn* y las demás operaciones lógicas y de desplazamiento en que se basan las restantes. Por todo ello, además de las múltiples implementaciones existentes, resulta factible desarrollar una propia bastante rápida y adaptada al entorno de seguridad de que se trate.

Debido a las circunstancias que rodean su adopción, a la información que han proporcionado sus inventores acerca de los criterios de seguridad elegidos (lo que

facilita la labor de los criptoanalistas y evaluadores) y a la amplísima participación de la comunidad científica en el proceso de selección, debido a todo esto, decimos, es razonable tener en este algoritmo un alto nivel de confianza en cuanto a su seguridad y eficacia.

Por el hecho de convertirse en un estándar oficial dentro de Estados Unidos, cabe esperar que de forma creciente sea adoptado por todo tipo de asociaciones dentro del mundo comercial y de la seguridad cívica. El que esté libre de licencia conlleva que otros Estados y organizaciones puedan, fácilmente y sin problemas, adoptarlo como estándar, y que durante los primeros años de su incorporación vaya acaparando importantísimas parcelas dentro de los mecanismos de seguridad que requieren criptografía de clave secreta (la cuestión de la criptografía de clave pública no se ve afectada por las bondades y resistencias de este algoritmo).

Conforme a lo que ya se comentó en relación con la evolución que se prevé en las arquitecturas de los ordenadores, el tamaño de las claves con las que puede operar Rijndael hace que sea de esperar que este algoritmo aguante durante bastante tiempo los intentos de ruptura y vulneración que se apoyen en las mejoras que se obtengan en los procesos de computación.

Otra cosa más compleja de evaluar es en qué medida y con qué cadencia temporal las aplicaciones que están implementadas usando algoritmos convencionales vayan a ser reconfiguradas para incluir este nuevo estándar. Ya se indicó, al comentar la debilidad y vigencia del DES, que la eficacia y oportunidad de usar un determinado algoritmo no está sólo, ni principalmente, determinada por la robustez intrínseca de éste, sino por todas las medidas y circunstancias que rodean a la aplicación telemática segura que se utilice y a la política de seguridad que se diseñe para un determinado dominio.

Además, en el mundo comercial hay otros intereses y prioridades que pueden determinar que algunos algoritmos que son propiedad privada de ciertas empresas puedan continuar siendo usados durante todavía bastante tiempo.

4.7. CIFRADO MÚLTIPLE

Cuando se analiza el grado de confianza que proporciona el uso de criptografía simétrica como soporte de los protocolos de seguridad para usos cívicos, lo primero que es necesario resaltar es que hay que tener en cuenta que el tipo de información que intercambian los usuarios está inscrito en las operaciones convencionales que necesitan realizar (nada de espías, ni de mafia, ni de tesoros escondidos) y que el grado de robustez de los algoritmos empleados, junto con la aplicación de una adecuada *política de seguridad* (que pondrá más defensas cuando el documento que circula tenga más relevancia), va a ser suficientemente grande como para confiar en que al atacante le va a costar más el trabajo que tenga que realizar para romper la seguridad que el beneficio que va a obtener con dicha violación.

No obstante, puede pensarse que cuando la velocidad y potencia de los ordenadores aumente, también mejorará la facilidad con la que los criptosistemas pueden ser atacados. Lo que ocurre es que, en ese proceso evolutivo, siempre podremos cambiar de algoritmo (pasándonos a otro más evolucionado) o, simplemente, aumentar el tamaño de la clave. Es decir, en este juego del ratón y el gato, siempre es el ratón el que tiene más posibilidades de ganar.

En los algoritmos de clave pública este aumento del tamaño de la clave puede realizarse sin modificar la estructura del algoritmo, según se verá en el próximo

capítulo. En los criptosistemas simétricos, en cambio, un aumento en el tamaño de la clave representará un rediseño del algoritmo, con toda la complicación que ello supone. (Algunos algoritmos avanzados, como es el caso del Rijndael, son más flexibles en cuanto a la variabilidad del tamaño de la clave.)

Una alternativa interesante para aumentar la robustez del cifrado sin alterar la estructura de los algoritmos la constituye el *cifrado múltiple*, que consiste en realizar más de una vez el proceso de cifrado poniendo en juego más de una clave.

En el **doblo cifrado**, primeramente se aplica al mensaje en claro el algoritmo de transformación usando una clave k_1 :

$$c_1 = E_{k_1}(m)$$

A continuación se cifra de nuevo el c_1 obtenido en el proceso anterior, utilizando una nueva clave k_2 , resultando el criptograma final:

$$\begin{aligned} c &= E_{k_2}(c_1) \\ c &= E_{k_2}[E_{k_1}(m)] \end{aligned}$$

El proceso de descifrado es inverso del anterior. En primer lugar es necesario recuperar c_1 utilizando la clave k_2 y, a continuación, descifrar con k_1 :

$$\begin{aligned} c_1 &= D_{k_2}(E_{k_2}[E_{k_1}(m)]) = E_{k_1}(m) \\ D_{k_1}(c_1) &= D_{k_1}[E_{k_1}(m)] = m \end{aligned}$$

Es decir: $m = D_{k_1}[D_{k_2}(c)]$

Puede observarse que en la nomenclatura utilizada para la formulación de las expresiones anteriores se han usado indistintamente paréntesis y corchetes para delimitar el elemento al que se aplica la operación de cifrado o descifrado. El único objetivo de su uso combinado es facilitar al lector la interpretación de dichas fórmulas.

El triple cifrado se caracteriza, como su nombre indica, por aplicar tres veces consecutivas el proceso de cifrado. Para ello, pueden utilizarse tres claves distintas (como es el caso representado en la Figura 4.14) de forma que la transformación del mensaje en claro pasa por las siguientes fases:

$$\begin{aligned} c_1 &= E_{k_1}(m) \\ c_2 &= D_{k_2}(c_1) = D_{k_2}[E_{k_1}(m)] \\ c &= E_{k_3}(c_2) = E_{k_3}(D_{k_2}[E_{k_1}(m)]) \end{aligned}$$

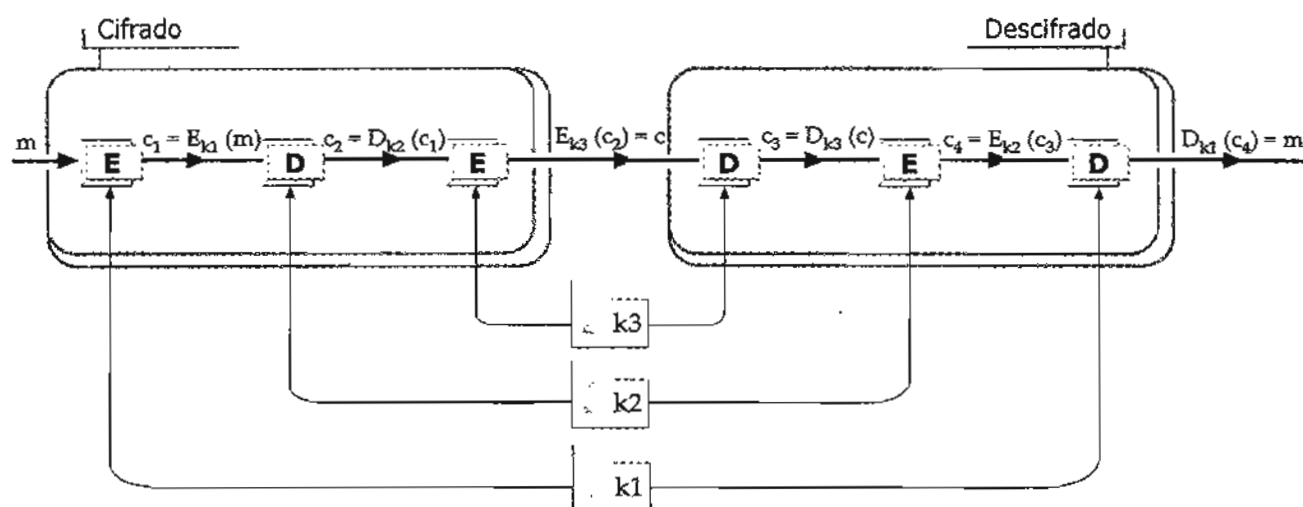


Figura 4.14. Triple cifrado.

Simétricamente, el proceso de descifrado sería:

$$\begin{aligned} c_3 &= D_{k_3}(c) = D_{k_3}[E_{k_3}(D_{k_2}[E_{k_1}(m)])] = D_{k_2}[E_{k_1}(m)] = c_2 \\ c_4 &= E_{k_2}(c_3) = E_{k_2}(c_2) = E_{k_2}(D_{k_2}[E_{k_1}(m)]) = E_{k_1}(m) = c_1 \\ D_{k_1}(c_4) &= D_{k_1}(c_1) = D_{k_1}[E_{k_1}(m)] = m \end{aligned}$$

Es decir: $m = D_{k_1}(E_{k_2}[D_{k_3}(c)])$

Si en lugar de usar tres claves distintas se usasen solamente dos de forma alternada, el resultado también reforzaría mucho la seguridad y tendría la ventaja de que la gestión de claves sería más simple. En este caso, las operaciones de cifrado y descifrado serían:

$$\begin{aligned} c &= E_{k_1}(D_{k_2}[E_{k_1}(m)]) \\ m &= D_{k_1}(E_{k_2}[D_{k_1}(c)]) \end{aligned}$$

La gracia del cifrado múltiple está en que si los procesos de criptoanálisis mejoran debido a cualquier causa, es posible aumentar la seguridad del cifrado realizando tantas «pasadas» como se quiera, sin modificar la estructura interna del cifrador.

4.8. CIFRADORES DE FLUJO

La característica principal del cifrado en flujo consiste en considerar el mensaje en claro como un flujo continuo de bits (o de caracteres) y generar a la salida el correspondiente flujo de bits resultante de la transformación producida en el proceso de cifrado.

En este sentido, los cifradores en bloque, trabajando en uno de los modos realimentados antes expuestos, pueden comportarse de la forma descrita en el párrafo anterior y ser considerados, por tanto, como cifradores de flujo. Esta circunstancia hace que los desarrollos y avances que se llevan a cabo dentro del campo de los cifradores en bloque (como el ya comentado AES) puedan repercutir como mejoras en los cifradores de flujo cuando las circunstancias así lo requieran.

No obstante, cuando se habla de los **cifradores de flujo**, sin más, se está haciendo referencia a un procedimiento específico de cifrado cuyo esquema, que fue ya intro-

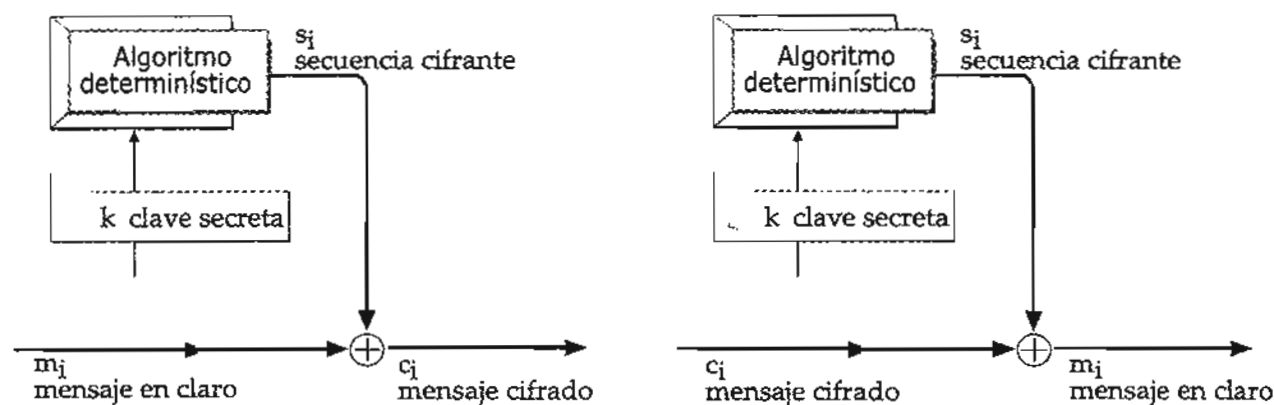


Figura 4.15. Cifrado y descifrado en flujo.

ducido al comentar la Figura 4.2, se representa de nuevo en la Figura 4.15, en su doble comportamiento de cifrado y descifrado.

La principal ventaja de este sistema es la simpleza extrema del proceso de cifrado: una sencilla suma módulo 2 (operación XOR) que puede conseguirse mediante un elemental dispositivo electrónico o una simple operación computacional. Se apoya en la propiedad que tiene la operación XOR, consistente en que realizando dos veces la operación con el mismo operador, se recupera el valor original. Así:

$$\begin{aligned}a \oplus b \oplus b &= a \\ a \oplus a &= 0\end{aligned}$$

Si se realiza la operación XOR entre el mensaje en claro y la secuencia cifrante que depende de la clave secreta k , se obtiene el mensaje cifrado:

$$c = m \oplus s$$

Si en el receptor se usa *esa misma* secuencia cifrante para realizar la operación XOR con el mensaje cifrado, c , se obtiene como resultado el mensaje en claro.

$$m = c \oplus s$$

La sencillez de las operaciones de cifrado y descifrado hace que con este tipo de sistema puedan conseguirse velocidades muy elevadas.

CIFRADOR DE VERNAM

Los cifradores de flujo se basan en un cifrador primitivo propuesto por el ingeniero Vernam en 1917 para ser utilizado apoyándose en los códigos usados por los teletipos. Estos dispositivos, que supusieron un avance notable sobre los sistemas telegráficos clásicos basados en el código Morse, representaban los caracteres y los signos a transmitir mediante un código en el que cada carácter está representado por un número fijo de bits. Cada uno de esos elementos (bits) puede presentar dos valores binarios (uno y cero) que se materializan mediante la aparición de señal o por la ausencia de ella (espacio), o bien mediante perforaciones sobre una cinta de papel.

El código Baudot, que fue el utilizado por Vernam (y que ha estado siendo utilizado hasta nuestros días en las instalaciones telegráficas y de teletipo en Europa), utiliza cinco bits para representar la tabla de símbolos, que divide en dos bloques de 32 caracteres cada uno, de tal forma que un carácter especial sirve para permutar el bloque. (El código ASCII, equivalente al Baudot y usado principalmente en América, utiliza ocho bits y un solo bloque. Como fue incorporado a las técnicas de representación usadas en los ordenadores, es bien conocido en el ámbito de los desarrolladores de aplicaciones informáticas.)

Volviendo al grano, lo que Vernam propuso fue generar una secuencia cifrante aleatoria de caracteres, constituidos por cinco bits, que eran sumados módulo 2 con los correspondientes caracteres del mensaje en claro. La secuencia cifrante era aleatoria y de un solo uso (*one time pad*), gracias a lo cual el cifrador obtenido era de **secreto perfecto** (por las razones que se discutieron en el epígrafe 3.1).

LA SECUENCIA CIFRANTE

La robustez absoluta del cifrador de Vernam se basa en la total aleatoriedad de la secuencia cifrante y en el hecho de que el tamaño de ésta (que actúa como clave) sea tan grande como el del mensaje a cifrar. Además, al no repetirse, los criptoanalistas no pueden sacar partido de las redundancias del lenguaje. El problema de este tipo de cifradores es que la distribución de la clave de cifrado (la secuencia cifrante) es posible en organizaciones cerradas de alto secreto, como son las agencias de espionaje y afines, pero resultan totalmente inviables para las aplicaciones de interés en la seguridad cívica.

Por ello, lo que se procura en los cifradores de flujo es diseñar unos algoritmos determinísticos que, a partir de una clave secreta compartida tanto por el emisor como por el receptor del mensaje, generen una secuencia cifrante con unas características lo más parecidas posible a una secuencia aleatoria del tipo de las utilizadas en el cifrador de Vernam. A este tipo de secuencias se las denomina pseudoaleatorias. Naturalmente, cuanto mayor sea el tamaño de la clave secreta (suele valer de 64 a 250 bits, según los criptosistemas) mayor será la dificultad que tengan los criptoanalistas para, conocido el criptograma, descubrir el mensaje original. Algunas de las características de la secuencia cifrante, véase [FGHM00], son:

- a) *Periodo*. Es muy importante que la secuencia no se repita dentro del proceso de cifrado de un mensaje. Con los cifradores vigentes se pueden conseguir periodos de tamaño suficientemente grandes como para que este requisito se cumpla sobradamente.
- b) *Distribución de unos y ceros*. La secuencia cifrante debe presentar una distribución lo más uniforme posible. Para que sea considerada como pseudoaleatoria debe cumplir tres postulados formulados por Golomb. De forma muy resumida, cabe decir que el primero impone que en cada periodo de la secuencia el número de ceros y de unos debe ser aproximadamente igual; el segundo de ellos acota la aparición de *rachas* (sucesión de dígitos iguales, unos o ceros) y el tamaño de éstas. El tercer postulado es de más difícil explicación y especifica los valores que debe tener lo que se denomina *autocorrelación*, parámetro que determina las coincidencias entre una secuencia y ella misma desplazada cíclicamente un determinado número de posiciones.
- c) *Imprevisibilidad*. Se exige que la secuencia cifrante sea imprevisible, queriendo esto significar que dada una secuencia cifrante de cualquier longitud, predecir si el siguiente dígito es uno o cero no puede representar una probabilidad superior al 50 por 100.
- d) *Facilidad de implementación*. La secuencia cifrante debe poderse generar de manera rápida y sencilla, poniendo en juego recursos asequibles.

GENERADORES DE SECUENCIAS CIFRANTES

Teniendo en cuenta lo dicho hasta aquí, no resulta difícil de deducir que el plato fuerte de los cifradores de flujo consiste en el estudio de los mecanismos de generación de las secuencias que se usarán para el cifrado de los datos. Una descripción mínimamente rigurosa de la robustez y debilidad de cada una de las alternativas propuestas escapa de los objetivos aquí marcados. En los párrafos que siguen se hará una somera reseña del abanico de soluciones propuestas para este fin, prestando atención sólo a las más

robustas e indicando, sin acudir a demostraciones, la mayor o menor seguridad de cada una de ellas.

Registros de desplazamiento realimentados

Un mecanismo utilizado, en sus diversas variantes, por los algoritmos de cifrado en flujo más significativos es el representado en la Figura 4.16. El registro está constituido por n bits o etapas binarias que se desplazan un lugar a la derecha a cada impulso de reloj. El nuevo valor que ocupa la etapa 1 está determinado por una función g que depende de los valores de las n etapas del registro. El valor inicial del registro, la *semilla*, constituye la clave bajo la que opera el cifrador. El periodo de la secuencia cifrante depende del valor de n , tamaño de la clave, y de las características de la función g . Cuanto mayor sea n más elevado será el periodo de la secuencia cifrante generada, cuyo máximo teórico será 2^n .

Dependiendo de si la función g es lineal o no lineal, tendremos dos tipos de estos registros: LFSR y NLFSR.

Registros de desplazamiento realimentados no linealmente (NLFSR)

En los registros NLFSR (*non linear feedback shift registers*), como su nombre indica, la función g de realimentación no depende linealmente de las distintas etapas. Estas funciones de realimentación pueden conseguirse mediante una combinación adecuada de puertas lógicas. Este tipo de mecanismos tiene poco interés para ser usado en cifradores de flujo porque pueden generar secuencias con ciclos cortos que los harían muy vulnerables.

Registros de desplazamiento realimentados linealmente (LFSR)

Son los realmente utilizados debido a sus mejores comportamientos, aunque para obtener una buena respuesta es necesario utilizar combinadamente un conjunto de ellos.

En la Figura 4.17 se muestra una representación genérica de este tipo de registros de desplazamiento. En este caso la función g de realimentación está definida por una sucesión de operaciones XOR en la que intervienen todos los bits del registro para los que c_i vale uno. Puede definirse un polinomio asociado a la combinación de bits «conectados» a la operación XOR de la forma:

$$f(x) = 1 + c_1 x + c_2 x^2 + \dots c_i x^i + \dots c_n x^n$$

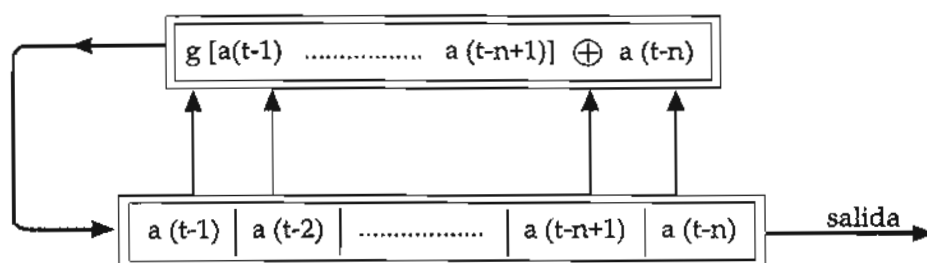


Figura 4.16. Registro de desplazamiento realimentado.

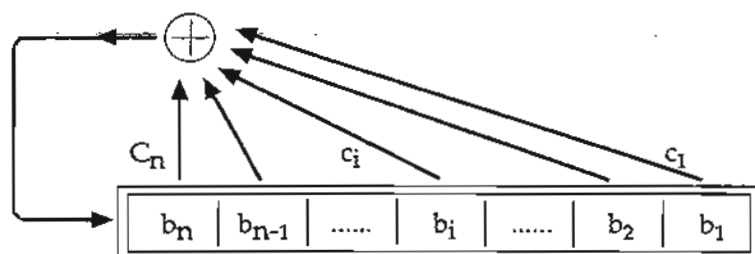


Figura 4.17. LFSR de n bits.

Por ejemplo, el polinomio asociado al registro realimentado representado en la Figura 4.18 es:

$$f(x) = 1 + x + x^4$$

Si el valor inicial fuese 1110, los dieciséis primeros estados del registro serían:

1110, 1111, 0111, 1011, 0101, 1010, 1101, 0110
0011, 1001, 0100, 0010, 0001, 1000, 1100, 1110

Sólo los quince primeros son valores diferentes, ya que el decimosexto coincide con el primero, lo que conlleva que a partir de él se repite de nuevo la secuencia. Por tanto, la secuencia de salida del registro sería, para este caso particular, la repetición sucesiva de:

0 1 1 1 1 0 1 0 1 1 0 0 1 0 0.....

A partir de la modelación matemática que representa el definir la función de realimentación $F(x)$ mediante un polinomio de grado n , la discusión de los distintos tipos de realimentación que pueden darse en los LFSR puede plantearse mediante el estudio de los distintos polinomios que pueden derivarse de la particularización de esta expresión, de forma que se fije el grado del polinomio (número de bits o etapas del registro) y el valor (uno o cero) que toman los distintos coeficientes c_i .

El estudio matemático que ello conlleva es de cierta envergadura, por lo que aquí nos limitaremos a resaltar las conclusiones que más repercusión tienen en las propuestas prácticas de criptosistemas (en [FGHJ00] y [Schn96] puede encontrarse una discusión sobre los distintos casos que se presentan, además de algunas referencias apropiadas). De todos los polinomios, los que resultan de mayor interés son los *polinomios irreducibles* (a los que ya hicimos referencia al estudiar los campos de Galois) y, dentro de este tipo de polinomios, los que ofrecen mejores resultados para ser usados en los cifradores de flujo son los de un tipo especial denominados *polinomios primitivos*. El polinomio $f(x) = 1 + x + x^4$ definido en la configuración mostrada en la Figura 4.18 es de este tipo.

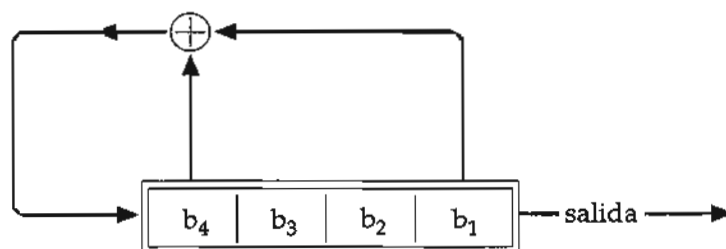


Figura 4.18. LFSR de cuatro bits.

Los generadores LFSR con polinomio primitivo cumplen bastante bien los postulados de Golomb y son los que consiguen un mayor valor para el periodo de repetición de la secuencia cifrante: $2^n - 1$, siendo n el número de bits o etapas del registro (quince en el caso representado en la Figura 4.18).

CIFRADORES DE FLUJO COMO COMBINACIÓN DE VARIOS LFSR

A pesar de las bondades de los LFSR para generación de secuencias cifrantes, el solo uso de uno de estos registros para configurar un cifrador de flujo no será suficiente para obtener criptosistemas robustos, por lo que se recurre a usar varios de estos registros operando de forma interactiva y coordinada.

Existen estudios pormenorizados de estas distintas configuraciones y de la fortaleza que poseen los algoritmos resultantes. Aquí sólo recogeremos dos de ellas. La primera es el generador de Geffe (Figura 4.19), que se ha elegido por su sencillez y porque sirve, en cierto modo, para ilustrar esta familia de cifradores. El segundo ejemplo seleccionado es el del cifrador A5 (Figura 4.20), cuyo interés radica en que es el algoritmo usado en el estándar GSM de comunicaciones móviles.

Generador de Geffe

Consiste en una combinación no lineal de tres LFSRs (Figura 4.19) en la que uno de ellos, el LFSR-1, gobierna un multiplexor de tal manera que cuando $a_1(t)$ sea 1 la salida $b(t)$ tomará el valor correspondiente al bit aportado por el desplazamiento del registro LFSR-3, mientras que cuando el valor dinámico de $a_1(t)$ sea 0, será el LFSR-2 el que proporcione el bit de salida para la generación de la secuencia cifrante.

Se puede demostrar que el periodo de la secuencia de salida es el mínimo común múltiplo del periodo de los tres generadores. Si estos números son primos entre sí, el periodo final resultante sería, por tanto, el producto de todos ellos, lo que implica que se pueden conseguir cifras muy elevadas, de forma que es posible cifrar mensajes en claro de gran tamaño sin que se produzca la repetición de la secuencia cifrante. No obstante, este tipo de generadores, de apariencia tan robusta, han sido criptoanalizados con éxito, por lo que resulta necesario recurrir a otras configuraciones más elaboradas a la hora de interconectar varios LFSRs.

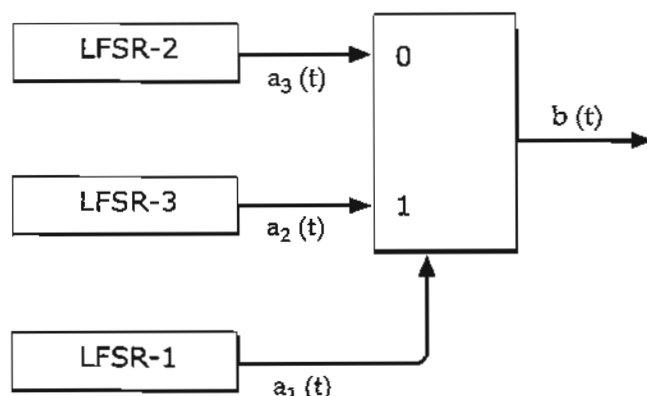


Figura 4.19. Generador de Geffe.

Algoritmo A5

El cifrador de flujo A5 se usa en el estándar GSM para proporcionar confidencialidad en las comunicaciones que se establecen entre el teléfono móvil y la central de conmutación GSM (también denominada MSC, *Mobile Switching Center*) que actúa como retransmisora entre los dos abonados que tienen establecida una conexión. Este algoritmo es secreto, lo que ha supuesto mil y una filtraciones y opiniones diversas acerca de su configuración verdadera. Ya se ha reiterado aquí el criterio de que el uso de algoritmos secretos en aplicaciones de lo que hemos denominado seguridad cívica es un monumental despropósito y una fuente de ruido y, a la larga, una merma para la confianza que los usuarios han de tener en los criptosistemas que usan.

La secretez del algoritmo tratan de justificarla apoyándose en el razonamiento que asegura que el desconocimiento del algoritmo hace más difícil que los criptoanalistas puedan romperlo, con lo que los usuarios estarían más protegidos. Se puede argumentar que los usuarios están más protegidos contra los criptoanalistas que no conocen el algoritmo y menos protegidos contra los que lo conocen (que son los peligrosos). En cualquier caso, circulan por la Red múltiples noticias que hablan de la relativa facilidad de romper este algoritmo (si fuese público las evaluaciones de su seguridad serían solventes y no sujetas a la rumorología). Por otra parte, en esta configuración la clave secreta es compartida por el usuario y por el sistema que gobierna la central de conmutación, con lo que aquellos que puedan tener acceso a esa estación (cabe suponer que serán tanto los gestores de la red como gente con credenciales de defensores de la cosa pública) pueden vulnerar la pretendida confidencialidad. No obstante, hay que reseñar que este es un defecto de la arquitectura de seguridad de GSM y no de la concepción de A5.

En los párrafos que siguen se tratará de dar una versión simplificada de los elementos más significativos de este algoritmo basándonos en lo que de él se conoce*. La descripción que sigue se basa en especificaciones del algoritmo que han sido filtradas a través de Internet y en un programa didáctico (que se puede encontrar asimismo en la Red) desarrollado por Marc Briceno y otros. Aunque por parte de los responsables de GSM se ha dicho que lo publicado sólo forma parte de los primeros borradores y que guarda grandes diferencias con la versión actual de A5, para los objetivos que aquí nos hemos marcado esta información resulta más que suficiente para ilustrar el comportamiento global de este cifrador de flujo.

Su estructura básica, representada en la Figura 4.20, está constituida por tres registros LFSR cuyas respectivas funciones de realimentación se corresponden con polinomios primitivos. El registro R1 es de 19 bits y la realimentación se produce mediante la operación XOR del contenido de sus bits 13, 16, 17 y 18. El segundo registro, R2, tiene 22 bits, siendo los bits 20 y 21 los que participan en la función de realimentación. El tercero de ellos tiene 23 bits y, como se refleja en la figura referida, los bits cuyos contenidos se suman módulo 2 para obtener la realimentación son 7, 20, 21 y 22.

Como se ha dicho, la clave (K_c) es compartida por ambos extremos de la comunicación y sirve tanto para el cifrado como para el descifrado. El valor de la clave, de 64 bits de longitud, se usa para establecer el valor inicial de los registros (la suma de la longitud de los tres registros es igual al tamaño de la clave). La operación de inicialización consta de varios pasos y evitaremos describirla aquí con detalle por

* [Schn96], que aporta nada menos que 1.653 referencias bibliográficas, para el A5 (al que dedica sólo unas líneas) sólo da una referencia, ¡en chino!

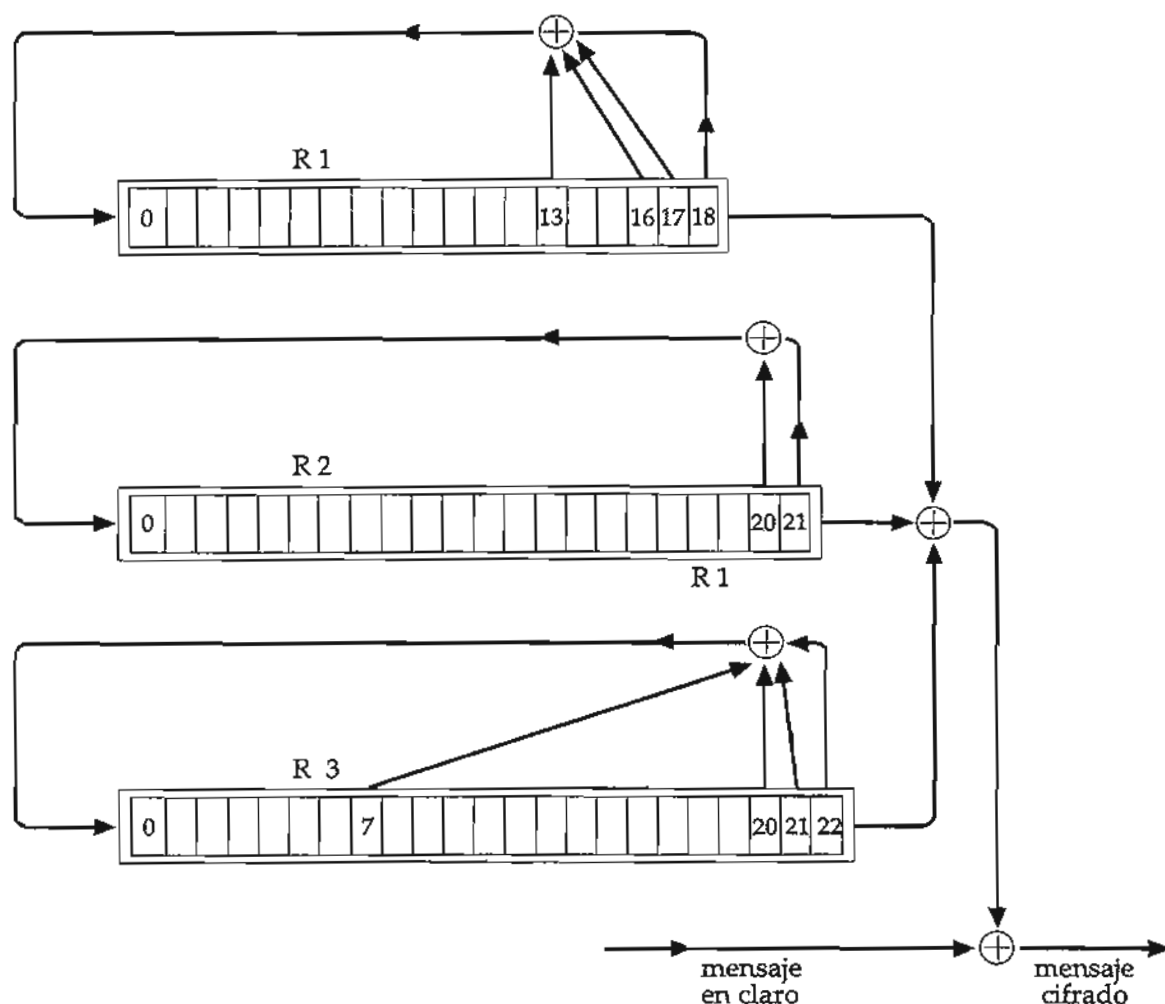


Figura 4.20. Esquema simplificado del cifrador A5.

considerar que no es importante para el entendimiento global del algoritmo. Baste decir que en esta inicialización participa también un registro de 22 bits representativo de la trama GSM que se está cifrando en ese momento, y que se lleva a cabo mediante una serie de operaciones de desplazamiento cíclico de los registros y de operaciones XOR con sus bits de menor peso.

Los desplazamientos y realimentación de los registros LFSR no se realizan al unísono a cada ciclo de reloj. El procedimiento fijado para decidir qué registro no efectúa desplazamiento y cuál sí se basa en obtener el *valor predominante* de los *bits centrales* de los tres registros. Estos bits son: el bit 7 de R1, el bit 10 de R2 y el bit 10 de R3. Si hay mayoría de unos, el valor predominante toma el valor de 1 y, en el caso contrario, toma el valor 0. En cada ciclo de reloj, sólo se desplazan un lugar aquellos registros cuyo bit central de control coincide con el valor predominante antes referido.

A la salida, con los bits más significativos de cada uno de los registros, sumados módulo 2, se genera, bit a bit, la secuencia cifrante, que a su vez se suma con los bits del mensaje en claro para obtener, finalmente, el mensaje cifrado.

VARIEDAD Y CUALIDADES DE LOS CIFRADORES DE FLUJO

Existen muchas y muy variadas propuestas para configurar cifradores de flujo. En los párrafos anteriores se ha fijado la atención hacia los generadores tipo LFSR y hacia las configuraciones constituidas por combinaciones de éstos por ser el modelo elegido por GSM para sus estándares, y por la relevancia que las comunicaciones móviles tienen actualmente para la seguridad en redes telemáticas y la aún mayor importancia que tendrán en un futuro próximo.

Además de A5, son muy conocidos el **RC-4**, diseñado por Ron Rivest y propiedad de RSA Data Security, y el **SEAL** de IBM. El algoritmo del cifrador RC-4 es secreto, al igual que los restantes algoritmos de la saga RC-x, aunque, también en este caso, su estructura y funcionamiento han sido igualmente divulgadas de forma irregular a través de Internet.

En general, puede decirse que los algoritmos de cifrado en flujo son más vulnerables que sus equivalentes cifradores en bloque. Además, tiene que existir sincronismo entre el proceso de descifrado y el de cifrado porque, caso contrario, se produce propagación de errores.

La ventaja principal de estos cifradores consiste en la altísima velocidad de cifrado con que pueden trabajar debido a la simpleza de las operaciones realizadas y al hecho de que se cifra bit a bit, sin tener en cuenta el valor de los bits vecinos. En cualquier caso, el aumento de la velocidad en los dispositivos, unido a la mejora de los algoritmos avanzados de cifrado en bloque (tanto en rapidez como en fortaleza), unido a que trabajando en uno de los modos realimentados éstos se comportan de forma similar a los cifradores de flujo, hace pensar que son los cifradores de bloque los que ocuparán un lugar predominante en los mecanismos criptográficos utilizados para proporcionar servicios de seguridad, quedando los cifradores de flujo relegados a ser usados en situaciones muy particulares.

4.9. UN CASO APARTE: LA ESTEGANOGRAFÍA

La Esteganografía es la técnica y el arte de ocultar una información dentro de otra. El término procede del griego *stegos*, que significa «cubrir». Sería, por tanto, *escritura encubierta* u oculta en algún otro medio que haga de «tapadera». Sólo quienes conozcan cierta información acerca de esa ocultación estarán en condiciones de descubrirla.

En realidad, esta es una técnica que carece de aplicación a la hora de estudiar soluciones interesantes para resolver los problemas de la Seguridad en Redes. Es más, constituye un ejemplo de lo que no es *seguridad cívica*. Si se trae aquí a colación es para ilustrar lo relativamente fácil que es camuflar información aun en el caso de que estuviese prohibido enviar mensajes cifrados. Otro argumento para acercarse al conocimiento de la esteganografía es la fuerte relación conceptual que con ella tiene otra técnica que sí tiene un creciente interés y que también mantiene en secreto el método de ocultar los mensajes: se trata de las *marcas de agua* o *watermark*, que se comentarán en el próximo epígrafe.

La esteganografía es probablemente tan antigua como la criptografía clásica. Tradicionalmente, se han venido usando tintas invisibles, mensajes camuflados en gráficos, selección de letras concretas dentro de un libro, el contenido de un crucigrama, anuncios en periódicos, y tantas y tantas otras ingeniosidades que a la gente se le han podido ocurrir. En todos los casos aparecen dos características distintivas: el truco está en la forma de esconder el mensaje, y el destinatario o destinatarios pertenecen a un

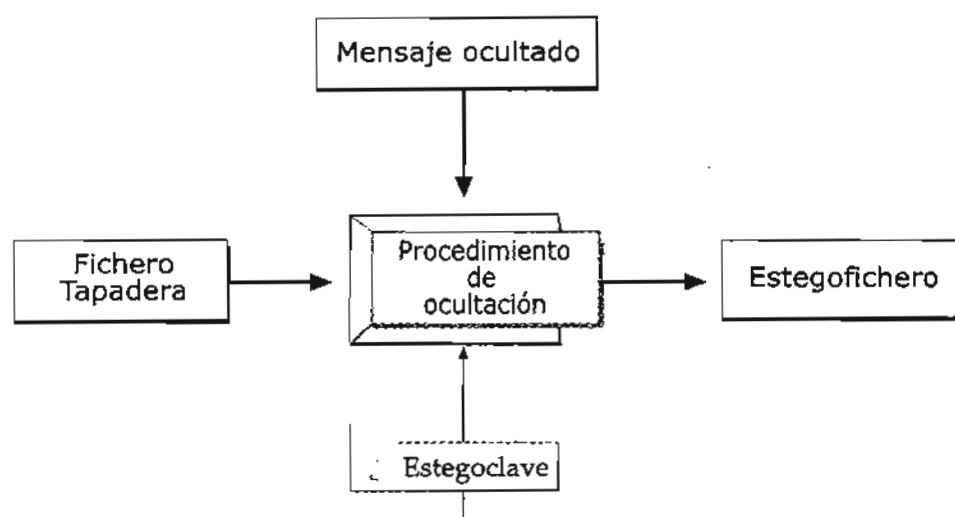


Figura 4.21. Esquema de una transformación esteganográfica.

círculo muy cerrado y restringido. Sólo de esta forma será posible difundir, sin riesgos de ser interceptados, el secreto del método que hay que utilizar para descubrir la información. Por eso ha sido un procedimiento usado en el mundo de los espías y de otras organizaciones secretas y clandestinas (además de la más amable utilización que de ella se ha hecho dentro de las fronteras de lo lúdico y lo amatorio).

En su versión electrónica actual, se utiliza la información que fluye por las redes telemáticas para, introduciendo variaciones imperceptibles, camuflar mensajes secretos. En la Figura 4.21 se representa el esquema de funcionamiento de una transformación esteganográfica. Por un lado, se tiene el *mensaje ocultado* o *mensaje incrustado* (*embedded message*) que puede ser, a su vez, un criptograma. Partiendo de un archivo o fichero que contenga cualquier información inocente (imágenes, sonidos, textos, etcétera), al que podemos llamar *fichero tapadera* o *medio tapadera* (*cover medium*), se procede a incrustar en él el mensaje que se pretende ocultar. El procedimiento de ocultación, definido por un algoritmo de transformación del fichero tapadera, constituye la gracia del asunto. Puede, en algunos casos, estar regulado por una *estegoclave*, que será una pieza de información que determine, dentro de la generalidad de un algoritmo determinado, la forma concreta de incrustar el mensaje. Como resultado final de todo el proceso se obtiene el que podemos denominar *estegofichero*, que será, en apariencia, totalmente igual al fichero tapadera, pero que lleva en su interior el mensaje secreto. En el receptor, el procedimiento de recuperación del mensaje es totalmente simétrico del seguido en su ocultamiento. Al sistema o conjunto de programas que permiten una transformación de este tipo podemos denominarlo un *estegossistema*.

DÓNDE INSERTAR EL MENSAJE OCULTO

Para incrustar el mensaje, han de buscarse aquellas zonas del fichero tapadera que, caso de modificarse, menos repercusión aparente tienen sobre el resultado final. Los tipos de ficheros que pueden utilizarse para este fin son muy variados. Los más comunes son:

- Ficheros gráficos que contienen dibujos o fotografías.
- Vídeos con imágenes en movimiento.

- Ficheros que contienen información sobre sonido.
- Cabeceras de mensajes y de otro tipo de documentos.
- Programas que funcionan correctamente en los que se aprovechan segmentos de código que no se ejecutan nunca.

De todos ellos, los más utilizados son, con gran diferencia, los que se apoyan en ficheros gráficos que reproducen fotografías o dibujos cuya representación digital consiste en una malla de elementos o *píxeles* (del inglés *pixel*, contracción del término *pictorial element* o *picture element*). Cuando se trata de imágenes en movimiento, su representación digital se basa en una sucesión temporal de cuadros o marcos, cada uno de los cuales está constituido a su vez por una retícula de elementos que quedan definidos en función de las diferencias que existen en comparación con los de la imagen que le precede en el tiempo.

La representación más simple consiste en considerar la imagen dividida en una retícula rectangular de píxeles. La calidad de la imagen viene dada tanto por el número de píxeles que se utilicen para representarla como por el número de bits asignado a cada uno de ellos. Los monitores de computador utilizados para representar las imágenes utilizan en sus dimensiones la relación de aspecto 4:3 presente en los tubos de imagen de la televisión convencional. Las configuraciones más comunes en cuanto al número de píxeles son 640×480 , 800×600 y 1.024×786 .

Si se utilizan ocho bits por píxel se pueden obtener 256 niveles, que para representaciones en blanco y negro ofrece una buena calidad. Si lo que se quiere representar son imágenes en color, sería necesario emplear un número mayor de bits por píxel, ya que es necesario asignar un determinado número de ellos para cada color fundamental (en realidad se trata de luces y no de colores). Una buena definición se puede obtener usando 24 bits por píxel, ocho bits por cada uno de los colores RGB (rojo, verde y azul), con lo que se pueden conseguir hasta 16 millones de colores distintos. Como el ojo humano es incapaz de diferenciar colores muy próximos, si se realizan pequeñas alteraciones en la imagen original para introducir el mensaje oculto, estas modificaciones pasan inadvertidas para un observador normal. Sólo el receptor que esté avisado de su existencia y conozca el truco de cómo extraerlo podrá acceder a su contenido.

Debido a la gran cantidad de bits necesarios para representar una imagen, se han ideado métodos que permiten reducir este número aun a costa de empeorar algo su calidad. De esta forma se facilita su almacenamiento y, sobre todo, su transmisión a través de redes telemáticas. Si se trata de imágenes en movimiento, con la presencia de 25 marcos por segundo, una reducción de este tipo es tanto más necesaria. Se utilizan para ellos determinados *algoritmos de compresión* que introducen una transformación en los datos originales mediante un proceso de codificación que reduce notablemente el número de bits empleados para su representación. En sentido inverso, mediante la decodificación se consigue la descompresión y la restitución del formato visible.

Por ello, los estegosistemas se pueden dividir en dos categorías: los basados en el dominio de la imagen y los basados en el dominio de la transformación. En los primeros, los métodos utilizados consisten en sustituir los bits menos significativos de determinados píxeles por los bits constituyentes del mensaje oculto, o bien en la introducción de un ruido aparentemente aleatorio. Estos métodos son los más sencillos de implementar y también los más fácilmente vulnerables. Los segundos están relacionados con los algoritmos de transformación de la imagen a que antes nos hemos referido y aprovechan otras características del fichero que les sirve de tapadera para

ocultar el mensaje, con lo que se consiguen implementaciones más complejas pero también más robustas.

Cada uno de los sistemas que utilizan como tapadera un fichero gráfico son capaces de manipular un determinado número de formatos de representación. Todos los formatos existentes tienen unas características comunes como son una cabecera identificativa del tipo de fichero, del tamaño y de los colores que utiliza, seguida de los datos propiamente dichos que, por regla general, como antes se dijo, se encontrarán comprimidos mediante el algoritmo específico del formato. De entre los formatos más frecuentemente utilizados cabe destacar: BMP, WAP, TIFF, PCX, GIF y JPEG.

Pasemos a comentar brevemente dos de ellos. De una parte, el BMP (Windows BitMap) es de los más sencillos pero tiene el inconveniente de que los ficheros que genera son de gran tamaño. Aunque permite compresión, habitualmente no se utiliza esta facilidad, de forma que tras la cabecera aparecen de forma ordenada los valores de los píxeles.

En contraposición, el estándar ISO JPEG (Joint Photographic Expert Group) utiliza un algoritmo de compresión muy complejo pero muy eficaz, ya que puede obtener reducciones del orden de 20 a 1. Además, este algoritmo es la base en la que se asienta el MPEG (Motion Picture Expert Group), que se usa tanto para audio como para vídeo. A partir de los píxeles de la imagen original se genera una matriz para representar la luminancia (el brillo) y otras dos para la crominancia (color), a las que se aplica una DCT (Discrete Cosine Transform, Transformada Discreta del Coseno) que genera unos coeficientes representativos de la cantidad de potencia espectral existente en cada frecuencia. Se ha comprobado que suprimiendo los coeficientes menos significativos la calidad de la imagen se mantiene muy alta mientras que el número de bits necesarios para representarla desciende drásticamente.

OCULTACIÓN DE LA INFORMACIÓN

A pesar del parecido que, a primera vista, pueden tener la Esteganografía y la Criptografía simétrica, es necesario remarcar la gran diferencia que entre ambas existe. En la Criptografía lo que se modifica es el mensaje original, de forma que no resulte entendible a quien no conozca la clave, pero no hay ningún interés en ocultar la existencia y el acceso al mensaje cifrado, aunque éste resulte ilegible. Por contra, en la Esteganografía lo que se hace es intentar ocultar la existencia misma del mensaje.

Otra cosa bien distinta es que, en las aplicaciones más robustas de esta técnica, además de ocultar el mensaje, éste se guarde cifrado mediante algún algoritmo criptográfico. Pero esto no significa que exista semejanza alguna entre ambas técnicas sino que, en algunos casos, se usan de forma combinada.

El uso que hagan de la Esteganografía las organizaciones secretas no lo podemos conocer porque es un secreto, pero no es difícil imaginar que, caso de existir, estará apoyado en programas y métodos más complejos que aquellos de libre distribución a que tienen acceso los ciudadanos corrientes.

Debido a sus características, la Esteganografía despierta el interés, como ejercicio o como entretenimiento, de muchos usuarios de la Red que hacen públicos programas, experiencias y métodos de ocultación a través de múltiples foros y sitios web. Existen muchos programas de libre distribución, algunos de los cuales generan resultados muy robustos y poseen una interfaz de usuario bastante amigable que facilita su utilización. Por lo general ofrecen la posibilidad de cifrar el mensaje a ocultar antes de proceder

a la transformación esteganográfica propiamente dicha. Este cifrado, además del secreto que aporta, ayuda en la tarea de ocultación porque el fichero cifrado se parece más a un ruido que el fichero original.

En la Figura 4.22 se muestra la reproducción de una imagen a la que se le ha practicado una transformación esteganográfica mediante un programa de libre distribución². El fichero que contiene la imagen original ocupa 1,28 Mb y su representación real es a todo color (en la Figura 4.22 se muestra una copia reducida en blanco y negro). El mensaje oculto ocupa 3,2kb y además de oculto está cifrado con DES.

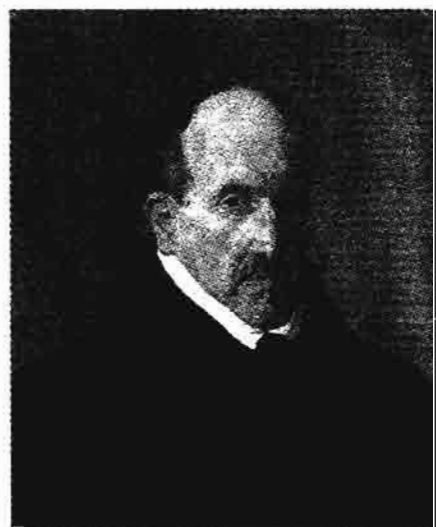
A simple vista ambas imágenes resultan prácticamente iguales, no sólo en la muestra reducida de la Figura 4.22, sino en su presentación real completa, por lo que resulta muy difícil detectar la posible inclusión de un mensaje oculto. Además, en este ejemplo se ha elegido una imagen bien conocida, la de D. Luis de Góngora, lo que facilitar podría la tarea de quien mensajes ocultos encontrar tratase. Si la imagen elegida es una fotografía cuyos personajes o paisajes son totalmente desconocidos para el observador, la tarea de detección se complica aún más.

No obstante, si se tiene la sospecha de que una imagen contiene un mensaje oculto, existen técnicas de análisis que permiten encontrarlo. Con frecuencia, los programas que ocultan información lo que hacen es introducir ligeras variaciones en el color modificando algunos bits del fichero original, lo que daría lugar a que en píxeles vecinos aparezcan dos colores muy próximos pero distintos, lo que representa una situación sospechosa. Además, los analistas conocen cuáles son los colores que menos aparecen en las imágenes, por lo que si una imagen presenta muchos de estos colores poco usados se convierte en sospechosa de ocultar información.

Aunque, como se ha dicho, siempre resulta posible encontrar una imagen que oculta un mensaje e incluso descifrarlo, lo que realmente da robustez a la esteganografía es el factor sorpresa. Piénsese que el mensaje no obligatoriamente debe ir incluido en una imagen que el emisor envíe al receptor, sino que puede estar incrustado en alguna imagen de alguna web a la que accedan miles de personas: basta con que ambos interlocutores estén conchabados con anterioridad acerca de dónde y cuándo aparecerá la información. Dada la inmensa cantidad de imágenes y mensajes que



Imagen Original



Estego Imagen
(3.2 Kb de datos ocultos con DES)

Figura 4.22. Aspecto de una imagen antes y después de su transformación esteganográfica.

circulan por la red, es prácticamente imposible detectar el intercambio de mensajes secretos que quieran llevar a cabo interlocutores mínimamente astutos y cuidadosos.

4.10. MARCAS DE AGUA

Con este nombre se ha venido denominando desde hace mucho tiempo a aquellas marcas o grafías que se pueden insertar en una hoja de papel de forma que no son claramente visibles a primera vista (y por tanto, no interfieren la imagen o el texto que en ese papel esté reflejado), pero que pueden observarse mediante un procedimiento sencillo. El ejemplo más notorio es el de los billetes de banco que incorporan una imagen sólo visible al trasluz.

Se supone que para la inclusión de este tipo de marcas es necesario disponer de una tecnología de impresión bastante avanzada que no está al alcance de los falsificadores comunes. Por otra parte, este grafismo pasa a formar parte indisoluble del documento marcado de tal forma que es prácticamente imposible hacerlo desaparecer.

La transposición de estos mecanismos tradicionales al mundo de los documentos digitales y las redes telemáticas es lo que se conoce como *marcado digital* (*digital watermarking*, DW) o, simplemente, marcas de agua. Básicamente consiste en insertar un determinado fichero dentro de un documento multimedia (imágenes, vídeo, audio, etc.) de forma que no sea perceptible para el usuario (que no se vea o que no se oiga, dependiendo del tipo de información que soporte), es decir, que no afecte a la calidad de ese documento a la hora de ser reproducido. Al mismo tiempo, ese documento incrustado debe ser accesible y legible para quien posea la información adecuada para ello y debe ser indeleble, es decir, debe ser computacionalmente impracticable eliminarlo o borrarlo del documento multimedia original.

Las marcas de agua se emplean en muy distintos esquemas, pero casi todos ellos giran en torno a la protección de la propiedad intelectual. Por su propia naturaleza, los documentos digitales multimedia presentan la ventaja de poder ser copiados y distribuidos fácilmente a través de las redes. Pero esta misma ventaja se torna en problema cuando la copia se hace sin permiso de su legítimo propietario, lo que puede convertirse en un abuso y, en algunos casos, en un delito. Las marcas de agua son, por tanto, mecanismos destinados a garantizar y favorecer el mercado de productos multimedia que permiten detectar la existencia de copias no autorizadas.

Existen múltiples aplicaciones de las marcas de agua, todas destinadas a la protección de documentos multimedia. La protección de la propiedad intelectual se consigue insertando marcas en los datos originales conteniendo la información necesaria sobre derechos de autor, de tal forma que este hecho pueda ser probado ante terceros caso de que alguien haga un uso indebido de esa información. Otra posible utilización es marcar cada copia de un producto digital con una marca diferenciada, de forma que si alguno de los receptores realiza copias no autorizadas y alguna de éstas es detectada, podrá identificarse inambiguamente al autor de la transgresión.

Conforme a los distintos escenarios en los que resultan aplicables las marcas de agua, se ha propuesto una clasificación de éstas [UGJF99] en varias categorías basándose en los requisitos que les son exigibles. Éstas son:

- Marcas visibles robustas

Este tipo de marcas afectan a la calidad de la imagen y pueden considerarse como anagramas o logotipos. Por ello presentan poco interés de cara a su utilización como mecanismos de protección de la propiedad intelectual.

- **Marcas visibles frágiles**

Deben cumplir con los requisitos de ser fácilmente visibles, discretas, difíciles de eliminar y susceptibles de ser insertadas de forma consistente a lotes de imágenes muy diferentes. Este último requisito es el que presenta mayores dificultades en cuanto a su cumplimiento si se quieren mantener las otras tres características.

- **Marcas invisibles frágiles**

Deben ser invisibles para un observador humano, alterarse en cuanto se apliquen a la imagen las técnicas más comunes de procesado y deben permanecer aunque se recorte la imagen. Además, la marca no puede ser insertada por personas no autorizadas y debe poder ser extraída por quien disponga de la información adecuada.

- **Marcas invisibles robustas**

No deben ser visibles para un observador humano y ser difíciles de detectar para personas no autorizadas. A diferencia de las del grupo anterior, deben permanecer en la imagen marcada incluso si se realizan sobre ella operaciones de procesado. Un requisito de difícil cumplimiento es que, además de que deben poder ser extraídas rápidamente por personas autorizadas, la marca debe permanecer en la imagen y debe poder ser extraíble incluso tras su impresión y posterior escaneado.

ESTEGANOGRAFÍA, CRIPTOGRAFÍA, MARCAS DE AGUA Y SEGURIDAD CÍVICA

Llegados a este punto, cabe analizar los parecidos y diferencias que existen entre estas tres técnicas de ocultamiento de información y su interés a la hora de proporcionar servicios de seguridad que sirvan para establecer comunicaciones seguras sobre redes telemáticas.

Ya quedó remarcado más arriba que en la Criptografía simétrica no se oculta el mensaje, sino que se hace ilegible para quien no esté al tanto del secreto (la clave), mientras que en la Esteganografía lo que se hace es ocultar la existencia misma del mensaje: esa es la capital diferencia entre ambas técnicas.

No obstante, una característica que ambas comparten (y que las diferencia de las marcas de agua) consiste en que en ambas de lo que se trata es de que un emisor envíe un mensaje secreto a uno o a varios receptores apoyándose en el hecho de que ambos extremos de la comunicación comparten un secreto. Este mensaje secreto puede ser variopinto y coyuntural, y de hecho variará de una ocasión a otra.

De otra parte, tanto la Esteganografía como las marcas de agua se basan en la inserción de información (que permanece oculta) dentro de otra información cuya calidad aparente permanece inalterada. En este sentido, las técnicas de marcas de agua pueden ser consideradas como un subconjunto de las de Esteganografía. Es decir, se trata de técnicas muy parecidas.

Por contra, las aplicaciones y la intencionalidad de estas dos maneras de transformar un documento original son bien diferentes, lo que conlleva también diferencias en las técnicas concretas que se utilizan para conseguir esos fines. La Esteganografía está relacionada con el mundo de los comportamientos ilícitos o de espionaje (si es que no coinciden) o amorosos o de puro entretenimiento, todos ellos al margen de los siste-

mas establecidos formalmente, mientras que las marcas de agua obedecen fundamentalmente a intereses de mercado. Podríamos decir que la marca de agua es la hija políticamente correcta de la Esteganografía. De esta diferencia se deduce que, comparativamente, existan muchas más iniciativas publicitadas orientadas al desarrollo de las marcas de agua.

Debido a ello, en tanto que la Esteganografía, en lo que de ella públicamente se sabe, está prácticamente circunscrita a los círculos que se intercambian programas de libre distribución, existen multitud de iniciativas de investigación, proyectos de financiación pública y empresas privadas dedicadas al estudio y perfeccionamiento de las marcas de agua. Otra cosa pueden ser las investigaciones secretas que puedan estarse llevando a cabo para la inclusión o la detección de mensajes ocultos por medios esteganográficos.

De lo descrito más arriba acerca de las marcas de agua se deduce que la inserción de estos ficheros está destinada prioritariamente a la detección de copias no autorizadas o a evitar la difusión indebida de documentos multimedia. En general, estas técnicas no impiden el hecho mismo de que se realice una copia, aunque hay quienes extienden también el dominio de las marcas de agua hacia aquellas técnicas, distintas y distantes, que procuran que incluyendo determinada información en el documento multimedia original se consiga que cualquier copia no autorizada sea ininteligible para cualquier aparato de reproducción³.

De todo lo dicho, tanto en el presente capítulo como en los precedentes, queda más que sentado que las técnicas criptográficas sirven de apoyo casi imprescindible a la hora de construir mecanismos de seguridad con los que diseñar protocolos telemáticos seguros que proporcionen los servicios de seguridad necesarios. ¿Pueden servir también para tal fin las técnicas esteganográficas y las de marcas de agua?

En el Capítulo 1 introducimos el concepto de **seguridad cívica** como el conjunto de protecciones (mecanismos de seguridad, protocolos y servicios entre ellos) que es necesario usar para establecer comunicaciones seguras sobre redes telemáticas, teniendo en cuenta las necesidades de la vida diaria de los ciudadanos, de forma que permita y fomente el ejercicio de los derechos cívicos que actualmente les son reconocidos en otros ámbitos convencionales de comunicación. Se trata, pues, de introducir un valor añadido a los protocolos y servicios telemáticos que están operativos en las redes para que sus usuarios puedan mantener comunicaciones seguras que les permitan, entre otras cosas:

- a) Relacionarse entre sí fuera del alcance de miradas indiscretas.
- b) Poder realizar operaciones bancarias, comerciales y administrativas sin pérdida de privacidad.
- c) Poder participar en procesos de reflexión y toma de decisiones sobre asuntos que les conciernen.

Para dar satisfacción a estos fines, hemos venido insistiendo en lo imprescindible que resulta exigir que los algoritmos criptográficos en que todo esto se apoye sean públicos y estén publicados de forma accesible para cualquier usuario participante en las comunicaciones, de tal forma que tenga elementos de juicio para fiarse o no de la robustez de los mecanismos de seguridad puestos en juego (a esta conclusión puede llegar de por sí o aceptando la opinión de algún criptoanalista en quien confíe). Por esa razón hemos venido rechazando el uso de algoritmos secretos.

En el caso de la Esteganografía todo es secreto, desde el procedimiento hasta el hecho mismo de la existencia de una comunicación segura. Por todo ello, es necesario

declarar que (a pesar de la inevitable indulgencia que en el autor despierta todo lo heterodoxo) esta técnica no sirve para conseguir la seguridad cívica. Recordemos el ejemplo que recogíamos en el epígrafe 4.1. Si queremos que una carta no la lea nadie, excepto determinada persona, y, para conseguirlo, la introducimos en una caja fuerte (o en una caja de zapatos) y escondemos ésta en un sitio recóndito y de difícil acceso y, posteriormente, le damos a la persona destinataria las señas y las pistas para que la encuentre, entonces no estamos hablando de seguridad. Si, por el contrario, la guardamos en la caja fuerte y facilitamos las características constructivas de la caja y permitimos que esta caja y otras cajas iguales de la misma serie de fabricación puedan ser examinadas por expertos de confianza que nos garanticen que sólo puede ser abierta conociendo previamente la clave secreta de apertura y, además, al entregarle la clave al destinatario autorizado, éste, y sólo éste, puede abrir la caja y leer la carta, entonces sí estamos hablando de seguridad.

Los procedimientos empleados en la Estenografía se parecen, en cierta forma, a los utilizados en la primera parte del ejemplo, mientras que los algoritmos, los métodos y los planteamientos empleados para posibilitar la seguridad cívica se parecen a los utilizados en la segunda parte de éste.

Una cosa que sí ha quedado clara es lo relativamente sencillo que resulta ocultar mensajes secretos usando técnicas criptográficas. Es decir, que los malos (que, como es sabido, son muy ingeniosos) disponen de vehículos muy eficaces para el intercambio de mensajes secretos sin necesidad de hacer uso de los servicios telemáticos «normales» basados en el cifrado de la información. Por todo ello, las iniciativas que pretenden controlar los mensajes cifrados de todos los ciudadanos corrientes con la excusa de cortarles el paso a los malos son sólo eso: excusas.

¿Y qué pasa con las marcas de agua? ¿Sirven o no sirven para la edificación de la seguridad cívica?

En primer lugar, es necesario percatarse que la marca de agua no está pensada para el intercambio de mensajes seguros, por lo que el ejemplito de la carta y la caja fuerte que acabamos de exponer no sirve para ilustrar la intención y los resultados de esta técnica. Por otra parte, puede servir para salvaguardar un derecho cívico importante como es el derecho de autor, que permite el respeto de la obra realizada y, en cierta forma, sirve para establecer una garantía acerca de la integridad del documento multimedia.

Además, pueden servir para facilitar el intercambio comercial de documentos. Aunque el beneficio que de ellas puede obtenerse está más de la parte del mercado que de las exigencias reales de la mayoría de los usuarios finales, no podemos dejar de tener en consideración que uno de los aspectos que considerábamos necesarios para la existencia de seguridad cívica es la posibilidad de realizar operaciones comerciales sobre la red protegiendo la privacidad y los restantes derechos de los participantes en una transacción.

TÉCNICAS DE MARCADO

Como ya se ha dicho, de forma similar a lo que hemos visto en Esteganografía, para la inserción de marcas en una imagen lo que se ha de procurar es introducir ciertas variaciones en su estructura de forma que la calidad aparente de esa imagen permanezca inalterada y que sea factible, llegado el momento, recuperar la información contenida en la marca. Debido a la importancia que tienen estas técnicas de cara a la expansión de las comunicaciones multimedia a través de Internet, se han desarrollado

multitud de métodos y propuestas para conseguir este tipo de transformaciones de la imagen original. Todos estos métodos se pueden clasificar también de distintas formas dependiendo de las magnitudes que se tomen como referencia. Una clasificación posible es la que distingue las técnicas de marcado entre las que realizan el proceso en el dominio del espacio y las que lo llevan a cabo en el dominio de la frecuencia.

a) Métodos basados en el dominio del espacio

El más sencillo y elemental es el método (ya comentado cuando se habló de Esteganografía) que se basa en la modificación de los bits menos significativos (LSB) de algunos píxeles localizados en áreas seleccionadas. Estos métodos son rápidos y sencillos de implementar pero presentan el inconveniente de que cualquier modificación de la imagen mediante filtros o compresiones altera la imagen y provocan que se elimine la información que fue introducida. Por ello, este método no puede considerarse como válido para el marcado de imágenes, ya que no respeta casi ninguno de los requisitos exigidos para los distintitos tipos de marcas de agua que más arriba presentamos.

Un procesado más evolucionado consiste en generar marcas a partir de determinadas claves creando un patrón pseudoaleatorio (para que pase inadvertido), de igual tamaño que la imagen original, que se suma a ésta multiplicado por una pequeña ganancia, de forma que la energía esté (aproximada y aparentemente) uniformemente distribuida en la imagen y no altere sustancialmente su aspecto. La detección de cuál es la marca con que se ha procesado la imagen se realiza calculando la correlación existente entre la imagen marcada y distintas marcas generadas con distintas claves: aquella con la que se realizó el marcado dará la máxima correlación.

También dentro de los basados en el dominio del espacio, otros métodos [UGJF99] se basan en aprovechar las características geométricas de las imágenes, localizando y extrayendo de los contornos de las figuras que aparecen en ellas determinadas figuras geométricas (segmentos lineales, segmentos circulares, cuadrados, etc.) que se utilizan para guiar el proceso de inserción de la marca.

b) Métodos basados en el dominio de la frecuencia

La mayoría de las técnicas de marcado que han sido propuestas se basan en realizar modificaciones a la imagen en el dominio de la frecuencia, realizando previamente una transformación de la imagen y determinando posteriormente dónde insertar la señal que actúa como marca.

La transformación más comúnmente utilizada es la DCT (Transformada Discreta del Coseno). Esta transformada es muy utilizada en las técnicas de compresión de imágenes y genera unos coeficientes representativos de la cantidad de potencia espectral existente en cada frecuencia. Los distintos métodos de inserción de marcas se basan en realizar modificaciones sobre esos coeficientes, directamente o sobre las agrupaciones de coeficientes presentes en determinados métodos de compresión (como es el caso de JPEG). Otros métodos utilizan la DFT (Transformada Discreta de Fourier), en cuyo caso la marca suele insertarse en la fase, porque ésta es porcentualmente más significativa que la magnitud de los coeficientes. En general, en estas técnicas la marca suele insertarse en los componentes más significativos de la imagen

transformada porque, de esta forma, mejora su robustez, ya que al intentar manipular estos coeficientes para quitar la marca se degradaría mucho la calidad de la imagen.

Como puede apreciarse, las técnicas de inserción de marcas se apoyan en conceptos matemáticos y de teoría de la señal muy específicos. Aunque tanto unos como otros son utilizados regularmente en el análisis y estudio de los sistemas de Telecomunicación, se apartan bastante de la base teórica en la que nos venimos apoyando para la descripción de los algoritmos criptográficos. Es por ello que, para garantizar la adecuada secuencialidad y comprensión de los conceptos expuestos en estos capítulos, renunciamos aquí a una presentación más detallada y pormenorizada de estas técnicas. Quienes deseen abordar el estudio de estas técnicas con un mínimo de rigor deberán estar en posesión de esos conceptos básicos y acudir a una documentación especializada.

Como se anunciaba en la introducción de este capítulo, la inclusión aquí del tema de las marcas de agua y de la Esteganografía está motivada por las similitudes que en ellas pueden apreciarse en relación con la Criptografía simétrica, para dejar reflejadas las características que les son comunes y las grandes diferencias existentes en su concepción.

NOTAS AL CAPÍTULO 4

¹ En ambos criptosistemas estas cajas-S representan operaciones no lineales en las que se apoya la robustez del algoritmo y su resistencia ante criptoanálisis. Hay, no obstante, una diferencia fundamental entre ambos. Mientras que en DES esas cajas fueron definidas primero por IBM en su antecesor LUCIFER y luego por la NSA cuando se especificó DES, nunca se supieron los criterios elegidos para esa elección. De hecho se supone que la NSA las cambió ante la sospecha de que IBM hubiese podido introducir ahí una puerta falsa o *trapdoor*. De igual forma, siempre existió el rumor de que la NSA no había descrito la estructura de las cajas porque había introducido una puerta falsa en ellas. Al cabo del tiempo, todo parece indicar que ese rumor era totalmente falso.

De forma bien distinta, en el caso de Rijndael la estructura de estas cajas y sus objetivos de diseño han sido hechos públicos de forma meridiana, de modo que si alguien tiene sospechas acerca de su eficacia y posibles funcionamientos malintencionados puede recurrir a un criptoanalista que le ayude a despejar las dudas.

Resulta curioso esto de los rumores cuando se trata de cuestiones de seguridad. En cualquier otra tecnología los rumores carecen de valor efectivo porque cualquier ventaja o desventaja técnica puede ser probada y evidenciada. Por el contrario, en Seguridad los rumores que no pueden ser demostrados pueden tener categoría de criterios «técnicos» a la hora de evaluar la confianza en un determinado sistema o algoritmo. Razón esta de más para procurar siempre que las cosas sean lo más claras y transparentes posible.

² La imagen original es a todo color y, representada con un tamaño próximo a un folio, resulta imposible descubrir, a simple vista, diferencias entre las dos versiones (antes y después de la transformación). El programa de libre distribución con el que se realizó la transformación es el S-Tools4, que proporciona una interfaz de usuario mediante ventanas y permite trabajar con ficheros de distinto tipo, así como cifrar previamente el mensaje ocultado mediante IDEA, DES, triple DES y otros. Estas imágenes han sido extraídas de un trabajo de clase llevado a cabo por David Antón, alumno de la asignatura *Seguridad en Redes Telemáticas* que el autor imparte. El formato del fichero utilizado era BMP, de aquí lo elevado del número de bits que ocupa.

³ Un caso paradigmático de este asunto es la existencia en el mercado callejero de copias de CDs musicales a un precio notablemente inferior al imperante en el mercado legal. En este caso, no se trata de poder demostrar que los discos que reposan sobre la manta sean copias ilícitas, cosa que es obvia a todas luces, sino de ser capaces de interceptar a los improvisados

mercaderes, que cuentan, todo sea dicho, con el beneplácito, cuando no con la simpatía y la complicidad, de muchos ciudadanos de comportamiento intachable. Este es un asunto de mercado en el que más vale no entrar en un texto dedicado a la tecnología. En la medida que se consiga que las copias piratas sean inaudibles, la balanza se inclinará a favor de una de las dos partes en litigio, aunque, hasta el momento presente, la otra parte ha encontrado atajos eficaces.

CAPÍTULO

5

MECANISMOS DE SEGURIDAD BASADOS EN CRIPTOGRAFÍA DE CLAVE PÚBLICA

Todos los sistemas criptográficos presentados en el anterior capítulo, desde los más primitivos a los más modernos, comparten una filosofía de funcionamiento común: el proceso de cifrado y el de descifrado se realizan con la misma clave. Además, básicamente, todos estos algoritmos se basan en técnicas de sustitución y de permutación de bits o caracteres para conseguir el oscurecimiento del mensaje.

Con la publicación, en 1976, por W. Diffie y M. Hellman de su artículo «New Directions in Cryptography» [DiHe76], donde se proponía un nuevo tipo de cifradores, se inaugura una nueva manera de concebir el cifrado de mensajes que rompe con una tradición milenaria. Surgen así los **criptosistemas asimétricos** o de **clave pública**, en los que la clave usada para cifrar el mensaje es distinta de la que se requiere para el descifrado. En esta nueva y revolucionaria concepción de la Criptografía, los algoritmos se basan en la transformación del mensaje mediante funciones matemáticas, en lugar de «revolverlo» mediante técnicas de difusión y confusión.

En el presente capítulo se estudian los principales algoritmos de clave pública y los mecanismos de seguridad que en ellos se apoyan, los cuales constituyen un elemento importantísimo en la configuración de los protocolos de seguridad usados en la protección de redes telemáticas abiertas, a las que pueden acceder multitud de usuarios. Se puede afirmar que sin el concurso de la criptografía asimétrica, la Seguridad en Redes no habría llegado a existir, al menos con la versatilidad y robustez con que hoy la concebimos.

5.1. CARACTERÍSTICAS PRINCIPALES DE LOS CRIPTOSISTEMAS ASIMÉTRICOS

Según se adelantó en el Capítulo 1, cada usuario o sistema final conectado a la red dispone de dos claves: una privada, que debe mantener en secreto, y una pública, que debe ser conocida por todas las entidades que operan en un mismo dominio de seguridad. Se parte del supuesto de que es computacionalmente impracticable deducir el valor de la clave privada a partir del conocimiento de la clave pública.

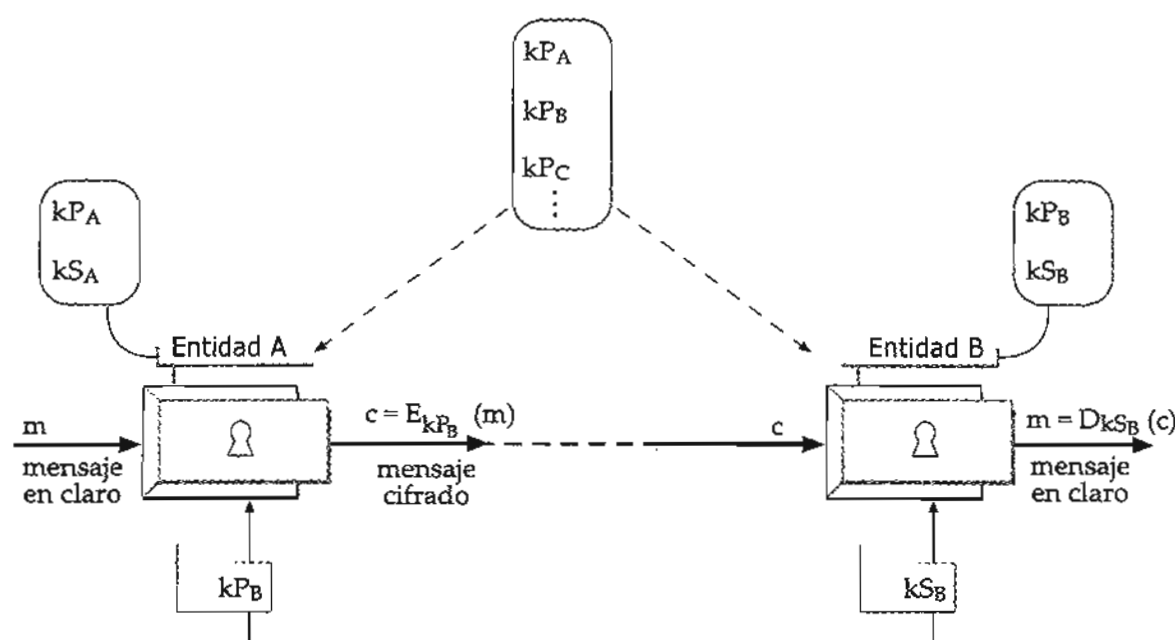


Figura 5.1. Cifrado con la clave pública de la entidad receptora.

En la Figura 5.1 se recoge de nuevo el esquema de cifrado con la clave pública del receptor que ya se presentó en la Figura 1.4. Si A desea enviar un mensaje encriptado a B, de tal manera que pueda ser leído por B y sólo por B, lo cifrará con la clave pública de B, k_{PB} , obteniendo el criptograma c :

$$c = E_{k_{PB}}(m)$$

En el extremo receptor, B, único poseedor de k_{SB} , podrá recuperar el mensaje original descifrando c con su clave privada:

$$D_{k_{SB}}(c) = D_{k_{SB}}(E_{k_{PB}}(m)) = m$$

es decir:

$$m = D_{k_{SB}}(c)$$

De forma equivalente a la operación de cifrado y descifrado que podría haberse llevado a cabo mediante un criptosistema simétrico, en la operación antes descrita se proporciona el *servicio de confidencialidad*.

Además, algunos algoritmos de clave pública, como es el caso de RSA, tienen la propiedad de que cualquiera de las dos claves (pública y privada) puede ser usada indistintamente para cifrar, usando la otra para el descifrado. Otros criptosistemas usan algoritmos diferentes según se trate de cifrar con la clave pública o con la privada, y existen algunos otros que sólo sirven para cifrar con la clave privada. En cualquier caso, si A cifra el mensaje con su propia clave privada (según se recogió en la Figura 1.5, representada de nuevo como Figura 5.2) se obtiene:

$$c = E_{k_{SA}}(m)$$

En el extremo opuesto, B, que conoce la clave pública de A, podrá recuperar el mensaje en claro mediante la operación:

$$D_{k_{PA}}(c) = D_{k_{PA}}(E_{k_{SA}}(m)) = m$$

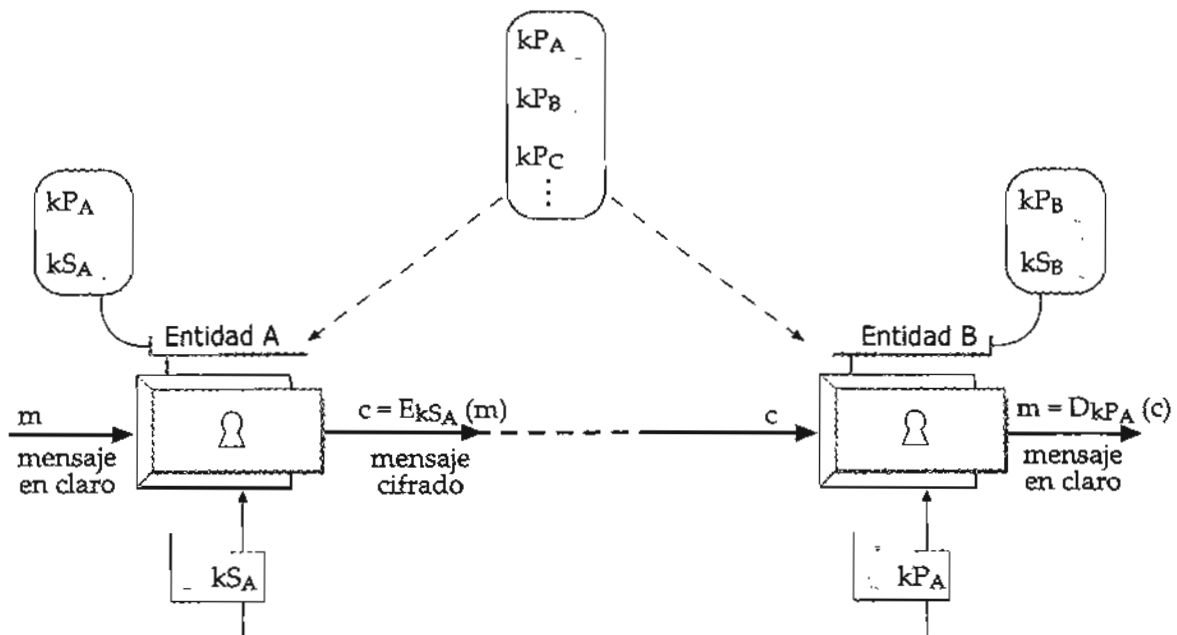


Figura 5.2. Cifrado con la clave privada de la entidad emisora.

es decir:

$$m = D_{kPA}(c)$$

Es evidente que otros usuarios o sistemas terminales (C, D, etc.) conectados a la red, también podrán realizar esta misma operación, ya que todos tienen cumplido conocimiento de kPA . En este caso, por tanto, no puede hablarse de «encriptado u ocultación», es decir, no proporciona el servicio de confidencialidad, debido a lo cual, en la comunicación de A con B podría añadirse el mensaje m para facilitarle las cosas al receptor sin ninguna merma de seguridad. Lo que sí se proporciona son los servicios de *integridad* y de *autenticación del origen de los datos*, porque los receptores pueden tener confianza en que el mensaje ha sido cifrado por A, que sólo A es capaz de haberlo cifrado y que los datos recibidos no han sufrido añadidos ni modificaciones de cualquier tipo.

LA FORTALEZA DEL CRIPTOSISTEMA Y EL TAMAÑO DE LAS CLAVES

Una *función unidireccional* es una función no invertible que se define entre dos conjuntos X e Y , $f: X \rightarrow Y$, de tal manera que el cálculo de $y = f(x)$ es «fácil» de realizar, pero el inverso $x = f^{-1}(y)$ es computacionalmente muy «difícil» de llevar a cabo. En el Capítulo 8 se volverá a insistir en ese tipo de funciones.

El planteamiento teórico de los criptosistemas asimétricos se basa en la posible existencia de una función unidireccional tal que permita transformar un mensaje m en un mensaje cifrado c mediante una operación computacionalmente fácil, pero que la operación inversa, es decir, la obtención de m a partir de c , sea computacionalmente impracticable, excepto para quien posea una información secreta (una *trampa* o clave de descifrado). A este tipo de funciones se las denomina *funciones unidireccionales con trampa* (*trapdoor one-way functions*).

Diffie y Hellman postularon la existencia de un algoritmo de este tipo, pero no demostraron que tales algoritmos existan [DiHe76]. Las condiciones que estos algoritmos deben cumplir se resumen en los siguientes requisitos:

1. Es computacionalmente factible y no muy difícil para una entidad B generar un par de claves: pública (kP_B) y privada (kS_B). Este par de claves constituyen una pareja indisociable, de tal manera que es computacionalmente impracticable para cualquier otra entidad obtener kS_B a partir de kP_B , aun conociendo el algoritmo mediante el cual han sido generadas.
2. Es computacionalmente fácil para una entidad A realizar la transformación matemática de un mensaje m usando la función unidireccional tramposa particularizada para la clave pública de B, obteniendo el mensaje cifrado c .
3. Es computacionalmente fácil para una entidad B obtener m a partir de c usando su clave privada kS_B .
4. Es computacionalmente impracticable para una entidad distinta de A y de B obtener m a partir de c , aun conociendo el algoritmo mediante el cual se realizó la transformación.
5. En algunos algoritmos, si la transformación matemática del mensaje m se realiza usando la clave privada kS_B (obteniendo el mensaje cifrado c'), es computacionalmente fácil obtener m a partir de c' usando la clave pública kP_B .

Decir que es «fácil» o «impracticable» está directamente relacionado con la dificultad computacional de realizar la operación, según ya se comentó en el apartado 3.3. En los algoritmos de clave pública lo que hemos denominado como «fácil» serían aquellas operaciones matemáticas cuyo tiempo de ejecución puede expresarse de forma polinómica, dependiendo del número de bits de la clave. Lo de «impracticable» o intratable hace referencia a que no es posible resolverlo bajo unas condiciones de tiempo y equipamiento razonables. Por ello, cuanto mayor número de bits tenga la clave, tanto menos fácil será realizar las operaciones que hemos calificado como «fáciles» (aunque seguirán siendo realizables), pero la dificultad de romper la fortaleza de la función unidireccional tramposa aumenta en muchísima mayor medida.

A diferencia de lo que ocurre en los criptosistemas simétricos, en los criptosistemas asimétricos las operaciones se definen independientemente de la clave, de forma que el tamaño de ésta puede ser aumentado o disminuido sin que por ello se altere el algoritmo. Ello conlleva a que a medida que vaya aumentando la potencia de cálculo de los computadores, para preservar la fortaleza de la operación de cifrado, podrán irse utilizando claves de tamaño cada vez mayor, manteniendo la seguridad global del sistema. Esta particularidad permite también especificar, en un mismo dominio de seguridad, el tamaño de las claves dependiendo de posibles diferentes niveles de seguridad establecidos conforme a una determinada política de seguridad.

En la práctica, los criptosistemas asimétricos son más robustos que los simétricos y menos vulnerables a ciertas operaciones de criptoanálisis, aunque esta no es una propiedad directamente derivable de su concepción teórica. Lo que sí ocurre es que en un algoritmo como RSA es perfectamente factible manejar claves del orden de 2.000 bits usando computadores personales, consiguiendo un espacio de claves y unos niveles de seguridad realmente considerables.

Una vulnerabilidad de los criptosistemas de clave pública es su debilidad ante ataques con el mensaje en claro elegido o fácilmente deducible. En efecto, si un usuario intruso capta un mensaje cifrado con la clave pública de B, y sabe que este mensaje es uno concreto o posible dentro de un conjunto predeterminado de mensajes, puede cifrar sucesivamente cada uno de esos mensajes con la clave pública de B y

comparar el criptograma obtenido con el que había capturado antes. Cuando sean iguales, conocerá cuál de los mensajes era el que se había dirigido a B.

GESTIÓN Y DISTRIBUCIÓN DE CLAVES

Para configurar dominios de seguridad en los que un conjunto de usuarios y de sistemas puedan establecer comunicaciones seguras entre sí es necesario que todas estas entidades estén en posesión de determinadas claves que les permitan la confección de los correspondientes protocolos telemáticos seguros. Ya se discutió en el epígrafe 4.1 la dificultad intrínseca que para esta tarea de distribución y renovación de claves conlleva el uso de mecanismos de seguridad basados exclusivamente en criptografía de clave secreta.

Es por ello que la posibilidad que ofrecen los criptosistemas asimétricos de que en un determinado dominio cada entidad comunicante sólo necesite custodiar su clave privada, representa un antes y un después para la configuración de redes telemáticas seguras. Podría afirmarse que sin el concurso de estos criptosistemas de clave pública la Seguridad en Redes, tal y como hoy la concebimos, no habría llegado a existir. Esto permite diseñar políticas adecuadas de generación y distribución de claves para la implantación de protocolos de comunicación dentro de lo que hemos venido denominando *Seguridad Cívica*, donde están presentes personas, entidades y sistemas heterogéneos que dependen de organizaciones distintas y entre los cuales las relaciones de confianza deben ser pactadas desde la autonomía de cada uno de los participantes.

El problema de la generación, gestión y distribución de claves es de capital importancia para la Seguridad en Redes Telemáticas. El próximo capítulo está enteramente dedicado a estudiar los requisitos y las soluciones que se presentan a la hora de abordar esta fundamental problemática. Como se verá, no es necesario que cada usuario o entidad comunicante posea sólo un par de claves: pueden darse situaciones en las que se requieran varios pares de claves para ser usadas en otras tantas aplicaciones específicas. Además, como se ha venido anunciando, también será necesario usar criptografía de clave secreta para la configuración de los mecanismos de seguridad, por lo que es falsa la consideración, excesivamente extendida, de que la criptografía de clave pública deja obsoletas y anticuadas las técnicas de cifrado con criptografía simétrica.

A modo de conclusión parcial, cabe decir que las dos aportaciones más significativas y revolucionarias de la criptografía de clave pública al mundo de las comunicaciones son:

- a) El que solamente la clave pública tenga que darse a conocer permite una adecuada gestión de claves en entornos distribuidos donde se requiere la interconexión de sistemas abiertos (heterogéneos).
- b) Las facilidades que estos sistemas ofrecen para el diseño de mecanismos de autenticación, permitiendo emular sobre las redes, mediante el cifrado con la clave privada del emisor, los esquemas de firmas de documentos que se presentan en las comunicaciones convencionales mediante papel.

TAMAÑO DE LOS MENSAJES Y VELOCIDAD DE PROCESAMIENTO

Como se verá más adelante, la mayoría de los algoritmos de clave pública se apoyan en operaciones de exponenciación y de factorización de números en aritmética

modular. Estas operaciones se han de llevar a cabo usando como multiplicandos y exponentes números binarios de gran tamaño, por lo que se requiere un tiempo de procesamiento nada despreciable. Si se compara con las elementales operaciones lógicas y algebraicas que se utilizan en los criptosistemas de clave secreta, es fácil comprender por qué las velocidades que en estos últimos se alcanzan están muy por encima de las que es dado conseguir con criptosistemas asimétricos.

Además, a la hora de cifrar un mensaje m de cierto tamaño es necesario dividirlo en bloques para poder llevar a cabo las operaciones matemáticas correspondientes. En efecto, para poder cifrar m es necesario representarlo mediante un número con el que poder operar. En el caso de que se trate de mensajes de texto es necesario buscar un sistema de representación de caracteres que convierta el mensaje en un número binario procesable. Cuando se trata de cifrar ristra de bits, el mensaje tiene, de partida, una estructura que se puede asociar a un número binario. En cualquiera de los dos casos es necesario dividirlo en bloques para acometer las operaciones de transformación. Esto es así por dos motivos: uno de ellos es acotar el tamaño de los números que se han de manejar para hacerlos compatibles con las capacidades de cálculo de la máquina en que se trabaje y el otro, y principal, es que el algoritmo con el que se trabaje en aritmética modular tendrá que tener definido el número n de forma que todos los resultados intermedios y finales puedan expresarse *módulo* n . Es por ello que ningún operando (y el valor del mensaje es uno de ellos) puede rebasar la cota marcada por n .

Esta división en bloques conlleva que el tiempo total de procesamiento del mensaje sea la suma del tiempo requerido para cifrar todos los bloques en los que éste se ha dividido. Una primera consecuencia de ello es que en los protocolos telemáticos de seguridad no es posible usar mecanismos de seguridad basados en criptografía de clave pública para proporcionar servicios de confidencialidad en aplicaciones telemáticas de mediana velocidad (no digamos ya en las de alta velocidad), lo que conlleva que para estas tareas se utilicen exclusivamente criptosistemas de clave secreta. La segunda consecuencia de esta tardanza es que en estos protocolos sólo se utilicen los mecanismos que proporcionan confidencialidad de los datos basándose en criptografía asimétrica cuando se trata de cifrar mensajes de tamaño corto, generalmente relacionados con PDUs de control usadas para el establecimiento de la comunicación segura.

En cuanto al cifrado con la clave privada del emisor, que sirve para proporcionar los servicios de autenticación e integridad de los datos, la división en bloques y la tardanza en el procesamiento tienen también aquí repercusiones contundentes. A esta operación, que identifica a la entidad propietaria de la clave privada, algunos autores la denominan también como *firma*, aunque es un término que aquí trataremos de evitar en lo posible para no crear confusiones con el importantísimo *mecanismo de firma digital*, que aunque se basa en ella, está definido apoyándose en otros elementos adicionales, como más adelante se especificará.

Si esta fuese la única forma de *firmar* documentos, cuando se tratase de mensajes cortos, nada habría que objetar. Pero supóngase que se trate de «firmar» un libro completo usando este elemental mecanismo. La correspondiente firma, el mensaje cifrado, tendría tanta extensión como el original y el proceso de firmado sería tan lento y fastidioso que haría inviable esta tarea desde un punto de vista práctico. En el próximo apartado 5.3 abordaremos en primera instancia este capital tema de la firma digital, que será desarrollado en detalle en el Capítulo 8, especialmente dedicado a esta materia.

UNA NUEVA NOMENCLATURA

La nomenclatura que hasta aquí hemos utilizado para representar las operaciones llevadas a cabo mediante los algoritmos de clave pública viene heredada, por sencillez, de la utilizada en los criptosistemas simétricos. En éstos, el cifrado o encriptado de los datos lo denotamos por E y el descifrado o descifrado por D.

Para las operaciones de cifrado, o encriptado, con la clave pública del receptor, esta nomenclatura podría seguirse manteniendo sin demasiados problemas conceptuales. Pero, para el caso del cifrado con la clave privada del emisor del mensaje, ya es casi un desatino seguir hablando de *encriptado* y *descifrado*, porque nada se oculta (no obstante, es una nomenclatura que siguen arrastrando algunos autores).

Como ya se ha dicho, a esta operación se la conoce también (con cierta falta de propiedad) como *firma*. Por ello, es el caso que algunos autores¹, continuando con la tendencia de reflejar en la nomenclatura la **intención** con que se realiza la operación, denotan el cifrado de m con la clave privada de A como $c = S_A(m)$, del inglés *signature*. La operación inversa, mediante la cual el receptor procede a comprobar o verificar la autoría de A, se recoge, conforme a esa nomenclatura, como $m = V_A(c)$. Si una entidad B encripta un mensaje, cifrándolo con la clave pública de la entidad A, la representación sería $c = E_A(m)$ y el correspondiente descifrado, usando la clave privada de A, se denotaría como $m = D_A(c)$. Puede observarse que, con esta nomenclatura «intencional», en un algoritmo del tipo de RSA, una idéntica operación matemática, cual sería el cifrado con la clave privada de A, puede aparecer como $D_A(x)$ o bien como $S_A(x)$, es decir, con notación totalmente dispar.

Debido a que, como se ha venido insistiendo, los algoritmos de clave pública están concebidos como transformaciones de un mensaje mediante funciones matemáticas, parece más interesante reflejar en la nomenclatura, de forma escueta, la operación matemática que se está realizando, dejando aparte la intencionalidad con la que la operación se realiza. Este es el criterio que se va a seguir aquí a partir de ahora. La nomenclatura que se propone para expresar las operaciones es, con ligeras variaciones, la utilizada en la norma X.509 de ITU-T (y en su equivalente de ISO²). Para la

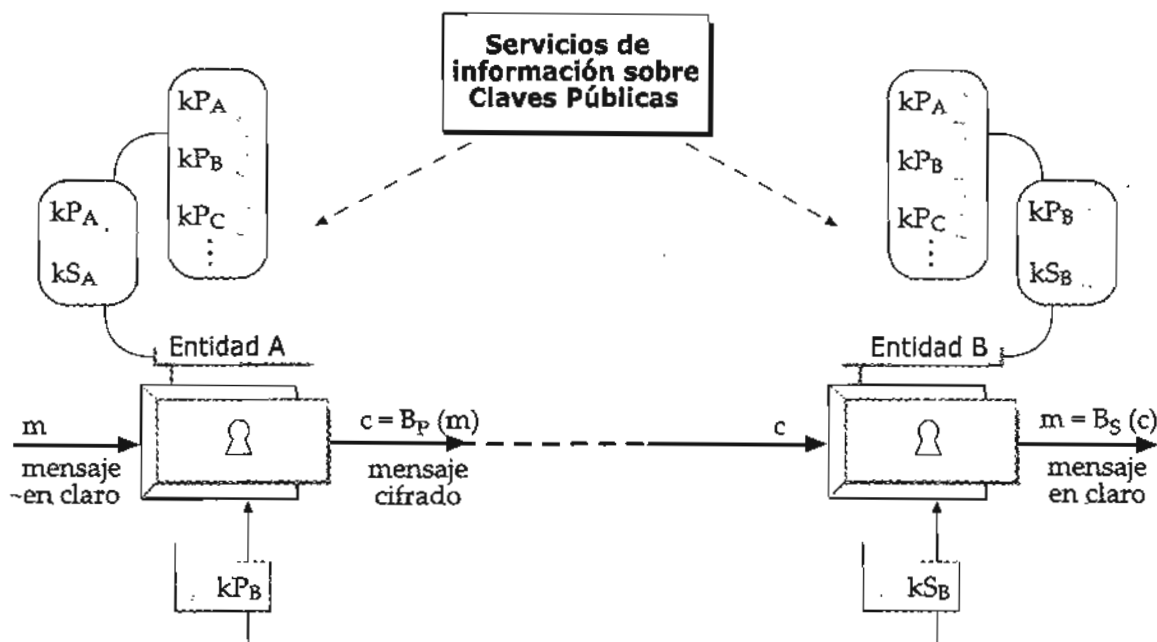


Figura 5.3. Cifrado con la clave pública de la entidad receptora.

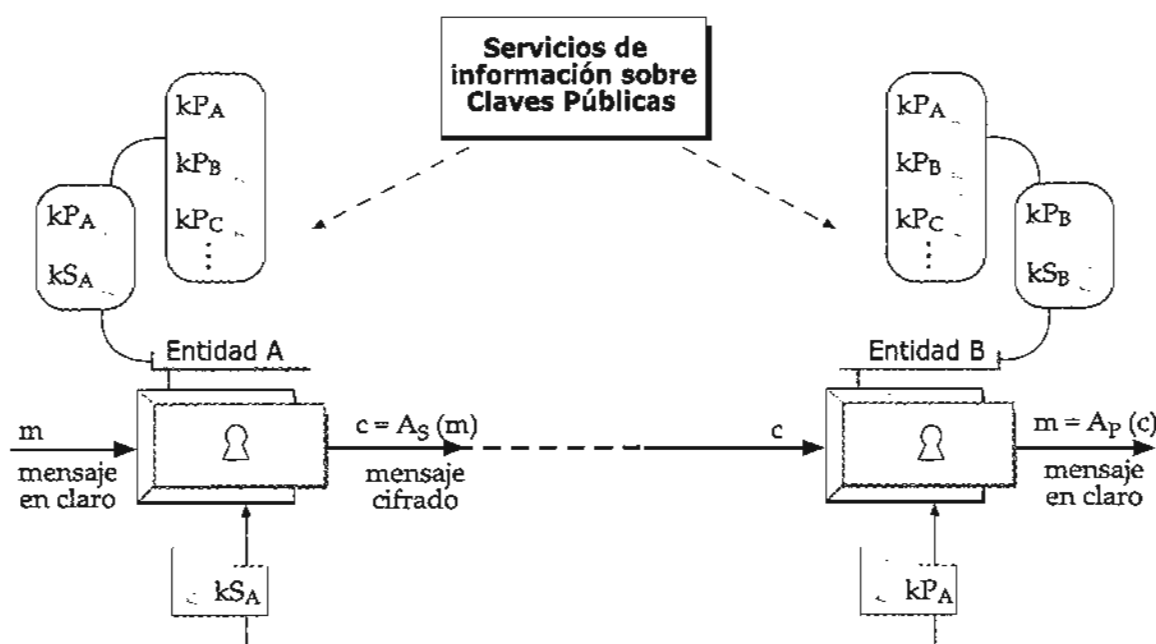


Figura 5.4. Cifrado con la clave privada de la entidad emisora.

representación de las claves públicas y privadas seguiremos con la misma nomenclatura que hemos venido utilizando hasta ahora, usando la «P» para denotar la clave pública y la «S» para la clave privada, recordando que ésta es secreta para cada entidad (para sortear la mala fortuna que hace que «privada» y «pública» empiecen por la misma letra, tanto en inglés como en español). En las Figuras 5.3 y 5.4 se recogen, respectivamente y usando la nueva nomenclatura, las operaciones de cifrado con la clave pública del receptor y la de cifrado con la clave privada del emisor representadas en las Figuras 5.1 y 5.2. En estas nuevas figuras se ha añadido la existencia de un servidor de claves y se refleja la circunstancia (muy frecuente) de que cada una de las entidades comunicantes esté en posesión de las claves públicas de todas las restantes entidades que participan en ese escenario de comunicación.

Esta nomenclatura es menos intuitiva y de más difícil interpretación que la usada anteriormente, pero, como podrá apreciarse más tarde, es más rigurosa y precisa. En concreto, la nomenclatura que se usará es la que sigue:

Una entidad genérica X posee un par de claves kP_X y kS_X , pública y privada, y tiene conocimiento, que le es proporcionado por algún servicio de información, sobre el valor de las claves públicas de todas las entidades comunicantes que participan en el mismo dominio de seguridad. El *cifrado de un mensaje q con la clave privada de la entidad X* se representa como:

$$X_S[q] \text{ o } X_S(q)$$

usando indistintamente el paréntesis y el corchete para delimitar el elemento sobre el que se aplica la operación. El único objetivo de su uso combinado es facilitar al lector la interpretación de las fórmulas. De forma similar, el *cifrado de un mensaje q con la clave pública de la entidad X* se representa como:

$$X_P[q] \text{ o } X_P(q)$$

Habitualmente, se usará el genérico «cifrar» para referirse a la operación matemática de transformación que se realice usando tanto la clave pública como la privada,

sin tener en cuenta si q es ya un mensaje cifrado, una o varias veces, ni la intencionalidad con la que se realice la operación. No obstante, de forma coyuntural, usaremos el término «descifrar» cuando la operación de cifrado que en ese momento se realice tenga, a ojos vista, tal cometido.

El carácter inverso de las operaciones de cifrado realizadas alternativamente con la clave pública y privada se refleja como:

$$X_p[X_s(q)] = q$$

$$X_s[X_p(r)] = r$$

Un elemento más de la notación que seguiremos:

Para reflejar las transferencias de información que se llevan a cabo entre las distintas entidades de la red, de ahora en adelante se utilizará la siguiente terminología:

$$A \rightarrow B: m$$

Que significa: «la entidad A envía a la entidad B el mensaje m ».

5.2. PROVISIÓN DE CONFIDENCIALIDAD, AUTENTICACIÓN E INTEGRIDAD SÓLO CON CRIPTOSISTEMAS DE CLAVE PÚBLICA

Mediante el uso exclusivo de criptografía de clave pública es posible proporcionar los tres servicios básicos de confidencialidad, integridad y autenticación del origen de los datos. Otra cosa es que este tipo de operaciones sean o no interesantes para la configuración de protocolos de seguridad, cosa que se discutirá en el siguiente epígrafe. Vamos a suponer que A es una entidad comunicante que desea enviar a otra entidad B un mensaje protegido con estos servicios de seguridad, a través de una red telemática y apoyándose en algún protocolo que permita la transferencia de datos. Para ello, consideremos los siguientes pasos:

1. La entidad emisora A cifra el mensaje con su clave privada:

$$c1 = A_s(m)$$

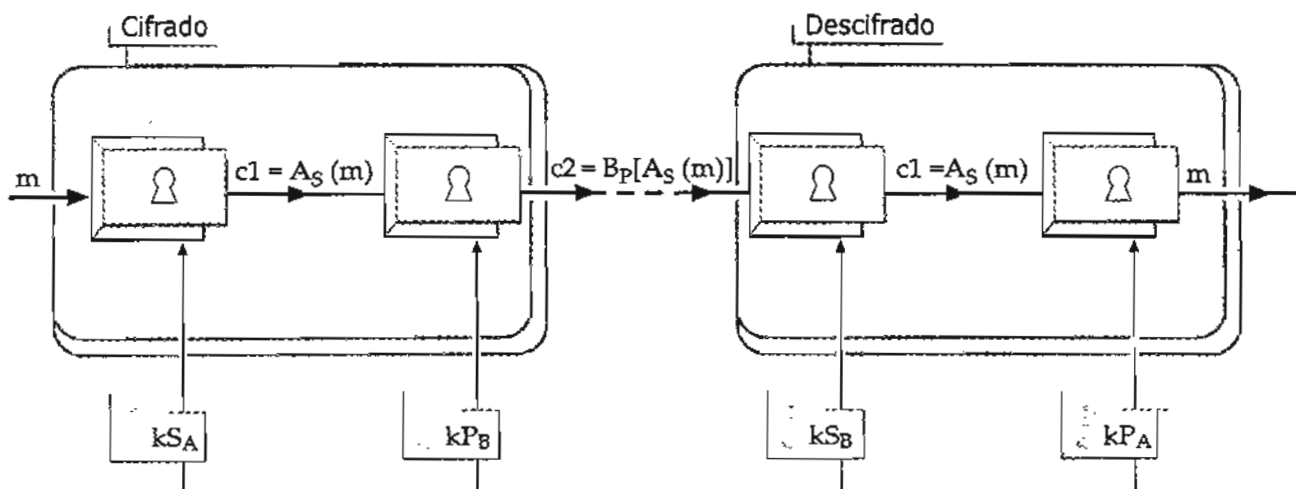


Figura 5.5. Mensaje cifrado sucesivamente con k_{SA} y k_{PB} .

2. Además, A lo cifra con la clave pública de B:

$$c2 = B_p(c1) = B_p[A_s(m)]$$

y envía c2 a B a través de la red:

$$A \rightarrow B: c2 = B_p[A_s(m)]$$

3. La entidad receptora B, tras recibir el mensaje c2, lo descifra usando su clave privada (sólo B puede hacerlo) y obtiene el mensaje c1:

$$B_s(c2) = B_s(B_p[A_s(m)]) = A_s(m) = c1$$

4. La entidad receptora B descifra c1 usando la clave pública de A:

$$A_p(c1) = A_p[A_s(m)] = m$$

Así, B obtiene m y suficiente certeza de que procede de A. También puede tener garantías de que, si su clave privada ha sido guardada y custodiada apropiadamente, B y sólo B está capacitada para conocer el contenido de m. Además, puede confiar en que el mensaje m recibido es exactamente el mismo que cifró A en el origen de la comunicación (siempre y cuando sea un mensaje entendible para B). Es decir, se han provisto los tres servicios básicos que se enumeraban más arriba.

Aunque se ha supuesto un escenario telemático para explicar el *mecanismo de seguridad* que posibilita la provisión de los tres servicios, lo que antecede no es la descripción de un *protocolo telemático de seguridad*, especificación que requeriría muchos más detalles adicionales³.

Es oportuno llamar la atención sobre el hecho de que tener certeza sobre la autenticidad del origen de los datos significa que B puede tener confianza sobre la autoría de A en el cifrado de los datos de origen, pero no en que haya sido A quien los ha enviado a través del protocolo de transferencia de datos (según se explica detalladamente en el Capítulo 9). Aunque con lamentable frecuencia podemos encontrarnos con descripciones que los confunden, este sería otro servicio, el *de no repudio de envío*, que exige otro escenario de comunicaciones⁴ y la presencia de terceras partes de confianza (TTPs).

A continuación vamos a considerar lo que ocurriría si B toma la iniciativa de enviar a A un acuse de recibo cifrado. Al escenario anterior le añadiríamos dos nuevos pasos:

5. La entidad B, tras recibir m, lo cifra primero con su clave privada y posteriormente con la clave pública de A:

$$c3 = A_p[B_s(m)]$$

y envía c3 a A:

$$B \rightarrow A: c3 = A_p[B_s(m)]$$

6. A obtiene m de forma similar a como lo obtuvo B en los pasos 3 y 4 antes descritos, comprobando que el mensaje coincide con el que A había enviado anteriormente.

$$B_p[A_s(A_p[B_s(m)])] = B_p[B_s(m)] = m$$

A modo de ejercicio, estudiaremos el posible ataque de intrusión que un usuario malicioso M podría perpetrar, dejando en agua de borrajas las protecciones establecidas. Para que funcione el ataque, vamos a suponer que B, de forma indebida, tiene automatizado el proceso en su ordenador y realiza una réplica (conforme a lo descrito en el paso 5) a todo lo que le llega sin analizar lo que recibe. Un usuario malicioso M se ha interferido en la comunicación y ha capturado el mensaje $c2 = B_p[A_s(m)]$ que A envió a B en el paso 2. M sabe que, aunque lo haya capturado, no puede leerlo porque no dispone de las claves adecuadas, pero, conocedor de la actitud compulsivamente replicativa de B, va a aprovechar esta inadecuada forma de proceder para averiguar el valor de m. Para ello, simulando que está en el paso 2, se lo envía a B, quien, conforme al procedimiento establecido, ejecuta correctamente los pasos 3 y 4. Esto es:

3'. y 4'. B trata el mensaje recibido c2, procedente de M, como si fuese un mensaje cifrado por M:

$$B_s(c2) = A_s(m)$$

$$M_p[A_s(m)] = m'$$

que es puro ruido y no significa nada.

5'. B, tras recibir m' y sin mirar que su contenido carece de significado y no debería, por tanto, seguir procesándolo, lo cifra primero con su clave privada y posteriormente con la clave pública de M y se lo envía a M:

$$c'3 = M_p[B_s(m')] = M_p[B_s(M_p[A_s(m)])]$$

$$B \rightarrow M: \quad c'3 = M_p[B_s(M_p[A_s(m)])]$$

6'. M cifra c'3 con su clave secreta:

$$c'4 = M_s(c'3) = M_s(M_p[B_s(M_p[A_s(m)])]) = B_s(M_p[A_s(m)])$$

A continuación cifra de nuevo c'4 usando la clave pública de B, consiguiendo quitar una nueva «capa» de cifrado al mensaje anterior:

$$c'5 = B_p(c'4) = B_p(B_s(M_p[A_s(m)])) = M_p[A_s(m)]$$

Prosigue con el proceso de simplificación cifrando con su propia clave privada:

$$c'6 = M_s(c'5) = M_s(M_p[A_s(m)]) = A_s(m)$$

Y, para rematar, cifra de nuevo c'6 usando la clave pública de A:

$$A_p(c'6) = A_p[A_s(m)] = m$$

obteniendo el mensaje original.

Puede apreciarse cómo el uso de esta nomenclatura «no intencional», aquí propuesta, facilita la interpretación de las operaciones y permite hacer un seguimiento sencillo de la evolución de las expresiones cuando se producen dos operaciones inversas (cifrado una misma información primero con la clave pública y después con la clave secreta, o viceversa).

5.3. MECANISMO DE FIRMA DIGITAL

El Capítulo 8 está especialmente dedicado a estudiar los escenarios telemáticos de autenticación y firma digital que tan crucial papel juegan en la configuración de los servicios de seguridad. Se estudian ahí con detenimiento los distintos algoritmos diseñados para tal fin y se resumen los criterios más relevantes de la reglamentación legal existente. No obstante, aquí, de forma resumida y a modo de adelanto, se presentan los conceptos relevantes y los algoritmos más significativos relacionados con estas técnicas.

Durante siglos y siglos, la firma caligráfica (ó signatura) ha sido un mecanismo de autenticación en el intercambio convencional de documentos escritos que ha gozado de un predicamento notable y que tiene una importancia de primer orden en todos los ámbitos sociales, tanto en las comunicaciones personales como en las relaciones económicas y jurídicas.

Por ello, la aceptación social (o no) de la firma digital como una proyección de la firma caligráfica al mundo de la Sociedad de la Información va más allá de la mera definición de un algoritmo y un mecanismo de seguridad y tiene mucho que ver con las garantías y salvaguardas que rodeen al uso de las claves y a las responsabilidades que asuman los que generan y certifican esas claves. Por ello se ha pospuesto su estudio hasta después de abordar con cierto detalle la cuestión de la gestión y certificación de claves en entornos telemáticos heterogéneos, cosa que se hará en el próximo capítulo.

Como se verá, hay dos algoritmos de firmado que gozan de una amplia aceptación tanto social como tecnológica. Éstos son la firma RSA y la firma DSA. En la Figura 5.6 se hace una representación genérica del mecanismo de firma digital que es válida, sin entrar en detalles, tanto para ambos algoritmos como para otros muchos que han sido diseñados.

La entidad A, que va a emitir la firma o signatura del documento m , recurre a un Proceso de Firma, PF, que genera la firma f :

$$f = A_{\text{sig}}(m)$$

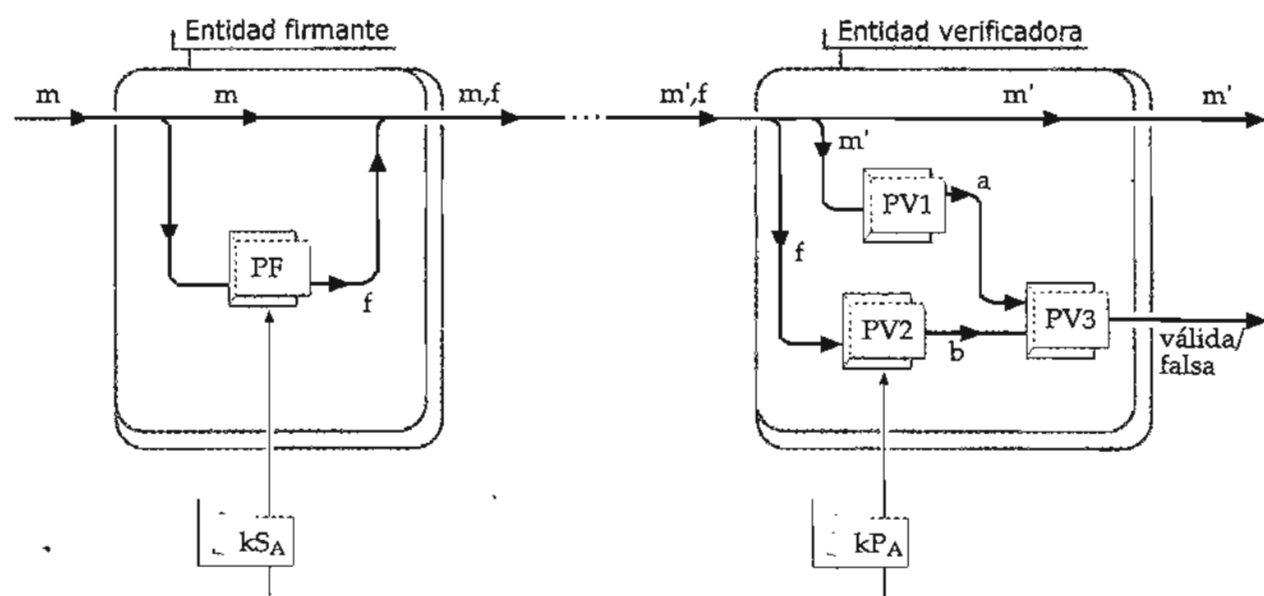


Figura 5.6. Mecanismo de firma digital.

De forma que se cumple que:

1. f es una pieza de información de tamaño reducido, fijo e independiente del tamaño de m .
2. La generación de f se realiza mediante un proceso de cifrado a partir del mensaje a firmar, m , y de la clave privada del firmante que la emite.
3. Es relativamente fácil obtener f a partir de m y de kS_A .
4. Es computacionalmente impracticable obtener f a partir de m sin conocer kS_A .

Cuando emite la firma, la entidad A envía m y f :

$$A \rightarrow B: \quad m, f$$

La entidad comunicante B, al igual que cualquier otra entidad que conozca la clave pública de A, puede verificar la validez de la firma mediante un proceso de validación (Figura 5.6) dividido en tres pasos relacionados entre sí:

1. La entidad B recibe conjuntamente m' y f . Partiendo de m' y mediante el paso 1 del proceso de verificación, PV1, obtiene a , que es una pieza de información de tamaño reducido, fijo, e independiente de m . El mensaje recibido se ha denotado por m' , ya que, aunque se espera que sea el mismo m que envió A, existe la duda de que haya sido alterado durante la transmisión, de forma fortuita o malintencionada.
2. Partiendo de f y mediante el paso 2 del proceso de verificación, PV2, obtiene b , que es una pieza de información de tamaño fijo y reducido. Esta operación se realiza mediante un proceso de cifrado a partir de f y de la clave pública de la entidad firmante.
Es relativamente fácil obtener a partiendo de m' .
Es relativamente fácil obtener b a partir de f y de kP_A .
3. Mediante el paso 3 del proceso de verificación, PV3, se establece una comparación entre los distintos campos de las piezas de información a y b , como resultado de la cual se conoce si la firma es válida, esto es, se corresponde con el documento m' recibido, o si, por el contrario, es falsa.

Teniendo en cuenta que A, y sólo A, debe conocer kS_A , el mecanismo de firma digital sirve para proporcionar los servicios de *autenticación del origen de los datos* y el de *integridad* del mensaje recibido.

En cierta forma, cuando la entidad A cifra un mensaje m completo con su clave privada, lo que hemos venido denotando como $A_s(m)$ (y que en algunos tratados de criptografía suele denominarse como «firma») puede ser considerado como un caso particular del mecanismo de firma digital, con la excepción de que en este caso el tamaño de la firma no es fijo, sino de una longitud similar a la del mensaje cifrado. Además, en este caso no es imprescindible enviar también m . Ya se comentó antes, al hablar de las características principales de la criptografía de clave pública, lo poco adecuado de esta solución cuando de lo que se trata es de firmar documentos relativamente grandes.

De forma resumida, veremos a continuación el esquema tanto de la firma RSA como de la firma DSA. Para ello, previamente, presentaremos el algoritmo de resumen o hash utilizado en ambos modelos de firmado.

RESUMEN DE M (VALOR HASH)

El **resumen** o **valor hash** del mensaje m es un elemento de información de tamaño fijo generado mediante una función unidireccional que también puede ser utilizado como autenticador de mensajes. No obstante, el verdadero interés de las funciones hash radica en que son parte fundamental en la estructura de los algoritmos de firmado.

Por tratarse de funciones unidireccionales (*one-way functions*), el cálculo de la función directa es computacionalmente «fácil» de realizar, pero la operación inversa es computacionalmente impracticable. Dado m (de longitud arbitrariamente larga) y H , función resumen o función hash, se obtiene:

$$h = H(m)$$

que es de longitud fija. Además, es impracticable el cálculo de m mediante $H^{-1}(h)$.

ALGORITMO DE FIRMA RSA

En la Figura 5.7 se muestra el esquema de funcionamiento seguido para la obtención de la firma conforme el algoritmo RSA y su posterior verificación. Obviamente, es una particularización del esquema mostrado en la Figura 5.6 que representa de forma genérica el mecanismo de firma digital.

La entidad A desea enviar a B el mensaje m firmado. Para ello, realiza las siguientes operaciones:

1. Realiza $h = H(m)$.
2. Cifra $H(m)$ con su clave privada para conseguir la **firma** f .

$$f = A_S[H(m)] = A_S[h]$$

3. Envía a B m y f .

$$A \rightarrow B: \quad m, f$$

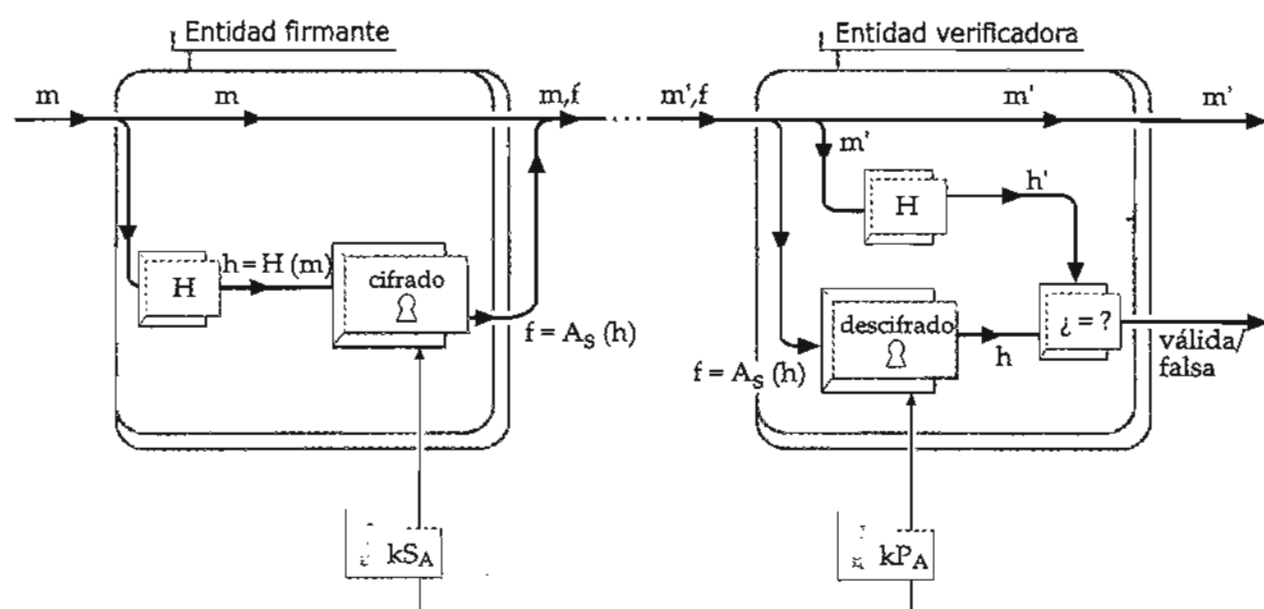


Figura 5.7. Esquema de la firma digital RSA.

La entidad B recibe conjuntamente m' y f (que es una pieza de información ininteligible para ella). Para efectuar la **verificación**, la entidad B realiza las siguientes operaciones:

1. Descifra f con la clave pública de A:

$$A_p(f) = A_p(A_s[h]) = h = H(m)$$

2. Realiza $H(m') = h'$.

3. Comprueba que $H(m') = H(m)$. Si esto es cierto, da por válida la firma y:

- Sabe que A es la entidad signataria del mensaje.
- Sabe que el mensaje m' recibido es exactamente el mismo mensaje que A firmó.

ALGORITMO DE FIRMA DSA

El algoritmo DSA es mucho más difícil de entender que el RSA, razón por la cual pospondremos al Capítulo 8 una explicación más pormenorizada de su estructura y la justificación de su proceso de verificación. De forma resumida, consiste en hacer públicos una serie de parámetros y, naturalmente, la clave pública de la entidad firmante. La entidad verificadora conocerá, pues, estos datos.

Los **datos públicos** para todo el dominio son:

- **p** Número entero con un número de dígitos binarios múltiplo de 64 y comprendido entre 512 y 1.024.
- **q** Número primo de 160 bits divisor de p.
- **g** Número entero relacionado con p que cumple con una determinada condición (véase Capítulo 8).

Para la **firma** del mensaje m , el procedimiento seguido por la entidad A signataria (Figura 5.8) es el siguiente:

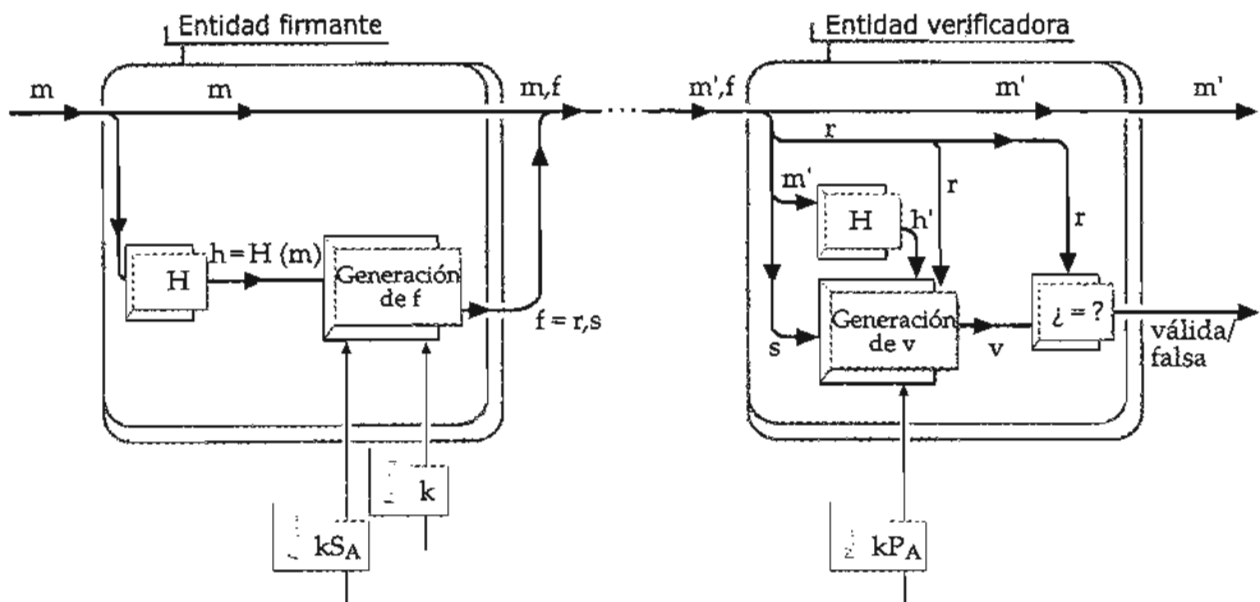


Figura 5.8. Esquema de la firma digital DSA.

1. Realiza $h = H(m)$.
2. A partir de h , de los datos públicos p , q y g , de la clave privada kS_A y de un número aleatorio k (mayor que cero y menor que q) obtiene la **firma** f :

$$f = r, s$$

3. Envía a B m y f .

$$A \rightarrow B: \quad m, r, s$$

La entidad B recibe conjuntamente m' y f (que es una pieza de información ininteligible para ella). Para efectuar la **verificación**, la entidad B realiza las siguientes operaciones:

1. Realiza $H(m') = h'$.
2. A partir de h' , de los datos públicos p , q y g , de la firma $f = r, s$ y de la clave pública de la entidad firmante obtiene el parámetro v conforme a una compleja expresión.
3. Se puede demostrar que si $h' = h$, entonces $v = r$. Por ello, B comprueba esta igualdad y si se produce:
 - Sabe que A es la entidad signataria del mensaje.
 - Sabe que el mensaje m' recibido es exactamente el mismo mensaje que A firmó.

5.4. USO COMBINADO DE LA CRIPTOGRAFÍA SIMÉTRICA Y ASIMÉTRICA

De lo que hemos venido describiendo, tanto en los párrafos precedentes como en el capítulo anterior, podemos sacar algunas conclusiones comparativas entre los criptosistemas de clave secreta y los de clave pública.

1. Los criptosistemas simétricos son muchísimo más rápidos que los asimétricos (del orden de 1.000 veces, como promedio), lo que los hace imprescindibles para proporcionar confidencialidad en aplicaciones telemáticas de media o alta velocidad.
2. Para conseguir confidencialidad usando solamente criptosistemas simétricos en un entorno en el que participen un número considerable de entidades es necesario disponer de un número tan elevado de claves secretas que hace inaplicable este esquema de comportamiento. En este caso, si lo que se pretende es que cada par de usuarios comparta una clave secreta, el número total de claves que sería necesario gestionar en un dominio abierto de seguridad (como el que puede darse para establecer, por ejemplo, un sistema de correo electrónico seguro) sería totalmente inmanejable.
3. Los criptosistemas simétricos son más vulnerables que los asimétricos ante ataques por fuerza bruta, lo que hace necesaria una renovación periódica de las claves.
4. El hecho de que en los criptosistemas asimétricos solamente las claves privadas tengan que mantenerse secretas permite una muy adecuada gestión de claves en entornos heterogéneos, donde las distintas entidades comunicantes sólo necesitan confiar en la validez de la clave pública de las restantes.

Para aprovechar las ventajas que cada uno de los criptosistemas ofrece, se postula el uso combinado de ambas técnicas, de tal forma que en primera instancia se utilicen las técnicas de criptografía de clave pública para generar una clave simétrica y, posteriormente, se use esa clave simétrica para encriptar el mensaje.

Para materializar este procedimiento se pueden distinguir, básicamente, dos esquemas de comportamiento:

- a) Generación de una clave simétrica de sesión mediante el mecanismo de *envoltura digital*, que será descrita en el siguiente apartado.
- b) Generación de la clave simétrica mediante acuerdo entre las partes basada en el algoritmo de concertación de claves de Diffie-Hellman, que se verá en los epígrafes 5.7 y 5.9.

CIFRADO CON CLAVE DE SESIÓN GENERADA MEDIANTE EL MECANISMO DE ENVOLTURA DIGITAL

Se puede conseguir la provisión de los servicios de integridad y confidencialidad sin necesidad de recurrir a varios intercambios de información entre las entidades. Este es el caso del mecanismo denominado *envoltorio digital* o *envoltura digital* (del inglés «*digital envelop*») que se representa gráficamente en las Figuras 5.9 y 5.10. Básicamente consiste en lo siguiente:

1. La entidad A genera de forma aleatoria una clave de sesión k .
2. La entidad A cifra el mensaje m mediante un criptosistema simétrico usando la clave k :

$$c1 = E_k(m)$$

3. La entidad A cifra la clave k con la clave pública de B, usando un criptosistema asimétrico:

$$c2 = B_p(k)$$

4. A envía a B ambos criptogramas $c1$ y $c2$:

$$A \rightarrow B: \quad c1, c2$$

5. La entidad B obtiene la clave de sesión cifrando $c1$ con su clave privada:

$$k = B_s(c2)$$

6. B descifra el mensaje m usando el mismo criptosistema simétrico y la clave k obtenida:

$$m = D_k(c1)$$

Como fácilmente se deduce, este mecanismo sirve para proporcionar los servicios de *integridad* y *confidencialidad* de los datos.

La clave de sesión generada es «de usar y tirar», por lo que los ataques contra el criptosistema simétrico sólo tendrían efectividad para la averiguación del mensaje m concreto que se ha transferido en esa operación, resultando necesario un nuevo gasto de energía en un nuevo ataque cada vez que se transfiera un nuevo mensaje. Se consigue así un eficazísimo procedimiento de renovación de la clave secreta, que es un requisito de seguridad al que ya hemos hecho reiteradas referencias.

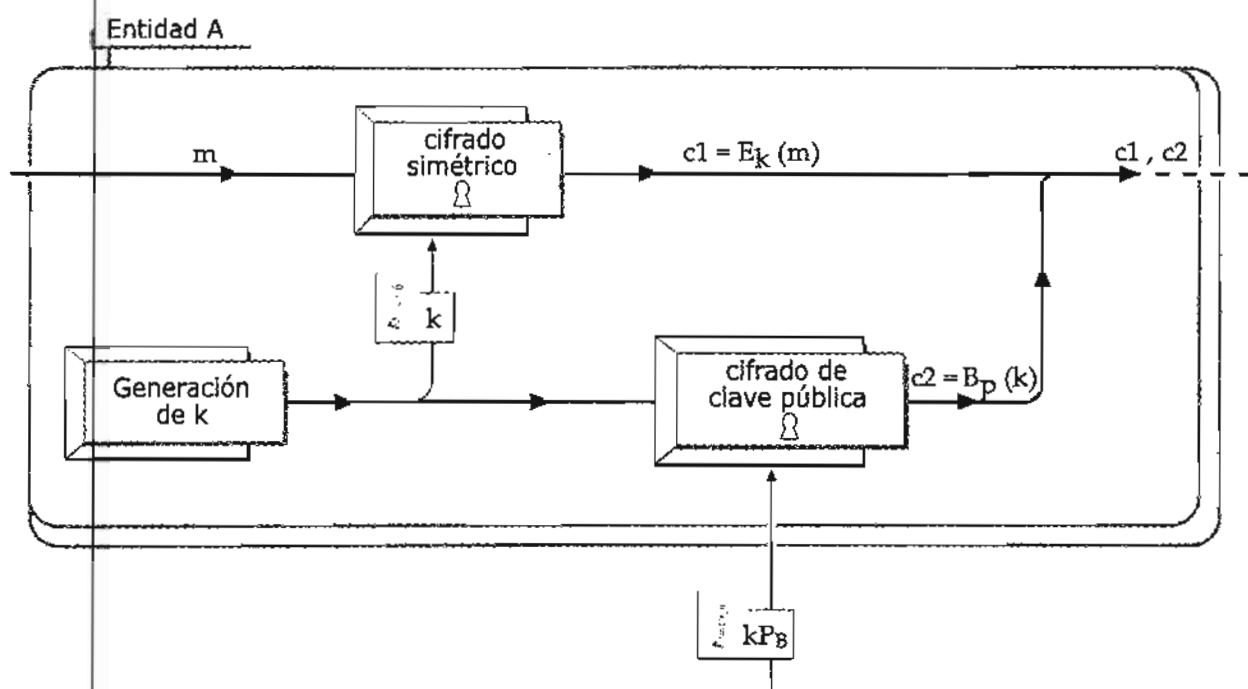


Figura 5.9. Mecanismo de envoltura digital: extremo emisor.

Al utilizar un criptosistema de clave secreta para cifrar el mensaje se pueden obtener tiempos de operación muy ventajosos, que redundarán en un buen funcionamiento del protocolo telemático del que sean parte constituyente. Asimismo, al ser k una pieza de información relativamente pequeña, el proceso de cifrado mediante un criptosistema de clave pública es sencillo y rápido.

Este mecanismo de seguridad, con algunas variantes, está presente en multitud de protocolos telemáticos de seguridad. Estas variantes presentan ligeras diferencias entre sí en cuanto al orden de operación, en cuanto a la información que viaja en claro y en relación con algunas piezas de información añadidas como pueden ser sellos de

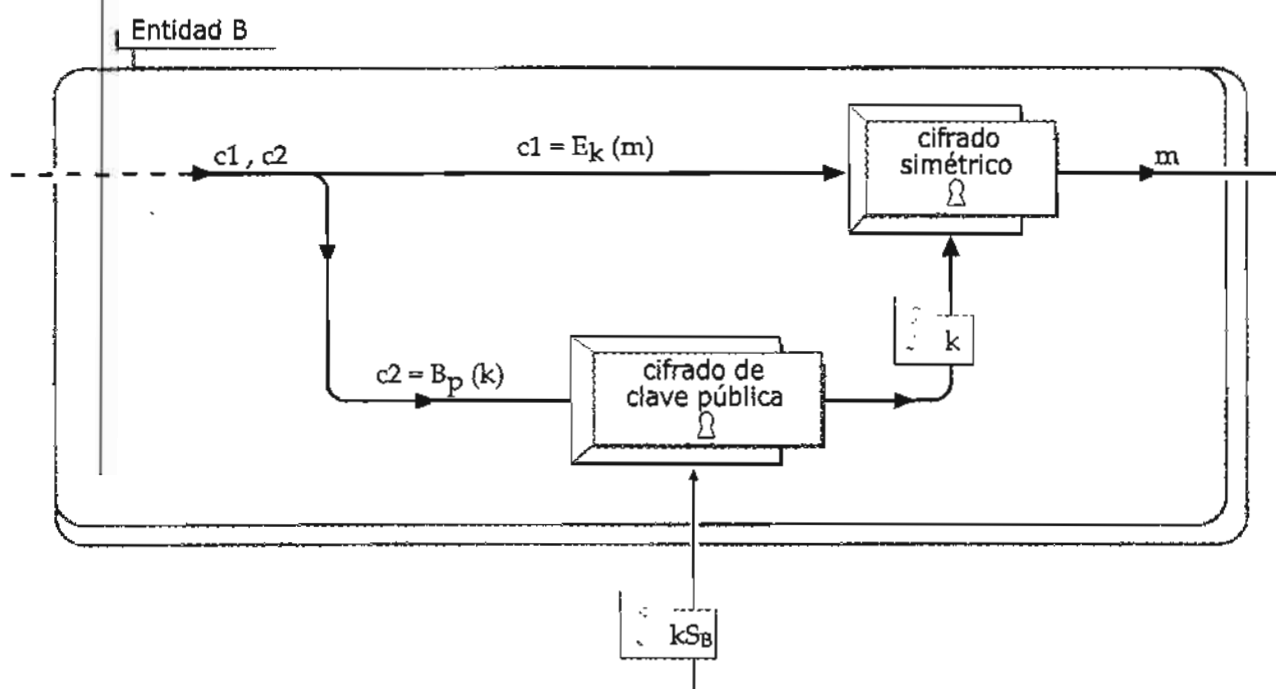


Figura 5.10. Mecanismo de envoltura digital: extremo receptor.

tiempo, certificados y firmas digitales. Si la entidad A incorporase su firma digital del mensaje cifrado, se conseguiría añadir el servicio de *autenticación del origen de los datos* y se reforzaría el de *integridad*.

5.5. OTROS MECANISMOS CRIPTOGRÁFICOS

En los epígrafes precedentes se ha procurado hacer un resumen de los mecanismos criptográficos más genéricos de cara a la provisión de servicios de seguridad. Como ya se ha dicho, los mecanismos de seguridad son las piezas con las que se compone un protocolo de seguridad, dándose la circunstancia de que la mayoría de ellos se obtienen por medio de técnicas criptográficas, por lo que *mecanismo de seguridad* es casi sinónimo de *mecanismo criptográfico*.

Debido a ello, cuando se diseñan protocolos telemáticos de seguridad, van apareciendo nuevos mecanismos criptográficos que contribuyen a conseguir la funcionalidad pretendida. En los capítulos siguientes, a medida que vayamos presentando estos protocolos específicos, iremos describiendo nuevos mecanismos auxiliares.

En el caso particular de los protocolos que proporcionan el Servicio de Anonimato, son necesarios algunos *mecanismos criptográficos avanzados*, calificados así no porque las técnicas y algoritmos criptográficos que se usen en ellos sean más complejas que las que se presentan en este capítulo, sino porque no se requiere de su empleo en los servicios de seguridad convencionales. A modo de reseña, podemos adelantar una breve descripción de algunos de ellos:

- *Firma opaca*. También llamada *firma a ciegas* (*blind signature*), se caracteriza porque el firmante no «ve» lo que firma. La confianza en que lo que está firmando es algo adecuado (que no perjudica su seguridad ni sus derechos) tiene que adquirirla por otros medios complementarios presentes en los protocolos donde se aplica.
- *Secreto dividido*. Se establece cuando existe una información muy sensible y se desea que sea compartida por varias personas o entidades de forma que solamente reuniéndose todas ellas pueda ser reconstruido el secreto, pero sin que ninguno de los participantes averigüe la información secreta que poseen los restantes socios. (Lo mismo que ocurre cuando tres directivos de un banco tiene cada uno una llave de la caja fuerte, de forma que ésta sólo puede abrirse cuando se juntan los tres.)
- *Secreto compartido*. Es una generalización del caso anterior. Igualmente, se reparten trozos de información (*sombras*) entre los participantes, pero no es necesario que se reúnan todos para recomponer el secreto: para conseguirlo basta con que se junte una «mayoría cualificada» de ellos. Se puede determinar cuál es esa mayoría y dar «pesos» distintivos según la jerarquía de cada uno de los poseedores de la *sombra*.

5.6. CRIPTOSISTEMA RSA

Como ya se ha dicho, la primera propuesta conocida acerca de los criptosistemas de clave pública es la de Diffie y Hellman en 1976. Han aparecido algunas noticias que atribuyen la primera propuesta de criptosistemas de clave pública a diversos autores que las habrían descrito en documentos más o menos secretos. Aparte de arqueologías

de dudosa comprobación, aquí nos guiamos por las referencias hechas públicas ante el mundo científico y académico, que son las que realmente han tenido repercusión en la evolución de esta tecnología. Según lo descrito por el propio Diffie, el descubrimiento fue realizado durante el curso 1974-75 por Whitfield Diffie y Martin Hellman, de la Universidad de Stanford, y Ralf Merkle, de la Universidad de Berkeley, aunque la primera publicación es la ya citada [DiHe76]. En este trabajo, ellos sólo proponían una solución parcial al problema de creación de un algún algoritmo concreto capaz de soportar ese esquema de funcionamiento.

La primera propuesta de un algoritmo que soporte dicho modelo fue realizada por Rivest, Shamir y Adleman [RSA78], tres jóvenes profesores del MIT, y se conoce bajo el nombre de Criptosistema RSA, conforme a las iniciales de sus nombres. Desde su aparición ha sido el sistema que se ha aceptado en todos los ámbitos donde se opera con criptografía de clave pública y, de forma especial, en los trabajos conducentes al diseño de los mecanismos de seguridad necesarios para configurar los protocolos telemáticos seguros de propósito general.

Su seguridad se basa en la dificultad de factorizar grandes números y en el hecho de que el cálculo de logaritmos en aritmética modular es un problema NP-completo inabordable computacionalmente para bases de valor elevado. El mensaje a cifrar se divide en bloques de tal manera que tanto el mensaje a cifrar como el mensaje cifrado son números enteros comprendidos entre cero y $n-1$. Las operaciones aritméticas se realizan módulo n conforme a las reglas que se resumieron en el epígrafe 2 del Capítulo 3. Por tanto, el número de bits de cada bloque debe ser menor o igual al $\log_2 n$. El procedimiento a seguir es el siguiente:

- Cada usuario o entidad comunicante elige dos números primos p y q muy grandes (más de 100 dígitos decimales para que ofrezca seguridad en la situación actual de la tecnología) y se calcula:

$$n = p \cdot q \quad (1)$$

Para más seguridad, p y q deben ser de igual o similar longitud. El conjunto en el que se va a trabajar a partir de ahora será el *conjunto reducido de restos* módulo n , $\mathbb{Z}_n$.

- Por ser primos, el indicador de Euler (ecuación 10, Capítulo 3) es fácil de calcular:

$$\phi(n) = (p - 1) \cdot (q - 1) \quad (2)$$

- Para formar parte de su clave pública, el usuario o entidad A elige e , con la condición de que $e \in \mathbb{Z}_n$ y que sea primo relativo con $\phi(n)$:

$$\text{mcd}(e, \phi(n)) = 1 \text{ mod } n$$

- La clave pública de A estará constituida por n y e :

$$kP_A = n, e$$

p , q y $\phi(n)$ deben permanecer secretos.

- Para formar parte de su clave privada, A calcula d de tal manera que sea menor que $\phi(n)$ y sea su inverso en el conjunto $\phi(n)$. Es decir:

$$d = e^{-1} \text{ mod } \phi(n) \quad (3)$$

o bien:

$$e \cdot d = k \cdot \phi(n) + 1 \quad (4)$$

donde k es un número entero.

Debido a que e y $\phi(n)$ son primos entre sí, ello implica (teorema de Euler, ecuación 15 del Capítulo 3) que existe un único elemento inverso de e módulo $\phi(n)$.

- La clave privada de A está constituida por n y d

$$kS_A = n, d$$

- Para conocer d a partir de e es necesario ser capaz de factorizar n .

CIFRADO CON LA CLAVE PÚBLICA DEL RECEPTOR

- La entidad B desea enviar a la entidad A un mensaje m cifrado de tal forma que A y sólo A esté capacitada para interpretarlo. Dicho mensaje m a cifrar debe estar representado por un número tal que $m \in \mathbb{Z}_n$, es decir, que sea menor que n .
- B calcula el mensaje cifrado c mediante la expresión:

$$c = m^e \bmod n \quad (5)$$

- Para obtener el mensaje en claro a partir de c se puede seguir uno de los dos procedimientos siguientes:

- a) Calculando, si fuese posible de plantear, la expresión:

$$m = \lg_e c \quad (6)$$

Esta operación podría ser abordada por cualquier entidad distinta de A pero, como ya se ha dicho, es un problema computacionalmente impracticable.

- b) Utilizando la clave privada de A, que sólo A debe conocer. Se puede recuperar m mediante la operación:

$$m = c^d \bmod n \quad (7)$$

Demostraremos que m se obtiene mediante la expresión (7):

$$c^d \bmod n = (m^e)^d \bmod n = m^{ed} \bmod n$$

En efecto, sustituyendo en la expresión anterior el valor de $e \cdot d$ recogido en (4) se tiene:

$$c^d \bmod n = m^{k \cdot \phi(n) + 1} \bmod n = m(m^{\phi(n)} \bmod n)^k$$

en virtud del teorema de Euler (ecuación 14, Capítulo 3):

$$m^{\phi(n)} \bmod n = 1$$

por lo que la expresión anterior resulta ser:

$$c^d \bmod n = m \cdot 1^k = m \quad \text{como queríamos demostrar.}$$

O lo que es lo mismo:

$$(m^e)^d \bmod n = m \quad (8)$$

CIFRADO CON LA CLAVE PRIVADA DEL EMISOR

El hecho de que e y d sean inversos respecto de $\phi(n)$ implica que si se cifra un mensaje m' usando la clave secreta de A, se obtiene:

$$c' = (m')^d \bmod n$$

Cualquier otra entidad conocedora de la clave pública de A puede recuperar el mensaje original mediante la expresión:

$$m' = (c')^e \bmod n$$

Dicha entidad obtiene garantías de que m' sólo ha podido ser cifrado por A, lo que puede servir para la provisión de mecanismos de autenticación e integridad de los datos.

Generación de claves de A

Elegir p y q primos y calcular $n = p \cdot q$ y $\phi(n) = (p-1) \cdot (q-1)$
 Elegir e | $\text{mcd}(e, \phi(n)) = 1$
 Calcular d | $d = e^{-1} \bmod \phi(n)$
 $KS_A = n, d$
 $KP_A = n, e$

Cifrado de m con la clave pública del receptor

$m < n$
 Cifrado (Entidad B): $c = m^e \bmod n$
 Descifrado (Entidad A): $m = c^d \bmod n$

Cifrado de m' con la clave privada del emisor

$m' < n$
 Cifrado (Entidad A): $c' = (m')^d \bmod n$
 Descifrado (Entidad B): $m' = (c')^e \bmod n$

Figura 5.11. Cuadro-resumen del algoritmo RSA.

EJEMPLOS DE CIFRADO USANDO RSA

Ejemplo 1

Veamos, en primer lugar, un ejemplo muy sencillo usando números muy pequeños para que sea fácil comprobar la validez de los cálculos realizados. Una entidad A elige:

$$p = 5 \text{ y } q = 11$$

Por tanto, $n = 55$

Al ser p y q números primos, según la expresión 10 del Capítulo 3, se tiene:

$$\phi(n) = (p-1)(q-1) = 40$$

A elige para su clave pública: $e = 3$, que es primo relativo con 40. Para calcular d , el inverso de e , puede usar el algoritmo de Euclides extendido o aplicar la fórmula de Euler, teniendo en cuenta que, según la fórmula 10 del Capítulo 3:

$$\phi(40) = \phi(2^3 \cdot 5) = (2^2 \cdot 1) (5^0 \cdot 4) = 16$$

con lo que:

$$d = 3^{\phi(40)-1} \bmod 40 = 3^{15} = 27$$

Si el mensaje que desea enviar B se representa por $m = 18$, el valor del mensaje cifrado sería:

$$c = 18^3 \bmod 55 = 2$$

Al recibir el mensaje c , A puede calcular m elevando a 27, módulo 55, el valor 2. Este cálculo se puede realizar de forma sencilla utilizando una técnica similar a la utilizada en los ejemplos de exponenciación presentados en el Capítulo 3. El resultado sería:

$$m = 2^{27} \bmod 55 = 18 \quad \text{que es el valor original asignado a } m.$$

Con números tan pequeños, resulta muy fácil romper la seguridad del cifrado. En efecto, conociendo el $n = 55$, es casi trivial obtener su descomposición en factores primos, utilizando un procedimiento de tanteo y comenzando por los números primos menores hasta llegar al 5.

Ejemplo 2

Partiendo de los mismos valores, $p = 5$, $q = 11$, $n = 55$, $\phi(55) = 40$ y $\phi(40) = 16$, vamos a considerar un ejemplo, en el que el mensaje en claro que desea enviar B sea el texto «tañer». Vamos a suponer, además, que el valor elegido por A para su clave pública es $e = 7$, de donde podemos deducir el valor del componente de su clave privada: $d = 23$.

Para codificar numéricamente el mensaje partimos del alfabeto español de 27 letras. A la letra «a» se le asigna el valor 0, y a la «z» el valor 26. Los valores relativos asignables a cada una de las letras del mensaje «tañer» serían, respectivamente, 20, 0, 14, 4 y 18. Para determinar el tamaño del bloque utilizado para cifrar con RSA hay que tener en cuenta que si éste estuviese constituido por dos caracteres, el valor de m para los dos primeros de la palabra «tañer» sería:

$$m = \text{«ta»} = 20 \cdot 27 + 0 = 540$$

que, como puede comprobarse, es muchísimo mayor que 55, valor de n . Por ello no tendremos más remedio que considerar cada bloque constituido por una sola letra.

Para la letra «t», se tendría:

$$c = 20^7 \bmod 55 = 15$$

Para realizar este cálculo puede hacerse uso de un algoritmo de exponenciación rápida propuesto por Knuth [KnDo69]. Para esa misma letra, cuando A reciba el primer bloque del mensaje cifrado, el valor del primer bloque del mensaje en claro sería:

$$m = 15^{23} \bmod 55 = 20 \quad \text{que es el valor asociado a la letra «t»}.$$

Con las restantes letras del mensaje se procedería de forma similar.

Ejemplo 3

Veamos a continuación otro ejemplo, tomado de [FGHM00], en el que el valor de n es considerablemente mayor que en los casos anteriores y en el que también se trata de transmitir un mensaje constituido por un texto. En este caso supondremos que el alfabeto que se va a utilizar es el inglés, constituido por solo 26 caracteres, de la «a» a la «z», numerados de 0 a 25. Para la configuración de su par de claves la entidad A ha elegido los números primos:

$$\begin{array}{ll}
 p = 281 \text{ y } q = 167 & , \text{ lo que arroja para } n \text{ un valor de:} \\
 n = 281 \cdot 167 = 46.927 & , \text{ siendo } \phi(n): \\
 \phi(46.927) = 280 \cdot 166 = 46.480 & , \text{ eligiendo a continuación el valor de } e: \\
 e = 39.423 & , \text{ que es primo relativo con } \phi(n). \text{ Es decir:} \\
 \text{mcd}(39.423, 46.480) = 1 & , \text{ calculando a continuación } d, \text{ inverso de } e \text{ res-} \\
 & \text{pecto de } \phi(n): \\
 d = 26.767 & , \text{ que mantiene en secreto, haciendo pública:} \\
 kP_A = n, e = 46.927, 39.423
 \end{array}$$

Aunque estos valores son considerablemente mayores que los vistos en los ejemplos anteriores, el número máximo de letras que puede tener el bloque que se codifique no será excesivamente grande y se calculará comparando con n las distintas potencias de 26. Como el valor de n está comprendido entre $26^3 = 17.576$ y $26^4 = 456.976$, el tamaño máximo del bloque será de tres letras, de forma que si el mensaje es más largo, será necesario dividirlo en grupos de tres caracteres. En concreto, si el mensaje que desea enviar la entidad B es «yes», teniendo en cuenta que los valores numéricos asignados a estas letras son, respectivamente, 24, 4 y 18, se tiene:

$$m = 24 \cdot 26^2 + 4 \cdot 26 + 18 = 16.346$$

que, obviamente, es inferior a n . El máximo valor numérico de un mensaje de tres letras correspondería a «zzz» y sería $26^3 - 1$. Algunos mensajes de cuatro letras podrían tener valores menores que n , pero aquellos en los que aparezcan letras de la parte final del alfabeto rebasarían ese valor máximo prefijado. En las implementaciones reales de RSA se manejarán claves de entre 500 y 2.000 bits, con lo que el tamaño de los bloques codificables es muchísimo mayor que el que se refleja en estos ejercicios. Volviendo al ejemplo, el mensaje cifrado que enviaría B es:

$$c = m^e \bmod n = 16.346^{39.423} \bmod 46.927 = 21.166$$

que se corresponde con el mensaje «bfic», que sería el que realmente se enviase a la entidad A.

En el extremo receptor, A deberá recuperar el valor numérico del mensaje recibido y proceder a aplicar la operación de exponenciación con su clave privada. Esto es:

$$m = c^d \bmod n = 21.166^{26.767} \bmod 46.927 = 16.346$$

que, una vez descodificado, dará lugar al mensaje original $m = \text{«yes»}$.

GENERACIÓN DE CLAVES Y FORTALEZA DEL RSA

Los ataques por fuerza bruta resultan tanto más exitosos cuanto menor es el espacio de claves posibles: es decir, cuanto menor sea el tamaño en bits de las claves utilizadas. En el caso de RSA, las claves más débiles que se usan son de 512 bits, siendo cada vez más frecuentemente utilizadas claves del doble o el cuádruple de esa cantidad. Y lo que es más significativo: el tamaño de las claves puede irse aumentando a medida que aumente la potencia y velocidad de los computadores, con la ventaja de que el aumento del tamaño afecta menos a la duración de los procesos de cifrado y descifrado que a las dificultades que este aumento representa para la ruptura del algoritmo. Por todo ello, la vulnerabilidad de RSA ante este tipo de ataques es despreciable y raramente se tiene en consideración. Un segundo grupo de ataques son los considerados bajo el epígrafe genérico de «ataques con texto cifrado elegido». Si una determinada entidad cifra con su clave privada todo lo que se le envíe, sin analizar la oportunidad de realizar tal cifrado ni el contenido de lo que cifra, pueden buscarse combinaciones de números que permitan descubrir el contenido de un mensaje ajeno interceptado, o bien averiguar el valor de la clave con que ha sido cifrado. En todos estos casos se trata de problemas de procedimiento o «modus operandi» que son fácilmente eludibles teniendo conocimiento de ellos. Además, para que estos ataques puedan funcionar es preciso imaginar supuestas TTPs que «firman» documentos usando solamente el cifrado con su clave secreta, cosa que se aleja de la realidad, en la que los mecanismos de firma digital implantados introducen elementos adicionales suficientes como para dar al traste con el ataque. Por tanto, se trata más de meros ejercicios académicos que de verdaderas debilidades del criptosistema.

Otro procedimiento de ataque utilizado con relativo éxito [Koch96] consiste en analizar el tiempo de descifrado de un criptograma: se trata por tanto de un ataque basado sólo en el mensaje cifrado. Se ha comparado con la información que se adquiere si se observa, durante la apertura de una caja fuerte, el tiempo que alguien tarda en mover la rueda de selección de la clave cuando se pasa de un número a otro. Como en el caso anterior, este tipo de ataques no es privativo de RSA, sino que puede afectar a todos los criptosistemas de clave pública. Si se conoce el algoritmo iterativo mediante el cual se realiza el descifrado y se pueden medir reiteradamente los tiempos de ejecución, se acaba teniendo una información probabilística creciente acerca del valor de cada uno de los bits intervinientes, lo cual permite la ruptura de la seguridad del sistema. Las contramedidas que se pueden adoptar para conjurar esta amenaza son relativamente sencillas y se basan en introducir elementos que hagan que todas las funciones de exponenciación duren aproximadamente lo mismo o que su tiempo de ejecución sea independiente del valor del exponente.

El ataque realmente **contundente** que se puede realizar sobre RSA es conseguir resolver en un tiempo razonable los problemas matemáticos en los que se basa. Si se resolviese el problema de cálculo de algoritmos en aritmética modular la ruptura sería inmediata. Otro tanto ocurre con el problema de la factorización: si se consigue **factorizar** n , se consiguen p y q , con lo que puede obtenerse d . Para evitar que ello sea factible, los inventores del algoritmo recomiendan que en la elección de estos números se tengan en cuenta una serie de restricciones. Es relativamente sencillo probar la necesidad de estas restricciones aunque, por razones de oportunidad, obviemos aquí su demostración. Son las siguientes:

1. p y q deben elegirse aleatoriamente de forma que difieran en solo unos pocos dígitos, pero al mismo tiempo, no deben ser números muy próximos.

2. Ambos $(p - 1)$ y $(q - 1)$ deben contener factores primos grandes.
3. El mcd $(p - 1, q - 1)$ debe ser pequeño.

Además, en la literatura especializada están documentadas diversas recomendaciones para conseguir determinar fácilmente estos parámetros sin disminuir la fortaleza del algoritmo. Otra recomendación práctica consiste en que se elija una clave pública pequeña, lo que conlleva que su inversa, la clave privada, sea mayor. No representa ninguna debilidad que muchos usuarios elijan idénticos valores para e porque en cada caso n será distinta. Los tres valores más frecuentemente elegidos son 3, 17 y $2^{16} + 1$. Con ello se consigue que el cifrado de un mensaje para conseguir confidencialidad sea más sencillo que la operación inversa de descifrado que realiza el usuario poseedor de la clave privada. Asimismo, la verificación de la autoría en un mensaje cifrado con la clave secreta será más rápida que el correspondiente cifrado.

De todo lo que antecede se deduce que la generación de claves es un problema fácilmente abordable, pero bastante más complejo que las operaciones de cifrado y descifrado (también calificadas como «fáciles» en el planteamiento teórico de los criptosistemas de clave pública que hicimos en el primer apartado de este capítulo). Se estima que, desde un punto de vista algorítmico, si k es el número de bits de n , el orden de magnitud (véase epígrafe 3.3) que informa sobre el número de *operaciones bit* necesarias es de $O(k^2)$ para las operaciones de cifrado con la clave pública, $O(k^3)$ para las operaciones de cifrado con la clave privada y $O(k^4)$ para la generación de claves. Esta diferencia en la dificultad que entrañan las distintas operaciones da lugar a que dentro de las tarjetas inteligentes que poseen facilidades de cifrado asimétrico, sólo las que son más potentes y tienen mayor memoria ROM son capaces de generar claves.

¿ES FIABLE EL USO DE RSA PARA LA SEGURIDAD EN REDES?

En líneas generales, RSA se considera que es entre 100 y 10.000 veces más lento que el DES. Si a eso se añade que, según ya se discutió, los algoritmos más avanzados, como son el IDEA y el futuro AES, son mucho más rápidos que DES, es fácil concluir que, como ya habíamos anunciado, RSA debe utilizarse solamente para cifrar piezas de información de reducido tamaño, siendo necesario recurrir a mecanismos de firma digital que no se basen en cifrar todo el mensaje completo con la clave privada, así como al uso combinado de criptografía simétrica y de clave pública cuando se trate de proveer los tres servicios básicos de seguridad.

El criptosistema RSA fue patentado en su día, lo que ha supuesto limitaciones en su uso. Por otra parte, las leyes de Estados Unidos han restringido notablemente el tipo de software que se permitía exportar, imponiendo unas condiciones para el tamaño de las claves y el tipo de operaciones que situaban la seguridad del sistema en sus cotas más bajas. Además del ingenio para no hacer efectivas estas limitaciones, la expiración de los derechos de patente en el año 2000 ha hecho que sea este sistema el que de forma mayoritaria se utilice para la configuración de protocolos telemáticos seguros.

Otra cuestión es la aparición, de vez en cuando, de noticias que dan cuenta de experiencias en las que se ha conseguido romper la seguridad del RSA. El día que esto se consiga de verdad, este algoritmo dejará de ser útil para la Seguridad en Redes telemáticas, pero seguirán siendo válidos los conceptos en que ésta se apoya y los protocolos que se han especificado, de forma que solamente será necesario rehacer las implementaciones utilizando un nuevo algoritmo. En la criptografía de curvas elípticas

o la criptografía cuántica, que se comentan más adelante, puede estar la solución. No obstante, la realidad presente está muy lejos de corresponderse con esta situación: las rupturas documentadas en la literatura especializada (reales o vislumbradas) han sido llevadas a cabo usando máquinas especiales o equipos de múltiples personas, y se han realizado con claves de un número de bits mucho menor que el de las claves que es posible manejar con un simple computador personal. Usando los algoritmos de ruptura más avanzados, para la factorización de números de entre 120 y 130 dígitos decimales se ha previsto un número de MIPS-año del orden de 1.000. (Un MIPS-año equivale a un procesador que ejecuta un millón de instrucciones por segundo funcionando durante un año.) Ello significa que si se usa para n un valor de más de 200 dígitos decimales (que podría irse aumentando a medida que progrese la potencia de los computadores), las posibilidades de rupturas son despreciables, al menos para los objetivos que se persiguen en el intercambio de mensajes que se producirán en el ámbito normal de ciudadanos normales.

5.7. ALGORITMO DE DIFFIE-HELLMAN PARA INTERCAMBIO DE CLAVES

Este fue el primer algoritmo de clave pública que se inventó. Se hizo público en el artículo, ya repetidamente citado [DiHe76], en el que Diffie y Hellman pusieron las bases de lo que más tarde ha sido la criptografía asimétrica. El nombre con el que se conoce (*Key Exchange*, en inglés) es bastante poco adecuado como reconoce su propio autor ([DiLa98] página 253), ya que realmente no existe ningún intercambio de claves. Debería llamarse *de negociación de clave* (en singular) o de *concertación de clave*, ya que de eso se trata: de elegir una clave secreta de forma acordada entre dos entidades que desean establecer una comunicación segura a través de un canal inseguro. Como, en su día, el procedimiento fue patentado con ese nombre de «exchange», así se ha seguido denominando en todos los foros (por aquello de «mantenella y no enmendalla»), denominación que aquí nos vemos forzados a conservar en aras a un mejor entendimiento. El procedimiento es como sigue:

- a) Las entidades A y B eligen, de forma pública, un número primo p muy grande y un generador $g < p$ tal que sea una raíz primitiva de p , es decir, tal que sus sucesivas potencias permitan generar todos los enteros comprendidos entre 1 y $p - 1$.
- b) A elige un entero muy grande x , calcula

$$X = g^x \bmod p$$
 y lo envía a B:
 A $\rightarrow$ B: X
 (Info A en Figura 5.12)
- c) B elige un entero muy grande y , calcula

$$Y = g^y \bmod p$$
 y lo envía a A:
 B $\rightarrow$ A: Y
 (Info B en Figura 5.12)

- d) A calcula: $Y^x = (g^y)^x \bmod p = k$
 B calcula: $X^y = (g^x)^y \bmod p = k'$
- e) A y B han encontrado por separado la misma clave, ya que $k = k'$. En efecto:
 $k = Y^x = (g^y)^x \bmod p = g^{yx} \bmod p = g^{xy} \bmod p = (g^x)^y \bmod p = X^y = k'$.

Esta clave k puede servirles a las entidades A y B para establecer una comunicación mediante algoritmo asimétrico.

Cualquier intruso I que intercepte la comunicación y capte X o Y (conoce, además, p y g porque son de dominio público) podría calcular k si fuese capaz de resolver:

$$y = \log_g Y \bmod p \quad \text{o} \quad x = \log_g X \bmod p$$

que constituyen problemas N-P completos, inabordables por tanto.

El algoritmo puede extenderse sin excesiva complicación a tres participantes, sin más que la nueva entidad C elija un nuevo valor z y su correspondiente potencia Z de forma que al final las tres entidades encuentren por separado el valor $g^{xyz} \bmod p$, que sería la clave simétrica que compartiesen. Siguiendo un procedimiento similar, el método puede extenderse a cuatro o más entidades participantes.

Este algoritmo es muy usado en los mecanismos de seguridad que constituyen los protocolos telemáticos seguros. Aunque en su día fue patentado, en 1997 expiró esa patente, lo que facilita su empleo en las implementaciones prácticas.

5.8. OTROS ALGORITMOS DE CLAVE PÚBLICA

Existen multitud de propuestas sobre criptosistemas asimétricos. La mayoría de estos algoritmos están basados en problemas de exponenciación en aritmética modular y en la dificultad que presentan la radicación y los algoritmos discretos. Algunos son inseguros porque se han publicado procedimientos de ruptura, otros son impracticables y sólo unos cuantos son útiles.

El de Merkle-Hellman o de la Mochila es considerado por algunos como el primer algoritmo de clave pública inventado, aunque tiene el defecto de que, además de haber sido ya vulnerado, no es bidireccional en el sentido de que no admite el cifrado con la clave privada, cosa que también ocurre con el de Massey-Omar. El algoritmo de Pohlig-Hellman tiene un esquema similar al RSA, y es bastante robusto, pero al igual que los dos anteriormente citados, sólo sirve para cifrar o encriptar mensajes mediante el cifrado con la clave pública del receptor.

Además de RSA, también los algoritmos de Rabin y de El Gamal posibilitan realizar operaciones bidireccionales, es decir, permiten cifrar tanto con la clave privada como con la pública. El de Rabin, que data de 1979, se basa en la dificultad de obtener raíces cuadradas módulo n , siendo n un número producto de dos primos de gran tamaño. El algoritmo de El Gamal, presentado en 1984, se basa en la dificultad de calcular el logaritmo discreto de un número, y el procedimiento seguido para el cifrado con la clave pública es distinto del seguido para el cifrado con la clave privada, por lo que este último se considera como un algoritmo de firma propiamente dicho. El interés del algoritmo de El Gamal radica en que el algoritmo de firma DSA utiliza un esquema de cifrado similar al suyo.

La descripción de estos algoritmos y la discusión acerca de su robustez puede encontrarse en tratados especializados en criptografía. A esas fuentes se remite al

lector interesado en estos criptosistemas, ya que aquí obviaremos entrar en mayores detalles, habida cuenta que, en lo que se refiere a los protocolos utilizados para la Seguridad en Redes Telemáticas, son los algoritmos RSA y Diffie-Hellman los que casi exclusivamente se utilizan (además de los algoritmos de firma que se describirán en el Capítulo 8).

5.9. CIFRADO CON CLAVE SIMÉTRICA CONCERTADA ENTRE LAS PARTES

Este esquema de trabajo, que ya anunciamos en el epígrafe 5.4, es la otra posibilidad de uso combinado de la criptografía simétrica. A veces puede interesar separar el proceso de transferencia de datos del de generación y transferencia de la clave de sesión, al contrario de lo que ocurre con el mecanismo de envoltura digital en el que los criptogramas que se envían contienen información suficiente para encontrar la clave de sesión y al mismo tiempo incluyen el mensaje cifrado (para garantizar la confidencialidad).

Esa situación de separación suele darse en el caso de que se desee transmitir un mensaje m largo con una velocidad de transmisión muy elevada, de tal manera que las tareas de autenticación y recuperación de la clave por parte de B pueda ralentizar indebidamente el proceso. Un ejemplo de ello lo podemos encontrar en las aplicaciones de videoconferencia protegidas con servicios de seguridad. Cuando se inicia una comunicación, lo primero que se hace es concertar una clave de sesión para ser utilizada en esa videoconferencia, y después, en una segunda fase, se inicia, con una elevada velocidad, la transferencia de tramas de vídeo y de audio cifradas con esa clave concertada.

Los distintos mecanismos existentes para conseguir este objetivo pueden agruparse en dos esquemas distintos:

a) ESQUEMAS BASADOS EN ALGORITMOS SIMILARES AL DE DIFFIE-HELLMAN

Generalmente se apoyan en este algoritmo, aunque existen otros algoritmos equivalentes y algunas variantes del de Diffie-Hellman que amplían las posibilidades del algoritmo básico, pero que, sustancialmente, conservan las mismas características principales.

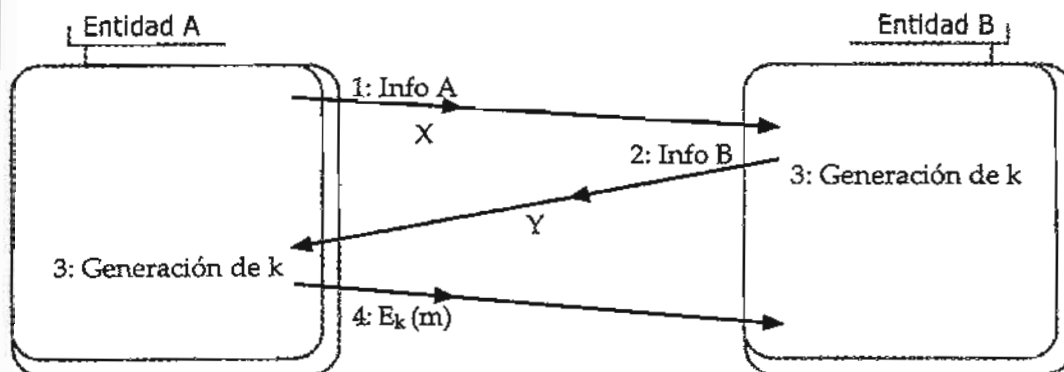


Figura 5.12. Cifrado con clave simétrica concertada (esquema Diffie-Hellman).

En la Figura 5.12 se recoge el procedimiento seguido:

1. En primer lugar, la entidad A genera una pieza de información, que en el esquema de la figura hemos llamado *Info A*, apoyándose en determinados parámetros (públicos) y características de un criptosistema de clave pública y en una información secreta generada internamente por A. Esta *Info A* se la envía a la entidad B.
2. La entidad B procede de forma similar y genera y envía su *Info B*.
3. Por separado y de forma independiente, A y B generan una clave secreta k que, debido a las características del criptosistema de clave pública, sólo A y B pueden conocer.
4. La entidad A cifra el mensaje m usando la clave k y un criptosistema de clave secreta.

La clave secreta generada es una *clave de sesión* porque puede ser usada para una sola transacción entre A y B o para varias seguidas, según decidan estas entidades. Además de necesitar tres intercambios de información entre las entidades para completar la transferencia segura del mensaje (cada vez que se renueve la clave de sesión), este esquema no conlleva autenticación, por lo que es muy vulnerable a un ataque de los denominados de «persona interpuesta» o «persona en el medio». En efecto, un usuario malicioso M colocado entre las dos entidades comunicantes que conozca la intención de A de establecer una comunicación segura con B, puede, al recibir *Info A*, hacerse pasar por B y enviar, tanto a A como a B, piezas de información por él generadas y obtener de A la información confidencial que cree estar enviando a B. Si engaña a B haciéndole creer que es A puede incluso conseguir que reciba los mensajes procedentes de A sin que se percaten de que están siendo objeto de un ataque pasivo (no se altera aparentemente el servicio de comunicación).

b) ESQUEMAS BASADOS EN ALGORITMOS DEL MISMO TIPO QUE RSA

En este caso no se recurriría al mecanismo de *envoltura digital* tal y como ha quedado descrito en 5.4, sino a un desdoblamiento del mecanismo: tres intercambios de información realizados uno detrás de otro para la concertación de la clave de sesión y otro (o más) para la transferencia del mensaje que se desee hacer llegar a la otra entidad comunicante.

En la Figura 5.13 se recoge simplificadaamente el procedimiento seguido (omitendo las firmas y algunos otros datos complementarios).

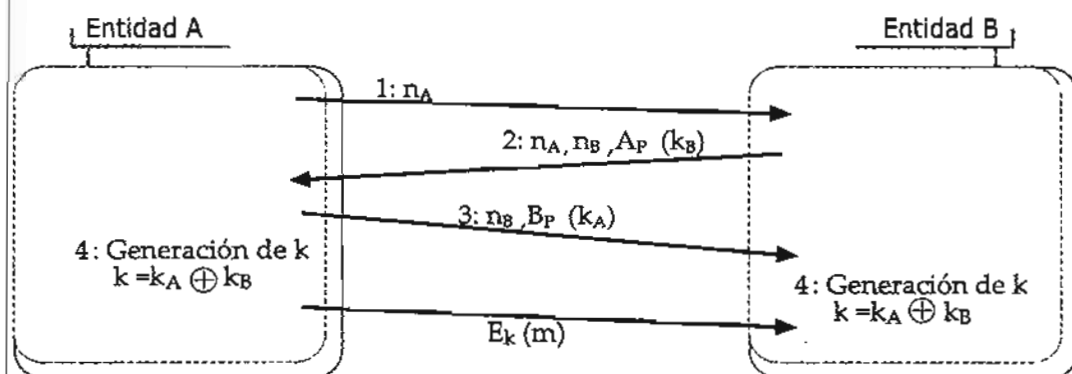


Figura 5.13. Cifrado con clave simétrica concertada (esquema RSA).

1. La entidad A averigua el nombre identificativo de B (al que denominaremos B), genera un número aleatorio n_A , firma el conjunto de estos dos datos y se los envía a B:

$$A \rightarrow B: n_A, B, A_{\text{sig}}(n, B)$$

2. La entidad B verifica la firma para identificar a A, averigua el identificador de A, lee el número aleatorio generado por A, genera un número aleatorio n_B , genera «su» parte de la clave k_B , firma el conjunto de estos datos, cifra k_B con la clave pública de A y se los envía a A:

$$B \rightarrow A: n_A, n_B, A, k_B, B_{\text{sig}}(n_A, B, k_B), A_P[k_B]$$

3. La entidad A verifica la firma para identificar a B y comprueba la integridad del mensaje, reconoce que n_A es el número que ella envió, descifra la parte de la clave generada por B, lee el número aleatorio generado por B, genera «su» parte de la clave k_A , firma el conjunto de estos datos, cifra k_A con la clave pública de B y envía a B:

$$A \rightarrow B: n_B, B, A_{\text{sig}}(n_B, B, k_A), B_P[k_A]$$

4. Por separado y de forma independiente, A y B generan una clave secreta k que, debido a las características del criptosistema de clave pública, sólo A y B pueden conocer:

$$k = k_A \oplus k_B$$

La entidad A puede realizar el cálculo inmediatamente después de realizar las comprobaciones especificadas en el punto 3. La entidad B lo realizará una vez recibido el mensaje especificado en el punto 3 y verificar la firma para identificar a A, comprobar la integridad del mensaje, reconocer que n_B es el número que ella envió y descifrar la parte de la clave generada por A.

5. La entidad A cifra el mensaje m usando la clave k y un criptosistema de clave secreta.

En realidad, en este mecanismo de concertación de clave, de forma previa se está llevando a cabo un proceso de autenticación. Para que ésta sea efectiva tienen que concurrir una serie de comprobaciones acerca de la validez de las claves públicas utilizadas en ese escenario. Esta no es una tarea simple y requiere del concurso de determinadas TTPs, por lo que se estudiarán con más detalle en el Capítulo 8 dedicado por entero a cuestiones relacionadas con la autenticación.

5.10. UNA PUERTA ABIERTA: CRIPTOGRAFÍA DE CURVAS ELÍPTICAS

La criptografía basada en curvas elípticas no representa, en el momento actual, ninguna novedad conceptual, ya que es una técnica que viene despertando el interés de los criptólogos desde hace bastantes años. Lo que sí puede ser novedoso es su utilización en protocolos de Seguridad en Redes, cuestión que está siendo considerada y sopesada en razón de las ventajas que ofrece en relación con los algoritmos, ya clásicos, que basan su fortaleza en problemas de difícil solución en aritmética modular.

Las primeras propuestas sobre criptografía de curvas elípticas, ECC, datan de 1985, y desde entonces han aparecido multitud de contribuciones que han ido aportando tanto explicaciones para su justificación teórica como realizaciones concretas de

algunos algoritmos. Asimismo, los distintos organismos que se ocupan de la normalización de protocolos en redes telemáticas han asumido esta temática con el objetivo de producir los correspondientes estándares. No obstante, como ya antes se dijo, el predominio de RSA y del clásico Diffie-Hellman es, hoy por hoy, casi absoluto debido al mayor nivel de confianza que todavía despiertan entre los usuarios.

Una primera cuestión que conviene aclarar es que con la ECC no se trata, principalmente, de crear nuevos algoritmos, sino de configurar nuevos elementos que constituyan grupos finitos en base a los cuales implementar los algoritmos de clave pública ya existentes, como puede ser el caso de los algoritmos de Diffie-Hellman, El Gamal, etc. Lo que sí se postula es que usando ECC los algoritmos son más rápidos y más robustos para **claves** muchísimo **más cortas**. Es por ello que pueden constituir, en un futuro próximo, una buena opción para la implementación de protocolos telemáticos seguros. En efecto, para defenderse del aumento de posibilidades de criptoanálisis derivadas del creciente aumento de la potencia de los computadores es necesario ir aumentando el tamaño de las claves, y ello complica el transvase de información que es necesario llevar a cabo mediante los mecanismos de distribución de claves, que quedaría aliviado si fuese mucho menor el tamaño de las claves puestas en juego.

Las matemáticas necesarias para dar una explicación suficiente de los conceptos que aparecen en la ECC superan, en mucho, las contenidas en el resumen teórico recogido en el Capítulo 3, resumen que se ajusta a los requisitos necesarios para entender adecuadamente los algoritmos antes descritos, que se apoyan en los postulados de la aritmética modular. Debido a la todavía no evidente e inminente utilización práctica de la ECC en los protocolos de Seguridad en Redes, por cuestión de oportunidad, se ha decidido no abordar aquí el estudio riguroso de los conceptos en que se basa (con la justificación matemática que requiere). Una descripción de este tipo puede encontrarse en [Mene93].

Lo que se presenta a continuación es un resumen sencillo y esquemático de los elementos principales que configuran la criptografía de curvas elípticas, prescindiendo de su justificación matemática, pero sí destacando las conclusiones principales a las que se ha llegado. Se ha procurado, además, que resulte inteligible a la luz de los conceptos antes adquiridos, y que deje abierto el camino para futuras ampliaciones y profundizaciones.

Una *curva elíptica* es una curva plana definida por una ecuación de tercer grado de la forma:

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

donde a , b , c y d son números pertenecientes a un cuerpo finito K . Por ser de tercer grado, dados dos puntos P y Q sobre la curva, la recta que los une corta a la curva en un tercer punto R .

La Figura 5.14 representa un caso particular de curva elíptica definida por la ecuación:

$$y^2 = x^3 + x + 1$$

En la figura se han representado tres puntos alineados. Se puede definir un tipo de suma tal que los puntos que yacen sobre la curva forman un grupo abeliano aditivo (véase apartado 3.2). Se define también un punto O , llamado *punto cero*, que es el elemento neutro respecto a la suma, de forma que:

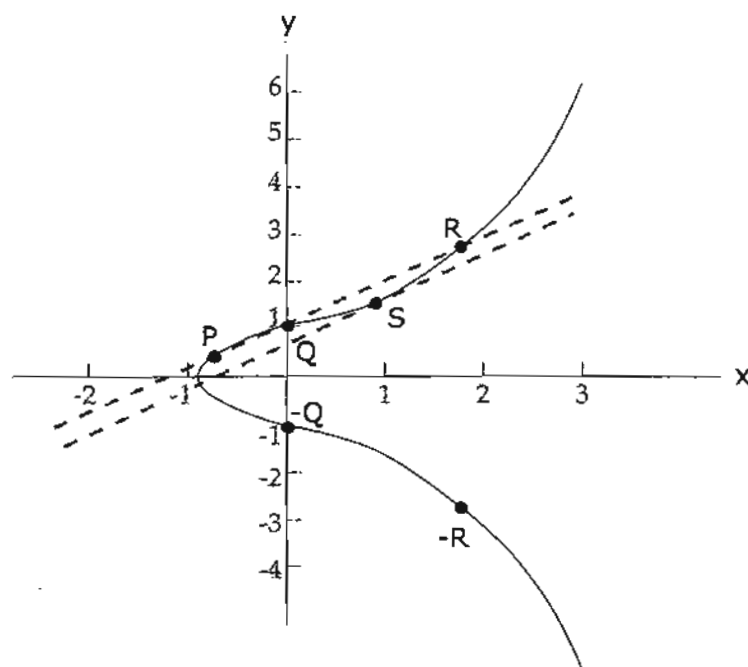


Figura 5.14. Ejemplo de curva elíptica.

- La suma de tres puntos P , Q y R , alineados que yacen sobre la curva es O .
- Una recta vertical corta a la curva en dos puntos $R(x_1, y_1)$ y $R'(x_1, -y_1)$, que tienen igual ordenada y distintas abscisas. Por definición, esa recta vertical también corta a la curva en el punto imaginario O . Es decir, $R + R' + O = O$, o lo que es lo mismo, $R' = -R$. También se representan en la figura Q y $-Q$, resultantes de los puntos de corte de la curva con el eje de abscisas.
- La suma de dos puntos P y Q de diferentes coordenadas es otro punto R que se encuentra en la intersección entre la curva elíptica y la recta trazada uniendo los puntos P y Q .
- La suma de un punto S consigo mismo, o sea la multiplicación por 2, se obtiene trazando una tangente a la curva en ese punto y encontrando el otro punto de corte, T . Se verifica que:

$$S + S = -T$$

- La multiplicación se define mediante sumas sucesivas de un punto consigo mismo.

A los efectos de la utilización de las curvas elípticas en criptografía, sólo interesa el caso en que estén definidas sobre el cuerpo finito $GF(q)$, donde q es un número primo o potencia de un número primo, y se cumplen, además, determinadas restricciones con los coeficientes (para más detalles véase [Stal99]). La curva del ejemplo pertenece a ese grupo de curvas elípticas.

Dada una determinada curva elíptica, por ejemplo la ecuación $y^2 = x^3 + x + 1$ representada en la figura, y un determinado número primo, que ha de satisfacer las restricciones antes citadas (por ejemplo, el número 23), para las operaciones matemáticas a emplear en las transformaciones criptográficas sólo interesan los puntos con coordenadas enteras positivas situados en el cuadrante delimitado por el origen de coordenadas y el punto (23,23).

En la suma y multiplicación de puntos se aplicarían las reglas antes indicadas, de forma que cuando una de las coordenadas resulte negativa o mayor de 23, sea susti-

tuida por el resultado positivo de operarla módulo 23. Así, por ejemplo, si observamos la figura o realizamos sobre la ecuación el cálculo correspondiente, Q es el punto de coordenadas (0,1) y $-Q$ es el punto de coordenadas (0,-1). Es decir, a efectos del grupo elíptico, las coordenadas de Q serían (0,22), ya que $(-1) \bmod 23 = 22$.

Se ha llegado a la conclusión de que, en cuanto a grado de dificultad, en ECC la operación de suma juega un papel similar al de la multiplicación de números en aritmética modular, que antes vimos. De igual forma, la múltiple adición de puntos sería aquí el equivalente a las operaciones de exponenciación en aritmética modular.

Al igual que en aritmética modular nos apoyábamos en la dificultad de factorizar primos muy grandes y calcular logaritmos discretos, en la criptografía basada en curvas elípticas, a la hora de buscar problemas de difícil solución se cuenta con la ecuación $Q = kP$, en la que k es un número entero menor que un primo q que se toma como módulo. Es relativamente fácil calcular Q conocidos k y P , pero es difícilísimo calcular k conocidos P y Q .

Apoyándose en estas propiedades, se han definido esquemas que reproducen las características del algoritmo de El Gamal, del de Diffie-Hellman, del algoritmo de firma digital DSA, o, incluso, de un equivalente de RSA. En todos estos casos, parece que la versatilidad de los sistemas que se han implementado permite un uso sencillo por parte de los usuarios a la hora de elegir y gestionar sus claves. Además, según se adelantaba al principio de este apartado, en comparación con los criptosistemas «clásicos» como RSA, el tamaño de claves que se necesita utilizar en ECC es, a igualdad de seguridad, muchísimo menor.

5.11. ¿CÓMO AFECTARÁ LA EVOLUCIÓN DE LA CRIPTOGRAFÍA A LA SEGURIDAD EN REDES?

En los tres últimos capítulos se ha procurado condensar lo más significativo de las técnicas criptográficas en lo que respecta a su utilización en las protecciones que proporcionan los servicios telemáticos de seguridad. Con anterioridad, en los Capítulos 1 y 2, se sentaron las bases de los principales conceptos y elementos que forman parte de la Seguridad en Redes Telemáticas: el análisis de riesgos, la definición de una política de seguridad adaptada a las necesidades, la determinación de los servicios de seguridad que se vayan a proveer, el diseño arquitectural que determine todos los agentes y protocolos que deben estar presentes, la definición de los protocolos telemáticos seguros y, en último término, los mecanismos de seguridad que, a modo de piezas digitales, sirven para configurar esos protocolos. Precisamente, en la materialización de esos mecanismos es donde entran en juego las técnicas criptográficas.

De todo ello se deduce que las técnicas empleadas para la securización de las comunicaciones simplemente utilizan esas piezas digitales. Podemos decir que, de alguna manera, la validez de los planteamientos presentes en Seguridad en Redes está por encima de ellas y a salvo de las obsolescencias que se vayan descubriendo. Dicho de otra manera: cuando algún algoritmo criptográfico o alguno de los esquemas en que se basan quede demostrado que ha perdido fortaleza, lo que habrá que hacer es sustituirlo por otro más moderno que resulte invulnerable en ese momento.

No obstante, además de las salvaguardas recogidas en el párrafo anterior, cabe decir que las expectativas acerca de la pervivencia de los planteamientos criptográficos recogidos en los tres últimos capítulos son francamente halagüeñas para las próximas décadas. En el epígrafe 4.6, con motivo de la discusión acerca del nuevo algoritmo avanzado de cifrado simétrico, el llamado AES, se hizo una breve referencia a

la computación cuántica que es posible que se desarrolle en un futuro no muy lejano y a la mejora que ello supondría en la resolución de problemas de cálculo complejos. También se llegó a la conclusión de que con los requisitos exigidos en AES se podrá contar, durante muchos años, con una seguridad suficiente en lo que respecta a los criptosistemas de clave secreta.

Desde luego que cuando llegue, si llega, el momento en que se desarrollen computadores basados en elementos de física cuántica, los problemas en que se basa la fortaleza de los algoritmos convencionales, como es el caso de la factorización de números primos o del algoritmo discreto, serían resolubles en tiempos razonables, lo que supondría la invalidez de estos esquemas. En efecto, algunos problemas matemáticos de difícil solución que en la actualidad requieren un número de operaciones bit con un orden de magnitud $O(n \cdot 2^n)$, en su versión cuántica el orden de magnitud se reduce a $O(n^2)$. En concreto, se pronostica que el problema de la factorización de un número primo grande de n bits podría requerir un esfuerzo computacional de sólo n^3 pasos.

Sin embargo, la realidad es poco dramática, ya que se estima [Beth00] que para una clave pública de 1.000 bits sería necesario un ordenador cuántico de 30.000 q-bits (bits cuánticos), mientras que hasta la fecha sólo se han realizado experimentos con un número de q-bits muy pequeño. Existen cálculos muy bien fundamentados que pronostican que un computador cuántico capaz de factorizar un número entero de 1.000 bits no debe esperarse antes de treinta años.

Esos mismos cálculos auguran que utilizando claves de 2.000 bits en los criptosistemas de clave pública convencionales se puede confiar en la seguridad de los cifrados hasta aproximadamente el año 2015. Además, a diferencia de lo que ocurre en los criptosistemas simétricos, en los de clave pública el tamaño de la clave puede irse aumentando sin que ello suponga ningún cambio en la configuración del algoritmo, de lo cual se deduce que nada impide que en los sistemas que se utilicen en los próximos años podamos contar con claves del orden de 4.000 bits o aún mayores.

Si a esto se añade la reserva tecnológica que representa la criptografía de curvas elípticas, que se encuentra en fase de superar los análisis criptográficos necesarios para determinar su adecuación a la securización de aplicaciones telemáticas, se puede adquirir una razonable sensación de tranquilidad en cuanto al futuro de las técnicas criptográficas para las primeras décadas del siglo XXI. Bien es verdad que es muy difícil aventurar pronósticos, sobre todo cuando se refieren a periodos de tiempo tan largos, por muy solventes que sean quienes los hagan, y es muy probable que nuevos descubrimientos den al traste con las previsiones que se están realizando. De lo que sí podemos estar seguros es de que, llegado el momento, se encontrarán las soluciones adecuadas para que sigan existiendo las protecciones adecuadas para ofrecer a la ciudadanía unas comunicaciones seguras.

NOTAS AL CAPÍTULO 5

¹ Este es el caso de B. Schneier en su libro [Schn96] tantas veces aquí citado. Además, como muchos otros autores estadounidenses, sigue prefiriendo usar el término «encrypt» (y sus derivados) frente al «cipher» recomendado por ISO, incluso en el caso de que la operación de cifrado no represente ningún tipo de ocultación.

² Esta norma es la ISO/IEC 9594 (véase el apartado de referencias). Es una parte del conjunto de normas sobre el Directorio. En ella se aborda el tema de la estructura del certificado (que se verá con cierto detalle en el capítulo siguiente) y el esquema de autenticación, que se estudia en el Capítulo 8.

³ En los tratados sobre Criptografía se acostumbra a denominar como «protocolo» a cualesquiera de los intercambios de información tendentes a obtener una determinada protección. Nosotros, cuando se trata de estos elementos atómicos que sirven para configurar un *protocolo telemático de seguridad*, preferimos utilizar (en concordancia con la norma ISO 7498-2 *Security Architecture*) el término *mecanismo de seguridad*, reservando el término *protocolo* para el conjunto de reglas y formatos que determinan el intercambio de piezas de información (por ejemplo, lo que determina el comportamiento de una *aplicación telemática*) que proporciona un *servicio telemático seguro*. En esos casos, además de los mecanismos de seguridad utilizados será necesario especificar cómo y de qué manera se da a conocer la identidad de las entidades, las garantías sobre la validez de las claves públicas utilizadas, etc.

⁴ Los escenarios de comunicación en los que unas TTPs específicas sirven para la provisión de los servicios de no repudio se estudian en el Capítulo 9 dedicado a ese servicio junto con el servicio de control de acceso.

CAPÍTULO

6

DISTRIBUCIÓN DE CLAVES EN REDES TELEMÁTICAS. CERTIFICADOS Y AUTORIDADES DE CERTIFICACIÓN

Aunque en los capítulos precedentes la selección de conceptos y la valoración de su importancia ha estado guiada, en todo momento, por el grado de repercusión que tienen para la Seguridad en Redes, hasta ahora no se ha recogido, en su justa medida, la influencia que tiene la presencia de un número elevado de entidades de comunicación. Sin embargo, cuando se trata de resolver el problema de la protección de las comunicaciones en las redes telemáticas, se comprueba que la dificultad que ello conlleva depende en gran parte del número de equipos conectados, de su distribución geográfica, del grado de independencia que exista entre ellos y de su organización administrativa.

Mediante las técnicas criptográficas se consigue extender la confidencialidad y la integridad de unas piezas de información de reducido tamaño, las claves, a otras piezas de información mucho más voluminosas y numerosas: los mensajes que circulan por la red o que en ella se almacenan. La adecuada generación, distribución y protección de las claves equivale, por tanto, a la adecuada protección de los mensajes.

En el presente capítulo se abordan algunos de los más importantes aspectos de la gestión de claves, tanto para entornos de clave secreta como de clave pública. No obstante, es este último caso el que, por su trascendencia, acapara la mayor parte de los apartados. En efecto, puede decirse que el capítulo está dedicado casi por entero al Certificado y a las Autoridades de Certificación. Es más, es tal la importancia y trascendencia del tema de la certificación, que en los dos siguientes capítulos continuarán apareciendo otros nuevos aspectos y conceptos con ella relacionados.

6.1. DISTRIBUCIÓN EN RED DE CLAVES SIMÉTRICAS

Una red telemática es un sistema que posibilita el intercambio de información entre los múltiples y diversos sistemas informáticos a ella conectados. El número y variedad

de estos sistemas informáticos puede variar entre unos cuantos PCs de similares características conectados a una red de área local, hasta los miles o millones de máquinas a las que se puede tener acceso a través de Internet. Los problemas que se presentan para la gestión de claves (generación, distribución, evaluación de vigencia, contabilización y eliminación de las no válidas) constituyen uno de los elementos fundamentales que es necesario resolver para la implantación de seguridad en estas redes, de tal manera que la dificultad que ello conlleva depende más del número de equipos involucrados en un determinado dominio de seguridad que de los algoritmos o criptosistemas que sea necesario soportar en cada extremo.

COMUNICACIÓN EXTREMO A EXTREMO

En un dominio en el que la seguridad de las comunicaciones descansa solamente en criptosistemas de clave secreta, la generación y el reparto de claves es una materia muy sensible y delicada. Dada una red con n entidades comunicantes, tal que todas puedan establecer comunicación con todas mediante conexiones punto a punto (Figura 6.1), de forma que no compartan clave secreta más que dos a dos, el número total de claves que sería necesario poner en juego sería del orden de n^2 (exactamente, $n(n-1)/2$).

Este esquema sólo puede ser viable cuando se trata de dominios de seguridad con un número reducido de usuarios, en el que, además, éstos tengan alguna relación personal o corporativa que haga que tengan entre sí cierto nivel de confianza y la posibilidad de intercambiarse claves manualmente o a través de un medio extratelemático. Este último extremo, aparentemente secundario, tiene en la práctica una gran importancia, porque antes de establecerse una comunicación segura no es posible mandar ninguna información sensible por vía telemática. Esto es así porque un posible usuario intruso o malintencionado, contra el que se tratan de establecer las protecciones, podría tener acceso a las claves de seguridad desde el principio. Otra cuestión no baladí sería definir una política de seguridad que determinase quién, en cada pareja, es el que genera las claves secretas y qué elementos de almacenaje y protección ha de poseer cada usuario para custodiar las $n-1$ claves de que tiene que estar provisto.

Otro caso muy distinto es cuando el número de usuarios es considerable. No digamos ya si se pretendiese que perteneciesen a organizaciones distintas y bastante separadas geográficamente, como es el caso de las entidades conectadas a redes de área extensa. Los problemas que ello conlleva son de tal magnitud, que, como ya se ha comentado con anterioridad, hacen inviable un esquema de este tipo.

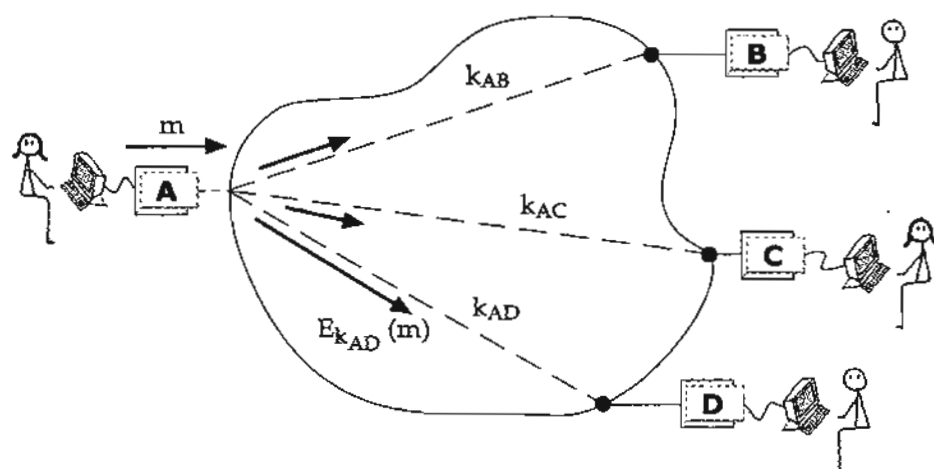


Figura 6.1. Comunicaciones cifradas extremo a extremo.

COMUNICACIÓN CENTRALIZADA

Presentaremos tres variantes de este tipo de comunicación. El esquema más primitivo que puede establecerse, al que denominaremos Esquema 1, se corresponde, de forma simplificada, con el que tenían los servicios secretos en la época de la guerra fría. Consiste en disponer (Figura 6.2) de un elemento central, al que podemos llamar CDC, Centro de Distribución de Claves (*key distribution center*, KDC, en inglés) de tal manera que cada entidad sólo puede establecer comunicación con el elemento central, que además es quien genera todas las claves y las distribuye, por medios extratelemáticos, a todos y cada uno de los participantes. Cuando una entidad A desee enviar información a otra entidad B, debe enviarla al CDC cifrada con la clave que con ella comparte, el cual la descifra y la reenviará a la entidad destinataria B cifrada con la correspondiente clave secreta. La responsabilidad acerca de la idoneidad de las claves y la determinación de cuándo es llegado el momento de la renovación corresponde, también en exclusiva, a dicho elemento central. La única responsabilidad que adquieren los usuarios terminales es la de custodiar adecuadamente la clave que les ha sido confiada.

Es evidente que los usuarios están protegidos contra la posible injerencia de terceros, pero tienen que confiar ciegamente en el CDC. Cuando se trata de una organización tan fuertemente jerarquizada y dependiente como se supone que puede ser el mundo ese de los espías y demás, este tipo de soluciones pueden ser consideradas como satisfactorias, pero cuando se trata de intercambiar información entre ciudadanos que quieren mantener la privacidad en sus comunicaciones, esta solución resulta de todo punto inadecuada e inaceptable.

Aunque cueste creerlo, con pequeñas variantes, esta es la solución adoptada por el muy moderno y europeo estándar GSM para proporcionar seguridad en las comunicaciones entre teléfonos móviles. Sobran comentarios.

El Esquema 2, representado en la Figura 6.3, es una variante algo edulcorada de la situación anterior y consiste en perfeccionar un poco el modelo para ganar algo de privacidad. Ya no se trata de que la entidad A haga pasar a través del CDC todo lo que quiera enviar a B. Al igual que en el caso anterior, el elemento central, que ahora podríamos denominar, de forma más genérica, como CGC, Centro de Gestión de Claves (*key management facility*, KMF, en inglés), comparte clave secreta con todos y cada uno de los miembros de la red. Si la entidad A desea establecer una

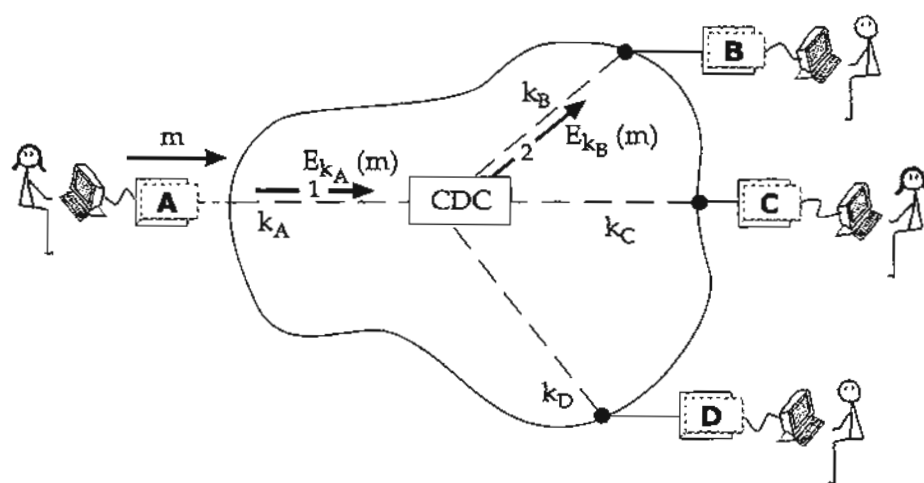


Figura 6.2. Comunicación a través de un Centro de Distribución de Claves.

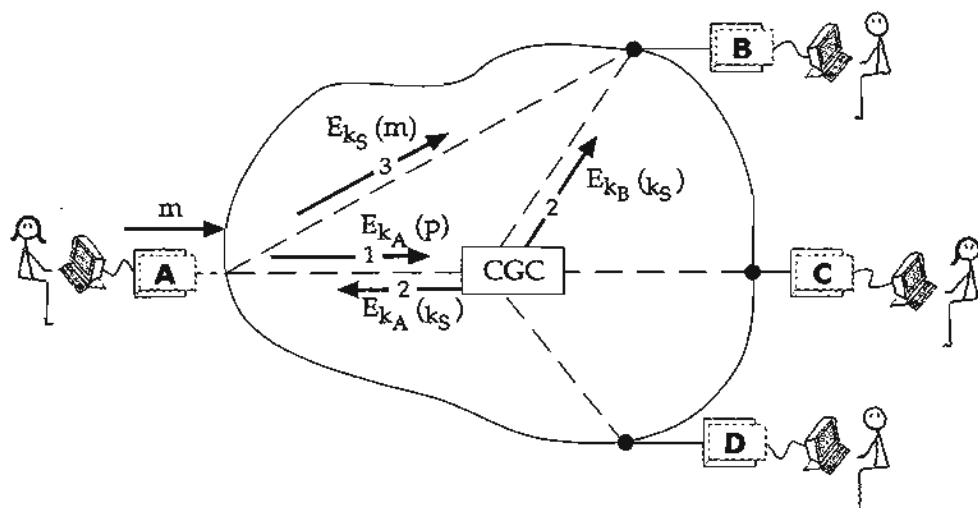


Figura 6.3. Obtención de claves a través de un Centro de Gestión de Claves.

comunicación con otra entidad B perteneciente al mismo dominio, que también confía, por tanto, en el CGC, debe procederse como sigue:

1. A debe enviar al CGC un mensaje p cifrado con la clave secreta que con él comparte, comunicándole que desea comunicarse con B.
2. Si el CGC reconoce a B como perteneciente al dominio, genera una clave de sesión (k_s) para uso exclusivo de A y B, y se la envía a ambas entidades interlocutoras cifrada, respectivamente, con la clave que comparte con cada una de ellas.
3. Las entidades A y B comparten ahora una clave para poder comunicar entre sí de forma confidencial sin que el elemento central tenga que cifrar cada uno de los mensajes m que intercambien. Dependiendo de la política de seguridad que esté establecida en el dominio, esta clave puede ser usada para una sola conexión (virtual o de circuitos) o mantenerla durante algún tiempo.

El Esquema 3 es una variante del anterior esquema, que no altera en lo fundamental sus características de seguridad. Consiste en que en el paso 2, en lugar de enviar la clave de sesión a ambas partes, el CGC le envía solamente a la entidad A un mensaje dividido en dos partes: la primera parte contiene la clave de sesión cifrada de forma que sólo A pueda leerla y la segunda contiene la misma clave de sesión de forma que sólo B pueda leerla. En el paso 3, la entidad A, además de cifrar el mensaje con la clave de sesión que ha obtenido, le pasa a B la pieza de información recibida del CGC que contiene la clave de sesión cifrada con la clave secreta de B. Al establecerse la conexión de A con B, ésta debe reconocer esa especie de *credencial* que A le presenta y descifrar la clave que va a permitir el establecimiento de la comunicación segura.

Desde un punto de vista estrictamente telemático, en el caso de que los tres esquemas representen comunicaciones telefónicas, el primero de ellos, el que sigue GSM, tiene la ventaja de que unifica el centro de conmutación telefónica con el centro de gestión de claves, aprovechando la misma llamada para resolver tanto el problema del encaminamiento como el de la securización de la comunicación. En los otros dos esquemas, por contra, existe un primer paso para la obtención de la clave de sesión y un segundo paso para el establecimiento de la conexión, lo que puede redundar en un aumento del tráfico y un peligro de congestión.

En el caso de que se trate de redes de conmutación de paquetes el CGC no tiene por qué coincidir con un nodo de conmutación, por lo que el esquema de funcionamiento representado en la Figura 6.3 y el Esquema 3 presentan, en comparación con el escenario representado en la Figura 6.2, algo más de independencia en cuanto al funcionamiento de las entidades.

En general, estos modelos centralizados tienen el gravísimo inconveniente de que si el elemento central es vulnerado por un atacante, toda la seguridad del sistema se viene abajo. Además, si se detecta esta situación, las tareas necesarias para restablecer de nuevo el esquema de confianza pueden ser muy laboriosas si el número de entidades integradas en el dominio es grande y están geográficamente dispersas. Esto es así porque los mecanismos de distribución inicial de las claves «permanentes» mediante las cuales las entidades están suscritas al sistema deben basarse en métodos no telemáticos, para evitar así una posible vulneración de la seguridad.

No obstante, en lo fundamental, en los tres modelos tanto A como B están obligadas a confiar plenamente en el CGC y a admitir que si su gestor lo considera oportuno, puede descifrar las comunicaciones que intercepte. En los dominios que hemos definido como de Seguridad Cívica este tipo de soluciones son totalmente inaceptables. Esto es así porque el elemento central nos protege de los posibles intrusos, pero ¿quién nos protege del elemento central?

CLAVES DE SESIÓN O CLAVES DE TRÁFICO

En los escenarios de comunicación con una estructura del tipo de la reflejada en la Figura 6.3 se puede observar una notable diferencia entre los dos tipos de claves puestas en juego. Unas, las que las entidades comunicantes mantienen compartidas con el Centro de Gestión de Claves, tienen un carácter más estable y permanente, más **ligadas a la identidad** de la entidad. Podríamos considerarlas como *claves de suscripción* al sistema y su vulneración tendría consecuencias graves durante toda la vigencia del esquema de comunicación. Es más, si habiendo sido comprometidas se procediese a su renovación, nunca más en la vida sería necesario recuperar el valor de la que ha sido revocada.

En cambio, las otras, las que genera el CGC para el establecimiento de una comunicación entre A y B, tienen un interés **coyuntural**, siendo válidas sólo para unas cuantas sesiones o transacciones. Podríamos llamarlas *claves de sesión* o *claves de tráfico* para reflejar el uso que en realidad tienen.

En este primitivo y poco robusto esquema aparece ya un concepto de sumo interés y trascendencia: la de una clave cifrada con otra clave. La clave cifrada es una clave coyuntural de sesión y la clave con la que se cifra es una clave de las de suscripción. En la variante representada por el Esquema 3, a la pieza de información que recibe A cifrada con la clave de suscripción de B le hemos asignado el nombre de *credencial* porque es algo que A presentará a B, cuya validez no se basa en la confianza que B tenga depositada en A sino en el contenido mismo de esa pieza de información.

Como se recordará, ya habíamos introducido el concepto de *clave de sesión* en el capítulo anterior también con motivo del cifrado de mensajes con criptosistemas simétricos. Los esquemas de *concertación de clave* y de *envoltura digital* representan, en realidad, dos escenarios más de **distribución en red de claves secretas**. Lo que ocurre es que en estos casos la distribución se realiza apoyándose en criptosistemas de clave pública y la fortaleza de la comunicación recae directamente en la mejor o peor

manera de distribuir estas claves públicas a través de la red, situación que se discutirá más adelante, en el epígrafe 6.3.

Una cosa sí es conveniente matizar ahora en relación con la necesidad de conservar copias de las claves que ya han quedado obsoletas. Más arriba se ha hecho notar que las claves de valor más permanente, las claves de suscripción, si han sido revocadas por sospecha (o certeza) de que han sido vulneradas, nunca más en la vida del sistema será necesario volver a recuperar su valor, por lo que no será necesario guardarlas en archivo alguno.

No es este el caso en lo que atañe a las claves de sesión. Aunque en el capítulo anterior, al describir el mecanismo de envoltura digital, se dijo que la clave de sesión era «de usar y tirar» para enfatizar el hecho de que a una sesión nueva le correspondía una clave nueva, nada más lejos de lo conveniente que tomar esa frase al pie de la letra: las claves de sesión (en todos los esquemas que en este apartado hemos considerado) necesitan ser guardadas si se desea almacenar, protegidos criptográficamente, los mensajes que han sido cifrados con ellas. Sólo de esta forma dichos mensajes serán legibles más adelante.

En cuanto a la disminución de seguridad que se produce, siempre es mucho menos grave que alguien malintencionado descubra una clave de sesión que si lo que ocurre es la pérdida de protección para la clave de suscripción (o la clave privada si la de sesión se obtiene mediante un criptosistema simétrico). En el primer caso afectará solamente a la seguridad de un mensaje concreto, mientras que en el segundo afectaría a todos los mensajes que se recibiesen durante el tiempo que la clave estuviese intervenida inadvertidamente. Esta es la razón por la que las medidas de protección deben ser distintas dependiendo del tipo de clave de que se trate.

A modo de conclusión parcial, podemos comprobar que la distribución de claves secretas en redes de área extensa donde sólo existan criptosistemas simétricos sólo es posible mediante la presencia de elementos centrales. Asimismo, se comprueba que la existencia de estos elementos representa una amenaza insalvable para la privacidad de los participantes. La única solución está en la eliminación de esta centralización: eso es lo que se consigue con los criptosistemas de clave pública.

6.2. DEPÓSITO Y RECUPERACIÓN DE CLAVES (KEY ESCROW)

En realidad el nombre que más ajustadamente corresponde a estos sistemas es el de **depósito de la clave secreta para recuperación (captura) de la clave de sesión**. En efecto, de lo que se trata es de que la clave secreta de un usuario esté depositada en algún lugar custodiado por la autoridad competente, para que, llegado el momento, esa misma autoridad permita, cuando se cumplan ciertos requisitos, que los poderes públicos puedan, sin conocimiento del interesado, averiguar el valor de la clave de sesión que está utilizando en una comunicación concertada con otro usuario cualquiera.

Cuando los sistemas eran de sólo clave secreta con un elemento central de gestión de claves (usamos el pretérito imperfecto, aunque GSM mantenga todavía tan imperfecto planteamiento), lo que en párrafos anteriores hemos estado considerando como un inconveniente, era una ventaja para las autoridades policiales y todas aquellas que se consideran, a su manera, defensoras de la seguridad colectiva. La posible injerencia en la comunicación mediante la intervención del sistema central es factible con el solo hecho de conseguir una autorización judicial que permita a la autoridad correspondiente acceder a la comunicación a través del proveedor del servicio. No obstante, con el

advenimiento de las claves de sesión concertadas entre dos usuarios apoyándose en los muy robustos sistemas de clave pública, esa intervención no es posible, ni por orden judicial, ni *motu proprio*, ni con ayuda o sin ella del proveedor del servicio: la única forma sería romper la fortaleza del criptosistema asimétrico. Por eso es por lo que se persiguen iniciativas del estilo del *Key Escrow*.

TÉCNICAS PARA EL DEPÓSITO Y CAPTURA DE CLAVES

Dejemos para más adelante la discusión acerca de la bondad o no del planteamiento y centrémonos en la componente técnica del asunto:

Aunque ha habido varias propuestas al respecto, la más importante de todas fue la que se planteó a mediados de los noventa en Estados Unidos con el nombre genérico de *Escrowed Encryption Standard* [Denn94], más conocida como la iniciativa *Clipper Chip*, por basar su aplicación práctica en la existencia de un chip o circuito integrado que debería ser insertado obligatoriamente en todos los dispositivos de comunicación que cifrasen la información, ya fuesen computadores o simples teléfonos. A pesar de los cuantiosos fondos que en ella se invirtieron, esta iniciativa no prosperó por la oposición interna que se suscitó. No obstante, a ella nos referiremos para fijar las características técnicas más relevantes de este tipo de planteamientos.

En las Figuras 6.4 a) y 6.4 b) se recoge de forma esquemática lo más sustancial de la iniciativa, obviando muchos detalles para así centrarnos en lo más importante. En primer lugar se procede, Figura 6 a), a la inicialización del chip que será obligatoriamente introducido en el dispositivo de comunicación. Se supone la existencia de dos agencias o instituciones a las que se les ha asignado legalmente, y de forma compartida, el depósito y custodia de las claves secretas de todos los usuarios o entidades comunicantes con objeto de que éstas puedan ser conocidas cuando la autoridad competente lo decida. Para la descripción que estamos haciendo, vamos a denominarlas *Agente1* y *Agente2*. En este proceso inicial concurren dos representantes autorizados de los dos agentes, cada uno de los cuales aporta un número aleatorio que actúa como entrada de un generador de claves simétricas. Como resultado de esta doble operación se generan las claves k_1 y k_2 que van a constituir el verdadero depósito del que venimos hablando.

Realizando una sencilla operación XOR con las dos claves k_1 y k_2 se obtiene la clave secreta del usuario, k_u , que se almacena directamente en el chip junto con un identificador de usuario, al que hemos denotado como *IDu*. Para terminar, el *Agente1* se lleva un registro con «su» parte de la clave, cifrándola previamente con una clave secreta, k_{A1} , que sólo él conoce, con objeto de garantizar al máximo su custodia en el proceso de transporte y almacenamiento en sus bases de datos. Para que la clave cifrada quede adecuadamente registrada la asocia con el correspondiente identificador de usuario. El *Agente2* procede de forma similar. En esta como en las restantes operaciones de cifrado llevadas a cabo en el chip, el algoritmo utilizado es el Skipjack (algoritmo secreto del que hicimos una breve reseña en el epígrafe 4.5). En la propuesta de estándar aparecen otras informaciones que aquí hemos obviado para ganar en claridad.

El chip de usuario, supongamos que se trata del chip de la *Usuaría A* que aparece en la Figura 6.4 b), se introduce en su dispositivo de comunicación, computador o teléfono, y actuará automáticamente cuando se inicie un proceso de comunicación segura. Se trata de un chip resistente ante ataques, *tamper-resistant*, de forma que no existen medios razonables para poder acceder ilícitamente a su contenido y mucho

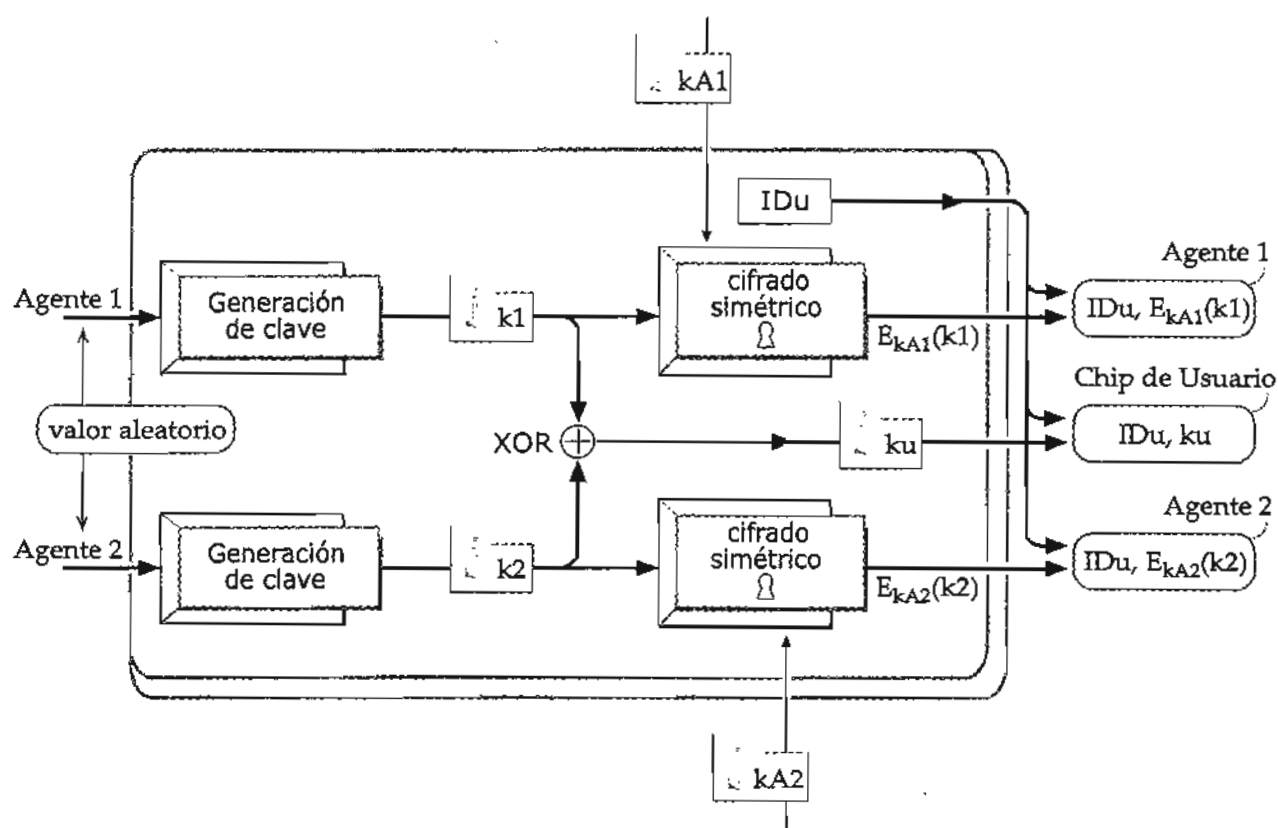


Figura 6.4. a) Depósito de la clave secreta: inicialización del chip y de los registros.

menos para alterarlo. Para empezar (operación número 1 de la figura) la *Usuaría A*, para poderse comunicar de forma confidencial, evitando que una tercera parte pueda acceder al contenido de la información que va a intercambiar con el *Usuario B*, ha establecido, de forma concertada con él, una clave de sesión, k_s , con la que cifrar los mensajes.

A continuación (operación 2) emite el mensaje m cifrado con esta clave de sesión que podrá ser adecuadamente descifrado por el *Usuario B*, y solamente por él. Además, en paralelo con esta información cifrada, el chip insertado en el dispositivo de la *Usuaría A* transmite de forma autónoma una información consistente en la identidad de la usuaria (que, recordemos, está grabada en el chip) y la clave de sesión cifrada con la clave secreta de la usuaria. Como su propio nombre indica, esta clave es secreta, por lo que nadie podrá conocer la clave de sesión que están utilizando. Esto es así siempre y cuando la *Usuaría A* sea buena chica y no despierte sospechas en algún celoso vigilante del orden.

Caso de darse esta sospecha, con permiso de la autoridad competente y dado que el chip no lo impide (sino que lo facilita) puede obtenerse la clave de sesión que está siendo usada en la comunicación. Para ello se le exige al proveedor del servicio que derive hacia la Oficina de Captura la información etiquetada con el número 2 en la Figura 6.4 b). En dicha Oficina, a partir de la identidad de la usuaria, IDu , puede reunirse la información almacenada en los dos agentes de depósito de forma que, mediante una sencilla operación XOR de las claves $k1$ y $k2$, quede recompuesta la clave secreta ku . A partir de aquí todo es coser y cantar: primero se obtiene k_s descifrando $E_{k_u}(k_s)$ y luego se tiene acceso al mensaje m que había sido cifrado con ella.

En la propuesta del estándar *Key Escrow*, el chip transmite más información de la que, por simplificar, se ha recogido en la figura. Esa información va cifrada a su vez

En efecto, en el mundo empresarial puede tener aplicación porque puede darse el caso de que se pierda la clave con la que estén cifrados muchos documentos almacenados. Esto puede producirse por algún error de procedimiento o porque un determinado empleado, que era el poseedor de la clave, se dé de baja en la compañía o quiera actuar con deslealtad. Además, las nuevas iniciativas no se centran exclusivamente en la clave de sesión, como es el caso de la Key Escrow, sino que se orientan hacia la posibilidad de recuperar la clave privada de un usuario en un criptosistema de clave pública (con o sin su conocimiento y autorización).

En el amplísimo debate que en su momento se suscitó en Estados Unidos con motivo de la iniciativa Clipper Chip intervinieron, tanto a favor como en contra, personas de muy alta cualificación. Desde un punto de vista exclusivamente técnico, científicos tan reconocidos como Diffie, Rivest y Schneier argumentaron de forma colectiva en contra de su implantación, mientras que el número de los que abogaron por su conveniencia (Dorothy Denning, entre otros) fue más reducido y su opinión tuvo un menor impacto.

De forma resumida, las objeciones que, desde un punto de vista tecnológico, pueden achacarse a las propuestas de recuperación de claves, podrían concretarse en las siguientes:

- La mayoría de los algoritmos y protecciones han sido diseñados pensando en el exclusivo control del usuario sobre los medios necesarios para descifrar los datos. Las propuestas que se realicen al margen de esta concepción pueden tener inconsistencias. De hecho en los proyectos piloto de esquemas de depósito y recuperación llevados a cabo a pequeña escala se han encontrado problemas de interoperabilidad con soluciones y estándares existentes o emergentes.
- Una infraestructura global, capaz de abarcar a todos los dispositivos de comunicación que utilicen mensajes cifrados, es previsible que sea extraordinariamente compleja y costosa. Los problemas derivados de su implantación en países diversos, la distribución de agentes y la concertación de políticas de uso pueden constituir un problema técnico de gran envergadura y muy difícil solución. Desarrollar a gran escala un sistema de estas características requiere de una experiencia y competencias que van más allá de las que actualmente se poseen y pudiera introducir riesgos y costes inaceptables.
- La mera existencia de bases de datos centralizadas conteniendo información tan sensible como la relacionada con las claves secretas de los usuarios representa un riesgo muy importante, ya que cualquier problema de robo o ataque con éxito a uno de estos sistemas tendría consecuencias desastrosas y de muy difícil restauración. (Las claves secretas de usuario, como antes hemos discutido, tienen una fuerte relación con su identidad en la red.)
- Aquellos a los que nominalmente se dirige la propuesta (crimen organizado, terroristas y servicios de espionaje extranjeros) pueden evadir el uso de estos servicios. Ya se ha comentado lo sencillo que es crear canales secretos de comunicación a través de criptosistemas simétricos secretos, a través de esteganografía, o mediante el uso combinado de ambas.
- Los criminales que deseen aparentar relaciones personales o actividades comerciales normales pueden enmascarar sus datos con otra aplicación de cifrado u ocultación antes de que sean recifrados con la clave depositada.

Resumiendo, los sistemas de recuperación de claves pueden tener utilidad en entornos empresariales reducidos, pero son, por su propia concepción, más costosos, más difíciles de utilizar y menos seguros que los sistemas convencionales basados en

el uso combinado de criptografía simétrica y de clave pública. A gran escala, supone meter una cuña a todos los usuarios en todas las comunicaciones que establezcan todos los días y a todas las horas para cazar a unos malos que tienen multitud de medios para evadir esas trampas (todos sabemos que los malos son muy listos).

SITUACIÓN DE LAS PROPUESTAS DE RECUPERACIÓN DE CLAVES

Lo que algunos se malician es que el verdadero interés de los que proponen los sistemas de recuperación de claves es crear mecanismos de vigilancia colectiva que permitan a los poderes públicos seguir haciendo en las redes seguras lo que han venido haciendo hasta la fecha en las convencionales (tienen tanto apego a estas prácticas de escucha como el que tienen los traumatólogos por las radiografías). Aunque el argumento que respalda este tipo de actuaciones es el de defensa de la comunidad, es sobradamente conocido el uso y abuso que se ha hecho de esta posibilidad y la ausencia de garantías jurídicas con las que, en multitud de ocasiones, es llevado a cabo el proceso.

Por contra, existen grupos que preconizan su implantación y mantienen organizaciones, como la norteamericana Key Recovery Alliance, un «lobby» que aboga por la implantación de estos servicios a nivel empresarial como parte del posterior desarrollo de una infraestructura global.

En el ámbito internacional las propuestas en esta dirección han ido decayendo en todos los foros desde su promoción a mediados de la década de los noventa, tanto en Estados Unidos como en Europa y otras partes del mundo. En el caso de Europa, dentro de un foro denominado Audiencias de Copenhague, auspiciado en 1998 por la Comisión Europea para el análisis de temas relacionados con el cifrado de los datos, las conclusiones acerca de los sistemas de depósito y recuperación de claves son marcadamente críticas y desaconsejan abiertamente su utilización. En la misma dirección abunda el documento elaborado por la Comisión para armonizar las políticas de firma digital en el que considera que estos sistemas y las TTPs asociadas conllevan un cierto número de consideraciones críticas que necesitan ser cuidadosamente atendidas, que los sistemas pueden ser fácilmente burlados y que los costes pueden ser muy altos.

Incluso en Francia, país en el que desde siempre se ha dado mucho peso al control gubernamental de las comunicaciones, han abandonado las iniciativas para la implantación de sistemas de depósito. Ello es significativo porque en ese país hasta poco antes de acabar el siglo estaba prohibido encriptar mensajes en la red sin previa autorización.

Más adelante, cuando se aborde el tema de las políticas de seguridad que pueden adoptarse en un determinado dominio, se volverá a insistir sobre las escasas ventajas y los muchos inconvenientes de este modelo de gestión de claves.

6.3. DISTRIBUCIÓN EN RED DE CLAVES PÚBLICAS

Cuando las distintas entidades de comunicación (usuarios, servidores, TTPs, agentes telemáticos, etc.) hacen uso de criptosistemas de clave pública, el esquema general de funcionamiento es el que se muestra en la Figura 6.5, en la que se ha representado un *Servidor de información sobre Claves Públicas* que es un sistema que trabaja fuera de línea (*off line*). Esto es, que se limita a proporcionar, a la entidad que lo solicite, la clave pública de las restantes entidades pertenecientes al dominio de seguridad de que

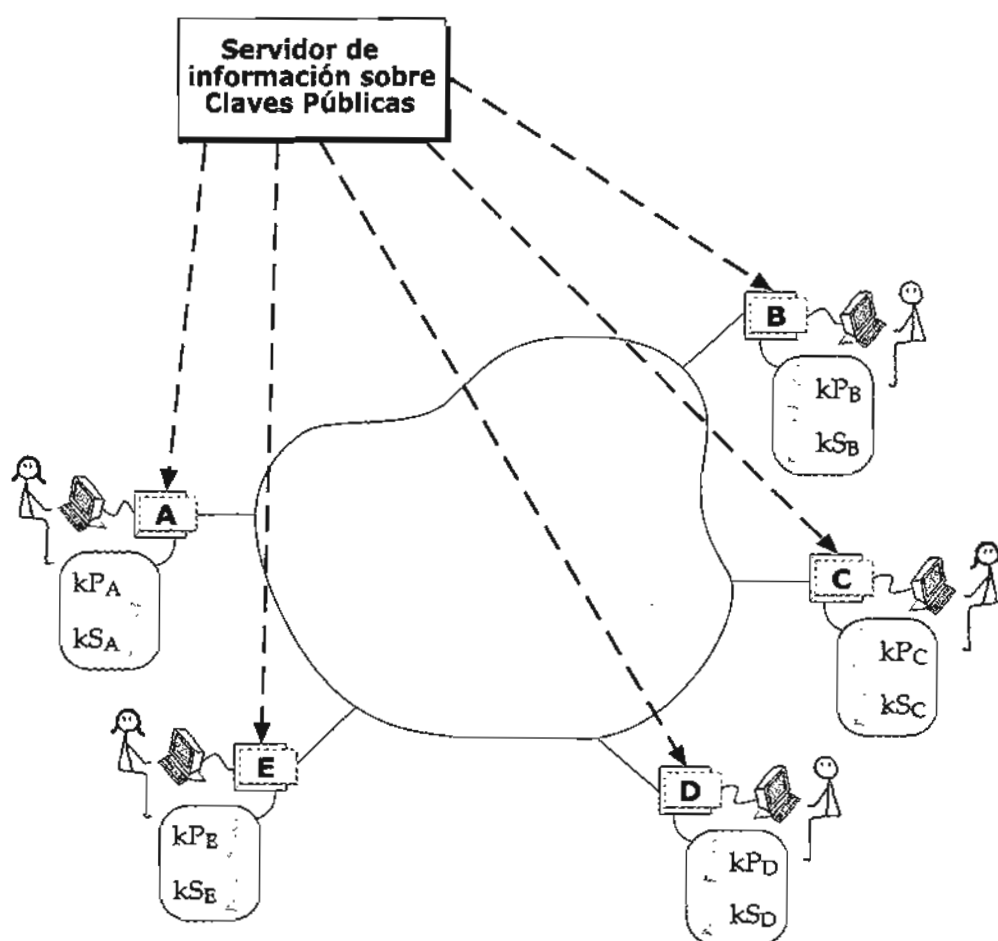


Figura 6.5. Esquema general de distribución de claves públicas.

se trate, pero sin tener que intervenir directamente en el momento del establecimiento de la comunicación entre dos o más entidades.

En una primera simplificación (muy distante de lo que debe ocurrir en un entorno de red real, como más adelante veremos), vamos a suponer que cada entidad mantiene solamente un par de claves: la privada que mantiene en secreto y la pública que es puesta en común conocimiento de las restantes entidades a través del Servidor de información sobre Claves Públicas.

Para introducir los problemas que van surgiendo a medida que el dominio de seguridad crece en número de entidades y complejidad organizativa, vamos a ir contemplando unos cuantos ejemplos de dificultad creciente. Además, a la chita callando, vamos a ir introduciendo algunos conceptos de *política de seguridad* directamente relacionados con la ya referida complejidad de los dominios que se vayan contemplando.

DOMINIO HOMOGÉNEO CON POCOS USUARIOS

En primer lugar vamos a considerar el caso de un número no muy grande de entidades pertenecientes todas a un dominio de seguridad muy homogéneo, es decir, que todas las entidades pertenecen a un mismo entorno organizativo con idénticas restricciones y necesidades de comunicación. En un entorno así es fácil pensar que exista una *Autoridad de Seguridad* claramente detectable en la que, además, todos los usuarios confían. Es decir, existirá una persona con manifiesta autoridad en ese entorno que

será la que dicte las normas de seguridad y en cuya imparcialidad, de grado o por fuerza, todas las entidades creen.

Como ejemplo, vamos a suponer un conjunto de quince o veinte alumnas y alumnos pertenecientes a un mismo grupo de laboratorio de una asignatura en la que se están realizando prácticas sobre protocolos de seguridad. Es razonable suponer que sea el profesor encargado de la docencia en ese grupo el que se autoasigne el papel de Autoridad de Seguridad en ese dominio y que sea él quien defina unas normas elementales en las que se recoja qué cosas se pueden hacer y cuáles no. En la Figura 6.6 se ha representado un esquema simplificado de este entorno. La entidad C se ha representado en forma de robot para querer indicar que no se trata de un usuario humano sino de alguna máquina o agente telemático que funciona automáticamente, sin intervención humana permanente: puede ser una impresora, una base de datos, un sitio web, etc., y se supone que también participa en el escenario de comunicación, por lo que puede necesitar tener conocimiento de las claves públicas de las restantes entidades de comunicación, por ejemplo la de la alumna A si ésta le envía un mensaje firmado.

Conforme a un algoritmo de clave pública, por ejemplo RSA, el profesor puede generar un par de claves para cada uno de los alumnos usuarios de la red en ese dominio concreto. Posteriormente, borrará todas y cada una de las claves privadas, excepto la suya, y repartirá a cada alumno un disquete conteniendo la clave privada del interesado más otro disquete, idéntico para todos, con las claves públicas de todos los usuarios. Además, el profesor se compromete a no repetir las claves, entregarle a cada uno un par diferenciado, y borrar todas las claves privadas una vez que haya realizado la distribución. Como el alumno confía bastante en el profesor (para esa actividad concreta) va a aceptar que, verdaderamente, a partir de recibir el disquete, va a ser él (el alumno), y sólo él, el único depositario de su clave privada y, por tanto, el único responsable de lo que se haga con ella.

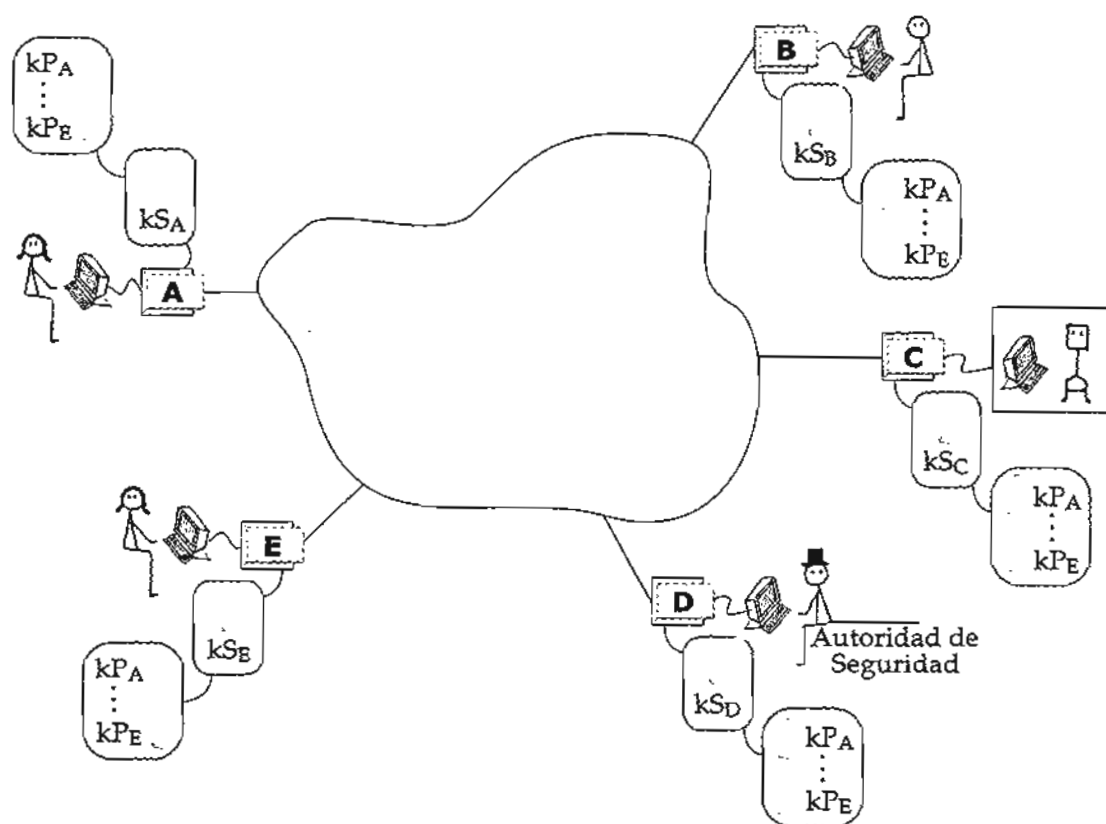


Figura 6.6. Escenario de comunicación con pocos usuarios.

En este entorno concreto no resulta nada chocante que el alumno confíe en que el profesor va a cumplir con lo prometido. En realidad, en cualquier escenario de seguridad que podamos imaginar siempre tendrá que existir algo de confianza en alguien. No obstante, una norma maestra de seguridad es que *cuanto menos se confíe en terceros, tanto mejor*. Por eso, más adelante volveremos al problema de en quién confiar para la generación de las claves.

Hemos querido suponer que el profesor entrega dos disquetes a cada alumno, y no uno solo que reúna toda la información que afecte al interesado, para hacer hincapié en el distinto comportamiento que el receptor debe seguir en el uso de cada uno de ellos. El disquete que recibe con las claves públicas es meramente informativo y los datos que contiene son de público conocimiento, por lo que no tiene ninguna importancia que alguien pueda conocer o copiar su contenido. Sí tiene que tener cuidado con su integridad, porque le servirá para poder enviar información confidencial al resto de los usuarios del entorno.

En cuanto al disquete específico que tiene grabada la clave privada, esa pieza de información y el dispositivo que la contiene son para quien los recibe un marchamo de validez, un testigo, una prenda de garantía acerca del verdadero valor de su clave secreta: representa parte de su identidad en la red. Diremos que ese disquete es un **testigo de seguridad** (*token*). En la medida en que lo custodie adecuadamente y procure que nadie pueda copiarlo ni modificarlo, tendrá garantizado que nadie podrá realizar operaciones en su nombre. Entre otras cosas, deberá tener garantías de que en la memoria del terminal de laboratorio que use para sus prácticas no quede huella alguna del valor de su clave privada.

Otra cuestión importante y no poco complicada que ya aparece incluso en un entorno tan sumamente simplificado como éste, es la relacionada con las revocaciones. Cuando un alumno considere que, por algún motivo, su clave ha sido comprometida, bien porque tenga evidencias, o simplemente sospechas, de que alguien ha podido copiar su clave privada, deberá comunicarlo inmediatamente al profesor para que dé de baja esa clave y le genere un par nuevo. El profesor tendrá que ingeniárselas para definir un procedimiento para informar al resto de los usuarios del valor de la nueva clave pública y la fecha de revocación de la antigua. Dejaremos para más adelante la discusión, nada baladí, de cuáles pueden ser las normas que se dicten (al definir la Política de Seguridad) acerca de la validez de lo firmado o encriptado con claves revocadas. (En el apartado 6.1 ya se discutió algo parecido al comentar el tema de las claves de sesión y de suscripción.)

En este caso, es el profesor el que actúa como servidor de información sobre claves públicas. Como las claves públicas son de común conocimiento, podríamos tener la tentación de implementar un Servidor automático, accesible a través de la red, que gobernase una base de datos que contuviese todas las claves públicas del dominio. Una solución como esta daría al traste con la seguridad del entorno, a no ser que implementásemos servicios de autenticación relativamente complejos. Por ejemplo, aunque la base de datos estuviese protegida para que nadie, excepto su gestor, pudiese modificar los datos en ella contenidos, cuando la alumna A trate de conectarse con el Servidor para obtener la clave pública de B (a quien desea mandar un mensaje confidencial), un intruso I que se interpusiese en el camino podría engañar a A haciéndose pasar por el Servidor, y entregarle su propia clave pública en lugar de la de B. A partir de ese momento, los mensajes confidenciales que A envíe a B sólo podrán ser leídos por I. En el intercambio de mensajes firmados entre los alumnos participantes en la comunicación podría ocurrir otro tanto de lo mismo: alguien puede modificar el valor de una clave pública y firmar falsamente en nombre de la entidad atacada.

EL CERTIFICADO

Como acabamos de ver, el talón de Aquiles de los criptosistemas de clave pública consiste en que si a alguien se le engaña acerca del valor de la clave pública de otro usuario, la seguridad del tinglado se viene abajo. En realidad, la clave pública sólo hay que darla a conocer a los miembros del dominio de seguridad de que se trate, pero como su conocimiento no tiene nada de reservado, no importa que sea conocida por quienes no tengan nada que ver con ese entorno. Lo que sí hay que garantizar de algún modo es su **validez**.

La solución a este problema consiste en garantizar la propiedad y validez de la clave pública mediante la generación de un *certificado* de la clave pública firmado digitalmente por una *Autoridad de Certificación*, CA (*Certification Authority*, en inglés). La CA es una tercera parte de confianza (TTP) en la que confían los participantes en la comunicación que son miembros del dominio de seguridad de que se trate.

El certificado digital es una pieza de información en la que se asocia el nombre de una entidad con su clave pública durante un periodo de validez. En la Figura 6.7 se presenta la estructura de un certificado de complejidad mínima. Este certificado simplificado de la entidad A consta de dos partes: la primera contiene los DATOS en los que se aporta la información necesaria, y la segunda parte es la firma de esos datos con la clave privada de la CA. La nomenclatura que vamos a utilizar, conforme a la que se propone en la norma ISO/IEC 9594-8 [ISO 9594], es la siguiente: CA<<A>> significa «certificado de A generado o expedido por la CA». Genéricamente, si el nombre de la CA es, por ejemplo, X, el certificado de A generado por X sería:

X<<A>>

El certificado representado en la Figura 6.7 es una simplificación de lo que se utiliza en la realidad; más adelante estudiaremos los distintos formatos de certificados que se han definido. Como claramente se indica en la figura, los campos que como mínimo se le han supuesto a este certificado son, además de la firma, la identificación de la entidad a nombre de la cual se emite el certificado, el valor de la clave pública que se está certificando, la identificación de la CA que emite el certificado y el periodo de validez durante el cual el certificado es válido.

Es importante hacer notar que el certificado es una pieza de información segura en sí misma que, en principio, puede ser conocida por todos sin ningún género de restricciones. Su fortaleza la obtiene de la firma de la CA correspondiente y de la confianza que el receptor del certificado tiene en esa CA. Por tanto, no importa cómo

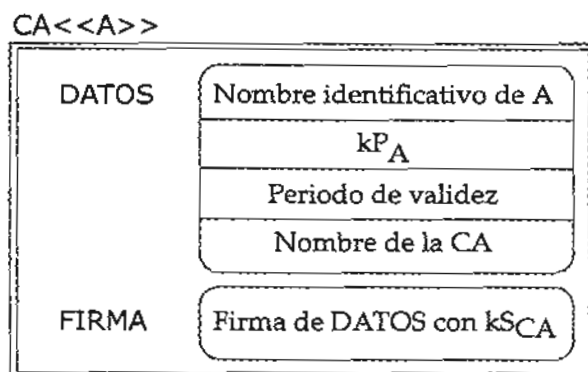


Figura 6.7. Formato simplificado de un certificado.

ni dónde esté depositado, y pueden ser distribuidos a través de canales o medios inseguros sin necesidad de establecer protecciones de ningún tipo para preservar su integridad o su autenticidad.

Es de capital importancia que la CA, antes de emitir el certificado del usuario A adquiriera la certeza suficiente acerca de la identidad del solicitante. Estos procedimientos estarán especificados en la política de seguridad que se defina, como veremos más adelante, aunque es conveniente adelantar que lo más aconsejable es hacerlo usando métodos tradicionales, no telemáticos, porque si la generación del certificado es una pieza clave para empezar a confiar en la autenticación de los usuarios conectados a la red, malamente se podrá garantizar esa autenticidad si no se comprueba adecuadamente su identidad antes de proceder a generarlo.

A la hora de plantear cuáles deben ser las funciones de una CA, además de la que justifica su existencia (la generación de certificados) suelen contemplarse otros aspectos como puede ser la generación de claves, el registro de las entidades usuarias adscritas a la CA o el almacenamiento de los certificados generados. A falta de una discusión más pormenorizada que se aplaza para más adelante, cabe decir ahora que la tendencia actual es crear unas TTPs específicas para el registro, las denominadas *Autoridades de Registro* o *RAs*, y dejar el tema del almacenamiento y distribución de certificados fuera de las responsabilidades de la CA.

Como resumen, cabe decir que la confianza en el certificado proviene de dos causas:

- a) La confianza en la CA. Todas las entidades pertenecientes a un determinado dominio se fían de que cuando expide un certificado a la entidad A, antes ha comprobado adecuadamente la identidad de A y el valor de su clave pública. Confían, además, en que no les va a engañar, bien porque han participado de la decisión de quien administra la CA, bien porque tienen que aceptarlo si desean entrar a formar parte de un determinado dominio.
- b) Todos los miembros del dominio de seguridad conocen la clave pública de la CA y están seguros de que es válida. Gracias a eso, al verificar la firma de la CA están seguros de la validez de la clave pública de la entidad certificada.

UNA SOLA CA EN EL DOMINIO

Volviendo al ejemplo de los alumnos del grupo de laboratorio, ahora podemos sustituir el escenario de la Figura 6.6 por el representado en la Figura 6.8, donde se ha particularizado el genérico Servidor de Información sobre claves públicas por una CA que guarda los certificados en un Servidor de Certificados, que puede ser un servidor convencional que gobierna una base de datos de acceso remoto que sólo admite, a través de la red, operaciones de lectura de los certificados allí almacenados. Las entidades pueden ser las mismas, pero ahora el profesor ya no constituye el Servicio de Información sobre claves públicas. Se ha supuesto que sigue siendo la Autoridad de Seguridad que dicta las normas, pero aparece como una entidad de comunicación más, sujeto a la necesidad de acceder al Servidor para recuperar los certificados de los demás participantes. Si acaso podemos suponer que sigue siendo quien genera las claves y que es el gestor de la CA, cargo de confianza extremadamente delicado.

Para no complicar las cosas, podemos suponer que la generación de claves y la distribución de la clave privada mediante un testigo de seguridad (el disquete) con

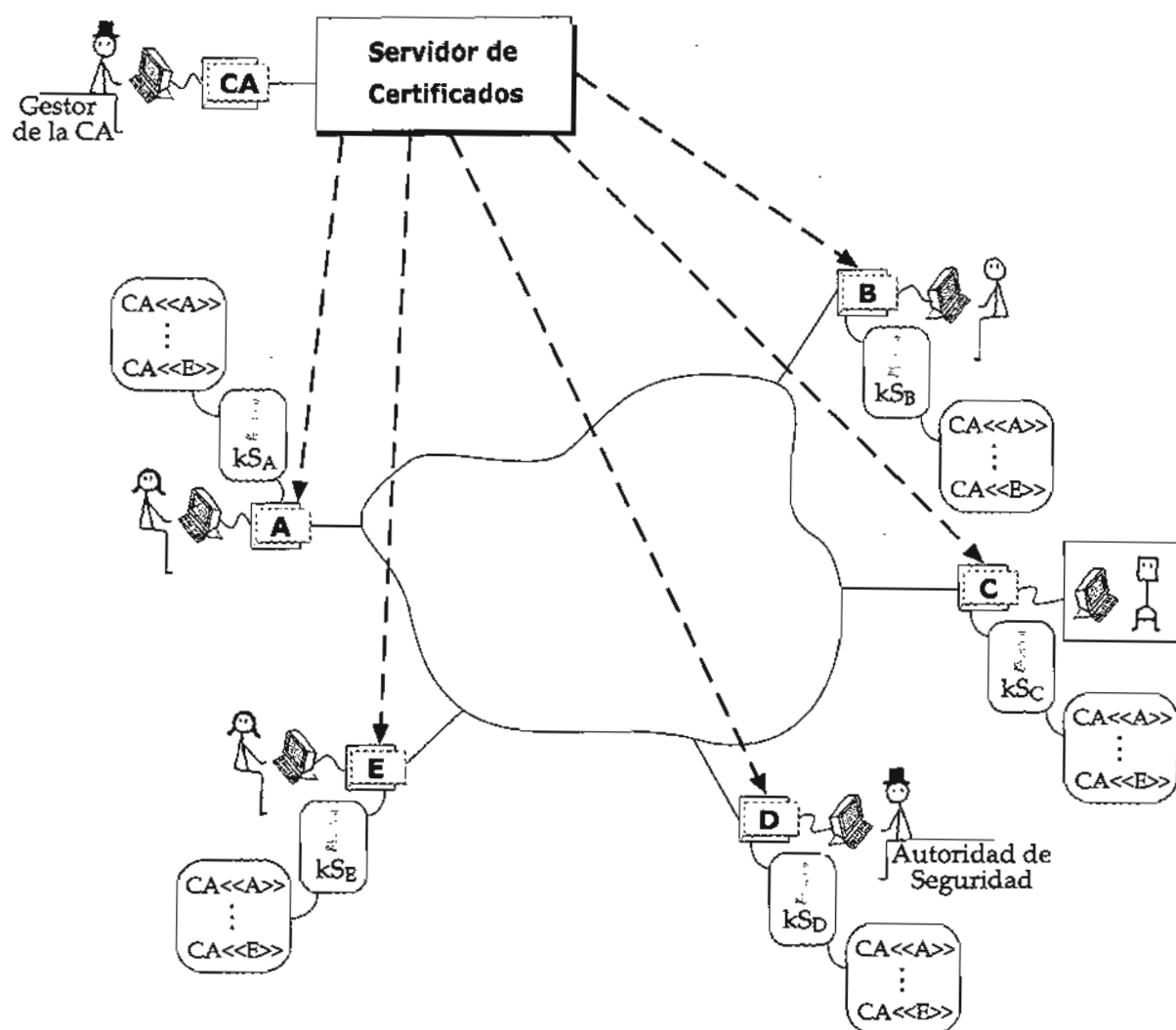


Figura 6.8. Escenario de comunicación con una sola CA.

marchamo de validez se realiza de la misma manera que se explicó en el caso anterior. Ahora el comportamiento de las entidades usuarias se ha simplificado notablemente. Queda, no obstante, un cabo suelto: cómo publicar y poner en conocimiento de todos el valor de la clave pública de la CA. Esto es importante, porque si alguien consigue engañar al grupo y les hace creer que su clave pública es la de la CA, la seguridad de todo el sistema se viene abajo. Una forma sencilla de curarse en salud es buscar un procedimiento extratelemático, que puede consistir en que, por ejemplo, la clave tenga validez durante un periodo de tiempo determinado y esté publicada en un tablón o boletín visible y accesible para todos.

Si la alumna A quiere enviar a B un mensaje confidencial, para cifrar de forma fiable el mensaje con la clave pública de B, sólo tiene que recuperar el certificado de B, extraer de él kP_B , comprobar la validez de la clave verificando (mediante kP_{CA}) la firma de la CA y configurar el mecanismo de seguridad correspondiente para integrarlo en el protocolo que esté manejando, un correo electrónico seguro, por ejemplo. Algo similar tendría que hacer B si recibe un mensaje de correo firmado por A: accedería al Servidor y recuperaría el certificado de A, obtendría kP_A , comprobaría la validez de la clave verificando mediante kP_{CA} la firma del certificado y verificaría la firma del mensaje mediante kP_A .

Hemos visto cómo con la inclusión de una CA (Figura 6.8) se puede mejorar de forma drástica la infraestructura de seguridad del entorno de comunicación homogé-

neo y con pocos usuarios que se presentó inicialmente en la Figura 6.6, en el que el profesor actuaba como Servidor de Información sobre claves públicas.

Usando una sola CA el dominio de seguridad puede ser mucho más amplio y heterogéneo que éste del grupo de laboratorio que estamos contemplando. Por ejemplo, podemos suponer un entorno constituido por una Escuela o Facultad completa, con sus miles de alumnos, múltiples servidores conectados en red, y un buen número de profesores y personal de administración y servicios. Es fácil entender que una situación como esta sería prácticamente inviable sin el concurso de los certificados.

(Ya de por sí es una simplificación notable el pensar que una organización de esta complejidad pueda funcionar con un esquema tan simplificado como el de la Figura 6.8, con un solo par de claves y un solo certificado por usuario, pero nos puede servir a modo de ejemplo. En este entorno se van a producir comunicaciones de muy distinto tipo (convocatorias, calificaciones, información técnica o científica, etc.) que van a requerir muy distinto nivel de protección, por lo que la definición de la política de seguridad tendrá que ser mucho más completa y discriminativa que la que hemos vislumbrado hasta ahora.)

Lo que sí podemos pensar es que en este escenario más complejo, la Autoridad de Seguridad podrá coordinar a un equipo de personas entre las que se encuentre el gestor de la CA, el gestor del Servidor de certificados y los encargados de generar las claves y distribuir las claves privadas entre todos los participantes mediante testigos de seguridad, es decir, dispositivos con marchamo de validez (hemos supuesto que eran disquetes, podríamos empezar a suponer que fuesen tarjetas inteligentes).

La Figura 6.9 recoge el esquema de confianza que se presenta en un escenario en el que sólo existe una CA. A partir del conocimiento seguro de la clave pública de la CA, todos los usuarios de ella dependientes pueden empezar a utilizar los criptosistemas de clave pública con entera confianza. Dentro de las entidades que confían en la CA cabría distinguir entre:

- *Entidad suscrita a la CA.* Es aquella que posee su propio certificado emitido por esa CA.
- *Entidad usuaria del certificado o entidad que se fía del certificado (relying party¹ o certificate user en inglés).* Es aquella entidad que confía en los certificados emitidos por esa CA. Es decir, que usa y reconoce los certificados emitidos en ese dominio.

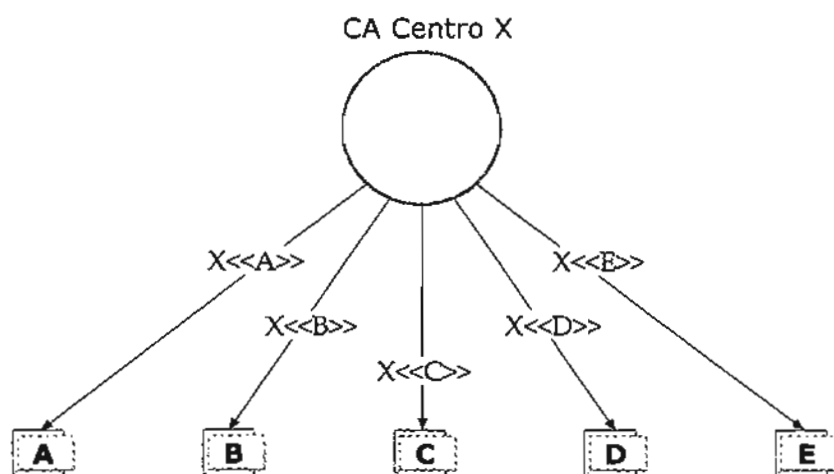


Figura 6.9. Esquema de Confianza en un escenario con una sola CA.

Es evidente que todas las entidades suscritas son entidades usuarias del certificado (*relying parties*), pero pueden existir entidades que no perteneciendo directamente al dominio de certificación de una determinada CA y no poseyendo, por tanto, ningún certificado de su clave pública generado por esa CA, sí que confíen en los certificados generados por ella y los usen en las comunicaciones seguras que establezcan con las entidades pertenecientes al mismo dominio de seguridad que la CA.

VARIAS CAS ORGANIZADAS JERÁRQUICAMENTE COOPERANDO ENTRE SÍ

Los entornos telemáticos organizados internamente de una forma fuertemente jerarquizada se prestan a una organización también jerárquica de la infraestructura de certificación. Continuando con el ejemplo que hemos utilizado en el apartado anterior, en el que contemplábamos el caso de una Escuela o Facultad y donde hemos querido suponer que con una sola CA podríamos resolver el conjunto de necesidades de securización de las comunicaciones que allí se producen, vamos a suponer ahora que quisiésemos extender el dominio de seguridad hasta alcanzar a la totalidad de la Universidad a la que dicho centro pertenece. Una primera consideración que es necesario tener en cuenta consiste en que, por razones de eficacia, el número de entidades ligadas a una sola CA debe estar limitado por posibilidades internas de gestión del sistema informático que la soporte, y no debe ser excesivamente grande, ya que, de lo contrario, el tráfico que se produciría en la recuperación de certificados haría que disminuyese la eficiencia global del esquema. Ello aconsejaría pensar en la implantación de varias CAs cuando en el dominio se constata la existencia de un número elevado de entidades de comunicación que, además, se encuentren geográficamente dispersas.

No obstante, la verdadera causa por la que es necesario prever la existencia de múltiples CAs que cooperen entre sí es que, con frecuencia, las dos entidades que deseen establecer una comunicación segura entre sí pertenecen a ámbitos organizativos diferentes. Aunque en el Capítulo 7 se analiza con cierto detalle esta temática de infraestructuras, veremos a continuación algunos ejemplos simplificados para entender globalmente el problema de la distribución de claves mediante la certificación.

En la Figura 6.10 se representa un hipotético entorno de comunicación en el que se ha supuesto la existencia de diversos centros, cada uno con una CA propia que se encuadra dentro de la política de seguridad vigente en cada uno de ellos. Podemos suponer que el Centro X1 se corresponde con el del ejemplo del apartado anterior y que en el Centro X2 existen, además de la CA principal, otras dos (X21 y X22) que dependen jerárquicamente de ella.

Si la entidad A1 del Centro X1 desea comunicar de forma segura con la entidad B2, perteneciente al Centro X2, deberá conocer con certeza la clave pública de esta entidad. Para ello deberá obtener el certificado de B2 recuperándolo de un Servidor de Certificados o pidiéndoselo directamente a B2. No obstante, el certificado que le ha sido expedido a B2 es: X21<<B2>>. Es decir, está firmado por la CA X21 del Centro X2 en la que A1 no tiene depositada ninguna confianza. Es necesario, por tanto, buscar una solución que incluya componentes en los que A1 confíe.

En este esquema, la entidad A1 debe confiar en todas las CAs que se encuentran en su línea jerárquica, esto es, además de en la CA X1, en la CA Y de la Universidad. Por la misma razón, B2 confiará en X21, X2 e Y. Para adquirir esta confianza las entidades pueden directamente conocer y confiar en las claves públicas de todas las

CA's de las que depende o, lo que es más adecuado, confiar directamente sólo en una CA y en las restantes hacerlo a través de certificados. Habría dos soluciones:

1. A1 sólo confía directamente en la clave pública de la CA Y de mayor nivel jerárquico y confía en la CA X1 mediante el certificado $Y\langle\langle X1 \rangle\rangle$ que la CA de la Universidad ha emitido a nombre de la CA del centro X1. Este sería el *certificado directo* (*forward certificate*), que coincide con el certificado que antes hemos definido.
2. Confiar sólo directamente en «su» CA más próxima y obtener la confianza en la de jerarquía superior mediante un *certificado inverso* (*reverse certificate*), esto es: $X1\langle\langle Y \rangle\rangle$. Esta solución teórica es poco recomendable y la mayoría de las infraestructuras jerárquicas que se han implementado usan exclusivamente la otra opción.

Supongamos que también aquí se ha optado por la solución del certificado «normal», es decir, el directo, situación que se ha reflejado en la Figura 6.10. Para que A1 tenga confianza en la clave pública de B2 deberá tener, además del certificado $X21\langle\langle B2 \rangle\rangle$, el $X2\langle\langle X21 \rangle\rangle$ que le ha emitido la CA del centro X2 a su CA subordinada X21, más el certificado $Y\langle\langle X2 \rangle\rangle$ emitido por la CA de mayor rango. Así, A1 aceptará como válida la clave pública de la CA X2 por estar certificada por la CA Y,

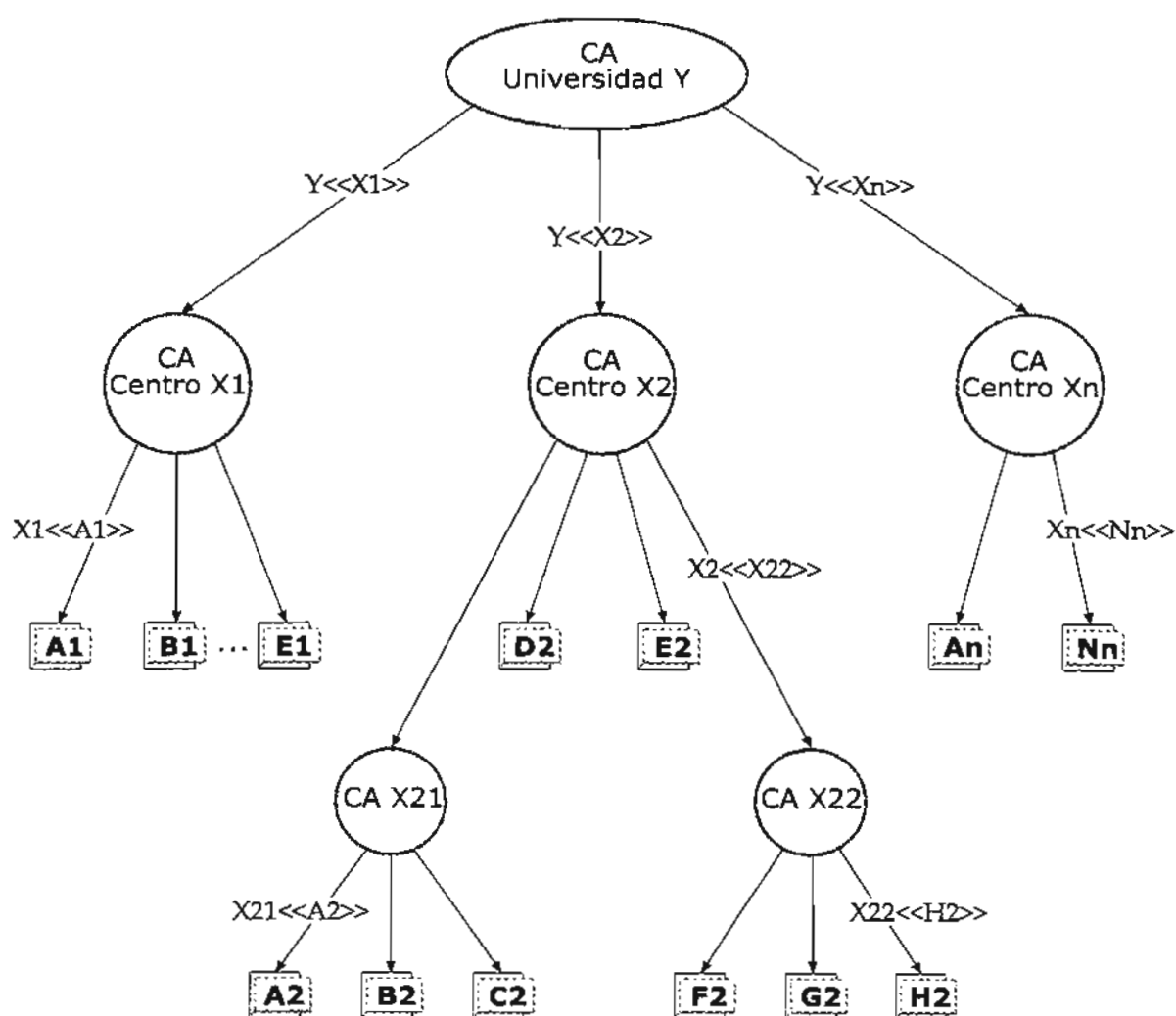


Figura 6.10. Entorno de comunicación con varias CAs organizadas jerárquicamente.

que es para ella una CA de confianza. Mediante un razonamiento análogo, transitivamente, dará por válidas las claves públicas de la CA X21 y de la entidad B2.

Esto es, la confianza se obtiene encontrando un *punto común de confianza* dentro de la estructura en árbol en que están organizadas las CAs. En el ejemplo de la figura, el punto común de confianza entre B2 y F2 sería la CA X2. Los distintos certificados necesarios para establecer la confianza entre dos entidades en una estructura jerárquica forman lo que se denomina *camino de certificación* (*certification path*).

VARIAS CAs DE IGUAL JERARQUÍA COOPERANDO ENTRE SÍ

La mayoría de las infraestructuras de certificación que se han diseñado se adaptan, con algunas variantes, al modelo jerárquico. No obstante, puede ocurrir que sea conveniente definir un dominio de seguridad compuesto por varios subdominios autónomos, cada uno de los cuales tenga su propia Autoridad de Certificación. Tal es el caso de la organización de CAs que se representa en la Figura 6.11, en la que aparecen cuatro CAs distintas, cada una de ellas con un conjunto diferenciado de entidades adscritas.

En este caso, las entidades dependientes de una determinada CA sólo tienen que tener cuidado en conocer la clave pública de la CA a la que están ligadas, para lo cual deberán seguir las normas de publicación acordadas en el subdominio al que pertenezcan. Normas que, dependiendo de la autonomía de que disfruten los subdominios entre sí, podrán ser más o menos dispares.

Para establecer relaciones de confianza entre las entidades del dominio que pertenecen a subdominios distintos, se hace uso de un tercer tipo de certificado: además del certificado directo y del indirecto que ya hemos visto, se define el *certificado cruzado* (*cross certificate*), que consiste en que dos CAs se certifican mutuamente entre sí. De esta manera, si la entidad A1 desea enviar un mensaje confidencial a la entidad B2, para conocer de forma fiable la clave pública de B2 ésta tendrá que presentarle, de una forma u otra, no sólo el certificado que ha sido expedido a su nombre, esto es CA2<<B2>>, sino también el certificado de mutua confianza que hay

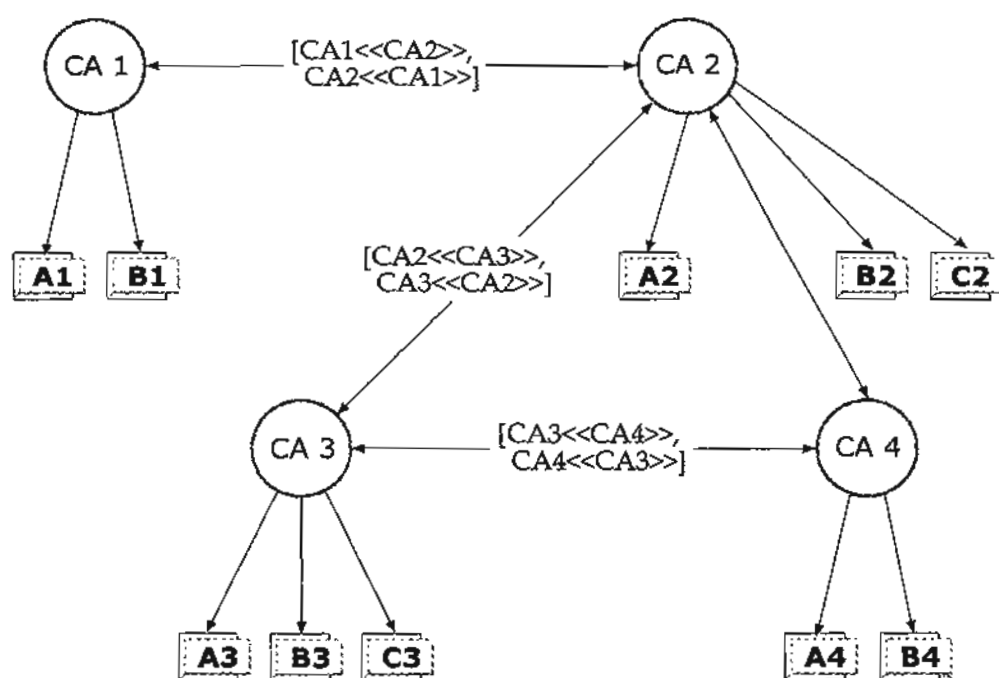


Figura 6.11. Entorno de comunicación con varias CAs de igual jerarquía.

establecido entre las CAs a las que respectivamente pertenecen, es decir, el certificado cruzado [CA1<<CA2>>, CA2<<CA1>>].

Aquí también puede hablarse de camino de certificación. Si, por ejemplo, esa misma comunicación segura fuese a establecerse entre A1 y B3, además del certificado propio, las entidades tendrían que mostrar dos certificados cruzados: el que se establece entre CA1 y CA2 más el establecido entre CA2 y CA3.

SEGURIDAD DE LOS SISTEMAS BASADOS EN CERTIFICADOS

La introducción del concepto de certificado supone un paso de gigante en el perfeccionamiento de los protocolos de distribución de claves en los dominios de seguridad, resolviendo el problema capital de la confianza en el valor de las claves públicas de las entidades comunicantes que en él participan. Por desgracia, siempre hay un «pero», todavía quedan algunos puntos débiles que es necesario detectar.

Queda claro que el certificado es una pieza de información segura en sí misma, independientemente del lugar donde se almacene. Si el Servidor de Certificados es atacado con éxito, lo más que puede hacer el atacante es borrar algunos certificados (negación del servicio) porque la falsificación de la firma es un problema computacionalmente impracticable que no vamos a tomar en consideración. Aunque esta debilidad se podría corregir dotando al Servidor de controles de acceso adecuados, este ataque tampoco tendría consecuencias excesivamente graves, ya que la seguridad básica de las claves queda inalterable.

Mucho más grave sería que alguien consiguiera violar la seguridad de la CA obteniendo su clave secreta y el acceso a la generación de certificados. Podría emitir un certificado a nombre de la entidad B con la clave pública de un usuario malicioso M, que podría, a partir de ese momento, leer los mensajes confidenciales que fuesen enviados a B o firmar falsamente en su nombre. Aunque el ataque, si es masivo, sería prontamente detectado, tendría unas consecuencias bastante catastróficas, ya que sería necesario cambiar la clave privada de la CA y rehacer de nuevo todos los certificados. Cuando se defina la política de seguridad, según veremos en el capítulo correspondiente, habrá que tener muy en cuenta las protecciones tanto físicas como lógicas con que se dote a la máquina que contiene a la CA.

De consecuencias semejantes a las del ataque anteriormente expuesto, aunque más graves, puede ser el hecho de que el gestor de la CA se corrompa y actúe maliciosamente. Esto está directamente relacionado con la elección de la persona o personas en las que los participantes en el dominio depositan su confianza y de las responsabilidades legales o penales que contraiga quien ejerce estas funciones. Este es un riesgo que hay que asumir: hemos ido poniendo capas de protección sucesivas que perfeccionan la seguridad (clave pública, certificado, CA, etc.), pero al final siempre es necesario confiar en alguien o en algo. Pudiera parecer, por tanto, que es ocioso colocar tantas capas, pero una sencilla reflexión nos convencerá de que lo que se está haciendo es ir alejando posibles ataques: desde los más evidentes a los más refinados. (Fuera de los entornos de comunicación a través de redes telemáticas, en el mundo «normal», estas cosas también se producen: si alguna vez se hubiese corrompido el máximo responsable de emitir los billetes del Banco de España, ¿qué valor les quedaría a esos «certificados» de papel? Y no hablamos a humo de pajas.)

Otra conclusión falsa que podemos sacar tras analizar los riesgos de un ataque a la CA (después de haber presentado a los criptosistemas de clave pública como una

panacea para la seguridad en redes), es pensar que poco o nada hemos ganado con el abandono de los sistemas centrales que gobiernan los entornos de seguridad basados en claves secretas. Nada más lejos de la realidad: allí las entidades comunicantes entregaban al CDC o CGC todas las atribuciones sobre su *clave de suscripción*, que es tanto como decir sobre su identidad en la red. Aquí el atacante que lograra vulnerar la CA, fuese éste quien fuese, no tendría capacidad ninguna para cambiar los pares de claves de las entidades adscritas, sino sólo engañar a los restantes participantes durante el tiempo que tarde en descubrirse el fiasco.

Poniéndonos en el peor de los casos, que sea el gestor de la CA el que empiece a actuar delictivamente, lo más grave que puede hacer es expedir un certificado a nombre de B con la clave pública de un compinche malicioso M. De esa forma M podrá empezar a firmar mensajes haciéndose pasar por B y todos los mensajes confidenciales que A (u otro usuario) envíe a B no podrán ser leídos por éste y sí por M. Esta situación no podrá durar mucho antes de ser descubierta. Bien por azar, bien porque A reclame una respuesta a un mensaje que obviamente B no ha recibido, bien mediante reglas rutinarias de comprobación, B se dará cuenta de que su certificado ha sido publicado incorrectamente. Las consecuencias que esto acarrearía para la CA serían contundentes: dejaría de ser una entidad de confianza y desaparecería del mapa. Cuando se dicten normas de política de certificación, a las CAs se les exigirán responsabilidades tanto jurídicas como financieras acerca de la corrección y validez de los certificados que emitan, debido a lo cual los responsables de su gestión se cuidarán muy mucho de introducir falsificaciones o de no poner los medios necesarios para impedir que alguien externo sea capaz de hacerlo.

En cualquier caso, conforme al lema de que, en seguridad, *cuanto menos se confíe en terceros, tanto mejor*, al definir de forma pública la política de seguridad, será necesario establecer medidas adecuadas para evitar que esto pueda ocurrir.

Un último problema, no contemplado hasta ahora, es necesario resaltar: un usuario al que se le haya expedido un certificado durante un cierto periodo de validez puede querer cambiar su par de claves ante el temor de que ya ha sido descubierta su clave secreta. Será necesario emitir un nuevo certificado y revocar el antiguo, pero éste, si sigue circulando por la red, tiene viso de ser auténtico: la firma y el periodo de validez son correctos. La solución, no trivial, a este problema se abordará en el apartado 6.7 al estudiar las listas de revocación de certificados.

OTRAS FORMAS DE ORGANIZAR LA CERTIFICACIÓN

La realidad es mucho más rica que lo que pudiera deducirse si pensásemos que todos los entornos de comunicación posibles podrían encuadrarse en uno de los modelos de organización de CAs que hemos visto en los apartados anteriores. Habrá que buscar modelos que recojan esas diversas formas de relacionarse de forma segura. Al mismo tiempo, se deberá tener en cuenta que cuando dos entidades distintas pertenecientes a dominios de seguridad distintos necesiten establecer una comunicación segura, habrá que tener en cuenta el grado de compatibilidad que tienen entre sí las políticas de seguridad bajo las cuales los respectivos certificados han sido emitidos.

Esta cuestión tratará de abordarse con el máximo de concreción posible en el siguiente capítulo, al estudiar de forma global el tema de las infraestructuras de certificación. No obstante las soluciones no son sencillas y hay pocas propuestas que pretendan dar respuesta a realidades tan complejas. De hecho, en cierta medida, sigue siendo una asignatura pendiente, aunque es de esperar que se vaya resolviendo de

forma creciente con la implantación masiva de los servicios de certificación en las redes telemáticas.

Digamos, para acabar, que existe otra forma de resolver la certificación que no pasa por la existencia de CAs. Nos referimos al esquema de confianza utilizado en el correo electrónico seguro PGP. Aunque en el próximo capítulo se describirá con algo más de detalle este asunto, cabe adelantar ahora que son los propios usuarios los que firman certificados garantizando la validez de las claves públicas de otros usuarios en los que ellos o ellas confían. Los certificados son aquí algo así como mensajes de recomendación que unos hacen acerca de otros. Por ejemplo, si la usuaria A desea enviar un mensaje firmado al usuario B, deberá adjuntar varios certificados que otros usuarios le hayan firmado garantizando la vinculación entre su nombre y la clave pública. Dependiendo del grado de confianza que B tenga en los «recomendadores», así de confianza adquirirá acerca de la validez de la clave pública que A dice tener.

6.4. FORMATO DE LOS CERTIFICADOS

Cuando los comités de ISO y de ITU-T encargados de especificar los servicios y protocolos de seguridad en redes telemáticas contemplaron la necesidad de definir el certificado digital, pensaron que el sitio adecuado para almacenar estas piezas de información sería el Servicio de Directorio, DS (*Directory Service*), más conocido como X.500, su nombre de serie en ITU [ITU X.500]. Fue concebido como un sistema distribuido que soportaría una base de datos de rango planetario, la DIB (*Directory Information Base*), destinada a contener toda la información de las entidades comunicantes que fuese relevante a los efectos de los restantes servicios telemáticos. Más adelante, en el Capítulo 7, se estudiará la estructura de este Sistema como elemento de utilidad para el almacenamiento de certificados. Baste aquí adelantar que al ser los grupos de trabajo responsables del conjunto de normas X.500 los encargados de definir el certificado digital, incluyeron su especificación en una de éstas, la ISO/IEC 9594-8 [ISO 9594] (o su equivalente Recomendación X.509 de ITU-T).

A pesar de la crisis provocada a principios de los noventa para negar la idoneidad de las normas emitidas en el marco del Modelo de Referencia OSI, los planteamientos y definiciones que en este caso se realizaron han seguido vigentes contra viento y marea, de forma que, hoy en día, el llamado *certificado X.509* es el universalmente aceptado. Una excepción particularísima es el caso ya comentado del certificado PGP, aunque existen versiones de esta especificación de correo electrónico que también trabajan con el certificado X.509.

Por todo ello, estudiar los formatos del certificado equivale a estudiar la evolución y distintas versiones y añadidos que han ido apareciendo a lo largo del tiempo que lleva vigente la especificación recogida en la ISO/IEC 9594-8 desde que, en 1988, apareciera su primera descripción completa.

VERSIONES 1 Y 2 DEL CERTIFICADO X.509

En la Figura 6.12 se representa el esquema de la estructura del certificado X.509 conforme a las dos primeras versiones aparecidas en 1988 y 1993 respectivamente. Como puede observarse, su estructura es análoga a la simplificada que se representó en la Figura 6.7, aunque ahora a los DATOS se le han añadido varios campos más. Antes de pasar a describirlos es oportuno hacer notar que, como se recoge en el

apartado 6.6, la especificación formal del certificado está realizada usando una notación normalizada única, por lo que, en dicha especificación, los nombres de los distintos campos están expresados en inglés. En la Figura 6.12 se ha optado por reflejar su nombre en español. Eso mismo se hace en la relación que sigue, indicando entre paréntesis su nombre en inglés cuando la traducción no resulte obvia o cuando se desee recoger la denominación que aparece en la definición formal del certificado. El nombre y contenido de los siete primeros campos es el siguiente:

- Versión*. Indica la versión del certificado conforme a la cual están definidos formalmente sus distintos campos. El valor 0 indica versión 1 y el valor 1, versión 2.
- Número de serie*. Cada CA deberá ir numerando correlativamente todos los certificados que emita, de forma que el número de serie sirve de identificador único para este certificado.
- Algoritmo de firma del certificado (signature)*. Una característica que debe acompañar siempre a las piezas de información cifradas que se incluyan en los protocolos de seguridad es la indicación expresa del algoritmo de cifrado con el que han sido generadas. En este caso ha de indicarse el que ha utilizado la CA en la firma del certificado. Normalmente será RSA o DSA, pero tal y como está concebido el certificado X.509 puede ser cualquier otro algoritmo que sean capaces de manejar las *entidades que se fían* de los certificados emitidos por esta CA.
- Nombre de la CA emisora (issuer)*. En la especificación del certificado está previsto que este campo recoja el *nombre X.500* de la CA.
- Validez*. Indica el comienzo y el final del periodo de tiempo durante el cual el certificado es válido.

CA<<A>>

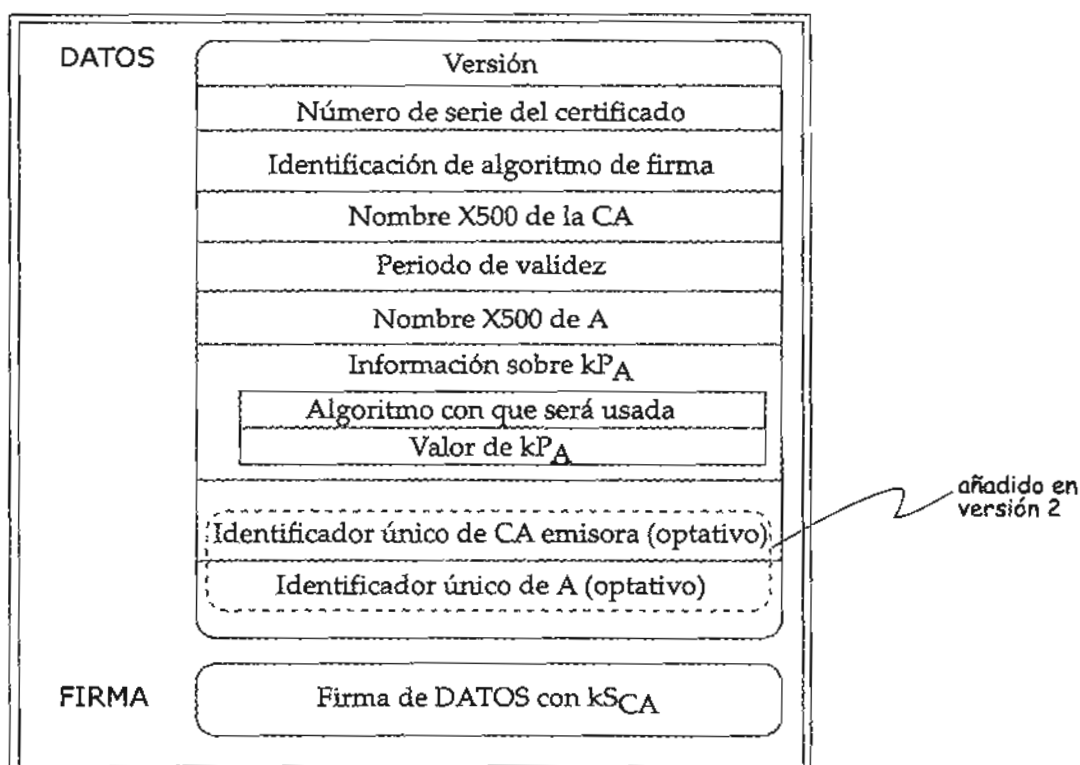


Figura 6.12. Estructura de las versiones 1 y 2 del Certificado X.509 (emitido a nombre de A).

- f) *Nombre del usuario o titular (subject)*. Es el nombre X.500 de la entidad adscrita a esta CA a la que se le ha expedido el certificado. Puede tratarse de una CA a la que otra CA le haya generado un certificado.
- g) *Información sobre la clave del usuario*. Es, evidentemente, el componente principal del certificado. Consta, a su vez, de dos elementos:
 1. *Algoritmo con el que será usada*: identificador del algoritmo con el que se ha previsto que la clave pública sea usada.
 2. *Valor de la clave pública (subject public key)*: es el valor de la clave pública de la que es propietaria la entidad comunicante para la que se ha emitido el certificado.

Los siete campos antes reseñados son todos los contemplados en la versión 1 del certificado X.509 definida en 1988. Posteriormente, se inició un debate, que duró bastantes años, acerca de la conveniencia de que en el certificado se recogiesen o no otras informaciones útiles para la política de seguridad del dominio en el que se utilizase el certificado. De una parte estaban los puristas que, con bastante razón, argüían que lo que realmente debería contener el certificado era lo ya recogido en la versión 1, es decir, la información estrictamente necesaria para relacionar el nombre de una entidad con su clave pública durante un determinado periodo de tiempo, de forma que otro tipo de información, por muy relevante que fuese, debería vehicularse a través de otros mecanismos que a tal efecto se diseñasen.

Por otra parte, se pensó que sería conveniente aprovechar la existencia de una pieza de información tan segura y robusta para incluir, al menos, aunque fuese de forma reducida, algunos aspectos relacionados con la adecuada utilización del certificado. Y fue este segundo criterio el que primó, aunque muy pacatamente, de forma que en un «sí pero poco», se aprobó en 1993 la versión 2 en la que se incluían los siguientes nuevos campos *optativos*:

- h) *Identificador único de CA emisora*. Se trata de una cadena de bits, sin formato específico, que de forma opcional sirve para contener información adicional sobre la CA emisora del certificado.
- i) *Identificador único de la entidad propietaria*. Análogamente al caso anterior, serviría para contener información adicional sobre la entidad a nombre de la cual se ha expedido el certificado.

El nombre de estos campos se deriva de que en la norma se indica que pueden ser usados para resolver la ambigüedad que podría producirse caso de que el mismo nombre X.500 que ahora ostenta la CA o, en su caso, la entidad propietaria del certificado, fuese reasignado a otra entidad por una autoridad de nombramiento en el transcurso del tiempo. Asimismo, indica que puede usarse para alojar el identificador de un objeto, un certificado o cualquier otra información que pueda ser de utilidad para la validación del DN.

Al ser una cadena de bits (*bit string*), sin sintaxis ni semántica puntualmente establecidas, estos campos podían ser usados como cajón de sastre para que en cada dominio, conforme a reglas internamente definidas, pudiesen colocarse ahí cuantos datos de interés considerasen convenientes los gestores del dominio. Esto significaba que cualquier entidad que recibiese el certificado y no estuviese al tanto de ese particular formato de representación, no podría interpretar la información allí contenida. No obstante, el hecho de ser *optativos* contribuía a no hacer perder la validez global del certificado, ya que los restantes campos obligatorios sí podrían ser

interpretados por cualquier entidad que conociese las estructuras de datos definidas en la ISO/IEC 9594-8.

Este mismo argumento de la optatividad de los dos nuevos campos sirve para asegurar la compatibilidad de los certificados emitidos conforme a la especificación de la versión 1, los cuales pueden seguir siendo interpretados correctamente en dominios en los que primen certificados emitidos conforme a la versión 2.

Como puede observarse, en la definición del certificado, para la identificación de las entidades (CA emisora y entidad adscrita a dicha CA) sólo se contempla el **nombre X.500**. Cuando se emitió la norma, se esperaba que el Servicio de Directorio se estableciese, de forma concertada, a lo largo y ancho de todas las redes y todos los países, de forma que este nombre fuese un *Nombre Distintivo*, DN (*Distinguished Name*²), único para cada entidad comunicante, compuesto por campos jerárquicamente organizados y perfectamente localizables, a escala mundial, en todo el ámbito de redes interconectadas. Cada uno de estos campos diferenciadores se denomina RDN (*Relative Distinguished Name*). Aunque en el próximo capítulo volveremos, con más detalles, a este asunto, cabe adelantar ahora algunos datos acerca del formato de este nombre distintivo. Veámoslo a través de un ejemplo:

Supongamos que el nombre de un alumno es Juan García González, que está matriculado en asignaturas del departamento Diatel de la Universidad Politécnica de Madrid. El campo de mayor jerarquía correspondería al del país en el que está ubicada la organización a la que pertenece la entidad comunicante nombrada. El identificador de este campo es «C» (del inglés *country*) y en este caso sería C = es. El segundo corresponde a la *Organización*, que aquí suponemos que es: O = UPM. A continuación puede aparecer un número indeterminado de campos correspondientes a diversas unidades organizacionales jerárquicamente dependientes de la institución catalogada como «organización»: el identificador de estos campos es «OU», del inglés *Organization Unit*. En este ejemplo vamos a suponer que existen solamente dos de estas unidades organizativas: el Departamento y la que agrupa al conjunto de alumnos. Los valores de estos dos RDNs serían: OU = Diatel y OU = Alumnos. Por último aparecería el RDN que contiene el nombre propio (*Common Name*) del usuario, es decir, CN = Juan García González. El conjunto de los distintos RDNs es el nombre X.500 completo:

{ C = es, O = UPM, OU = Diatel, OU = Alumnos, CN = Juan García González }

VERSIÓN 3 DEL CERTIFICADO X.509

La versión 1 del certificado X.509 fue ampliamente aceptada por la comunidad científica, siendo un hito significativo para su difusión la inclusión de este formato de certificado en la especificación del correo electrónico seguro de Internet, el PEM (*Privacy Enhanced Mail*), propiciando así el debate³ acerca de la conveniencia de mejorar la definición de la primera versión con la inclusión de nuevos campos.

Como consecuencia de ese debate, en 1993 se aceptó, como ya se ha dicho, la inclusión de dos nuevos campos, dando así lugar a la especificación de la versión 2. Debido a lo insuficiente de la reforma, esta versión tuvo poquísima aceptación, no habiendo tenido repercusión alguna en el ámbito operativo. De hecho, en la especificación renovada que en 1993 se hizo del PEM siguió usándose solamente la versión 1.

A pesar del paso en falso que supuso la aprobación de la versión 2, dentro de la comunidad científica se continuaron los trabajos conducentes a una nueva configura-

ción que supliera las insuficiencias de la primera versión (abreviadamente, v1). La opción que empezó a considerarse como la más adecuada fue la inclusión de un nuevo campo optativo, el campo de *extensiones*. Como su nombre sugiere, permite definir un número indeterminado de campos adicionales donde recoger los datos que resulten de interés para la política de seguridad definida en el dominio para el que el certificado es generado. Y así, tras un largo debate no exento de divergencias⁴, se llegó a que, a finales de 1996, ISO e ITU-T hiciesen público el documento que recoge la tercera y probablemente definitiva versión del certificado: el Certificado X.509 v3, cuya estructura básica se recoge en las Figuras 6.13 y 6.14. En el ámbito de competencia de Internet, el IETF (*Internet Task Force*) a través del grupo de trabajo PKIX WG (*Public Key Infrastructure Working Group*) se sumaron a esta iniciativa [HPFS02] aunque proponiendo algunas modificaciones no sustantivas y compatibles con la definición general, según se discute a continuación.

Lo primero que puede observarse en la nueva definición es que siguen aceptándose como válidos los dos campos añadidos en la v2, aunque, como se ha comentado, carecen de utilidad práctica. Esto es así porque la norma de que venimos tratando pertenece al grupo «de las antiguas», que en contraposición a la veleidad e ingravidez de muchas de las normas *de facto* con que tenemos que lidiar a diario, mantienen un compromiso de compatibilidad con versiones anteriores. En este caso, esta exigencia

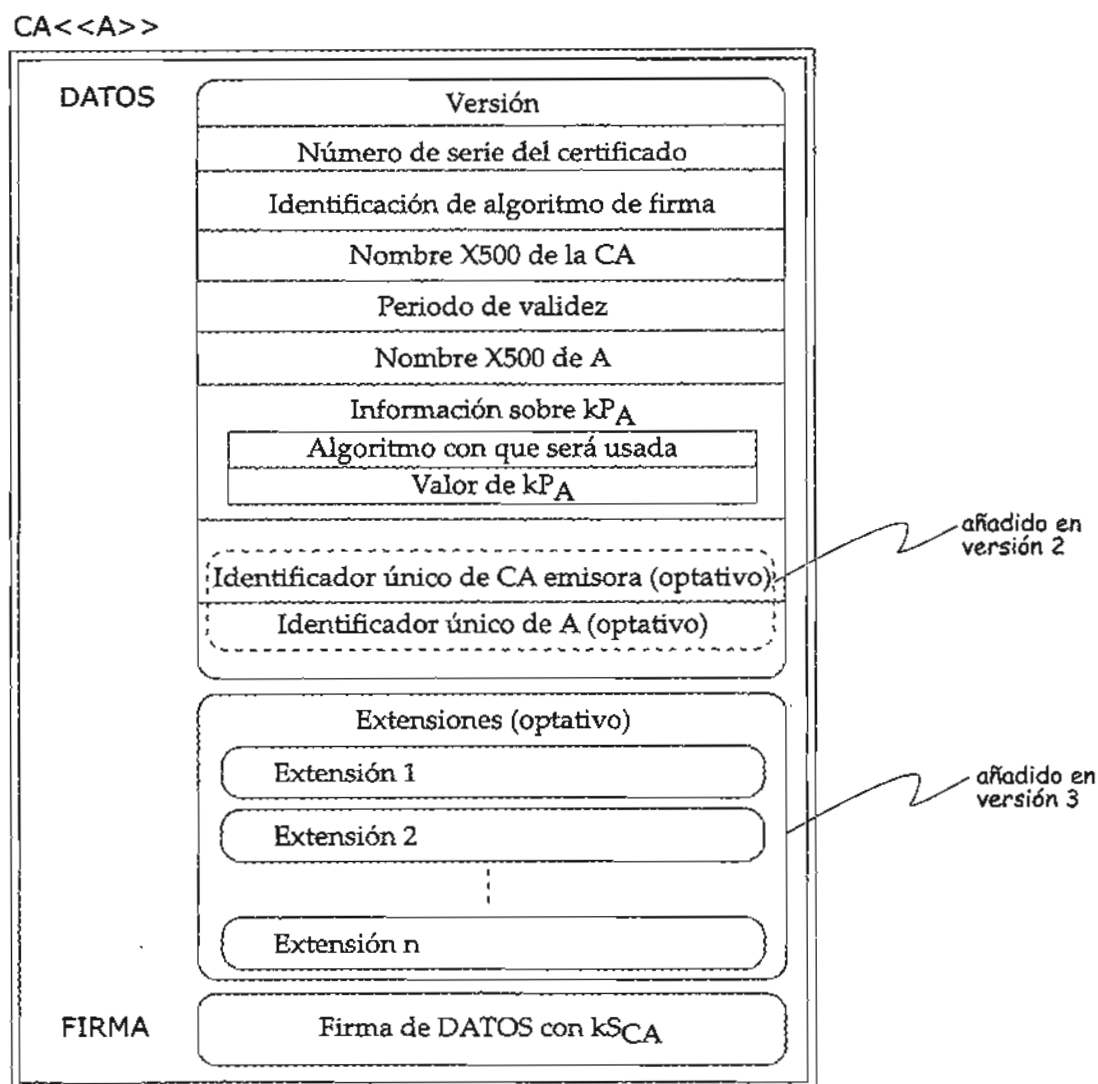


Figura 6.13. Estructura del Certificado X.509 V3 (emitido a nombre de A).

se ve reforzada por el hecho de que las legislaciones de muchos países imponen, en determinadas circunstancias, severas condiciones de vigencia para los certificados durante varios años. Por todo ello, nos vemos forzados a arrastrar en la definición los dos referidos campos como un quiste inoperante pero inofensivo (al ser optativos no tienen por qué aparecer en los certificados reales emitidos).

Como se representa en la Figura 6.13, el campo de extensiones, además de ser optativo en su conjunto, permite incorporar un número indeterminado de campos adicionales. Conforme a la política de seguridad que se defina, pueden elegirse las extensiones que se consideren convenientes a partir del «menú» de extensiones definidas por los organismos de normalización.

Esto quiere decir que, a medida que se detecten nuevas necesidades, los organismos de normalización podrán definir nuevas extensiones que permitan darles respuesta, sin por ello tener que modificar lo más mínimo la definición general recogida en la v3: de aquí la razonable esperanza de que esta versión del certificado pueda considerarse como muy estable (al menos durante muchos años). Además, en un dominio de seguridad concreto se pueden definir nuevas extensiones para satisfacer necesidades específicas, con la peculiaridad de que si no están registradas de forma estándar, sólo serán inteligibles en el ámbito particular en el que han sido definidas.

Las distintas extensiones (Figura 6.14) se reconocen por un identificador del tipo de datos que incorporan. Los organismos de normalización han ido registrando estos identificadores mediante un identificador de objeto (*object identifier*) (en el siguiente epígrafe volveremos sobre este asunto).

Además de ser optativas y elegibles, en la definición de cada una de las extensiones existe un campo (Figura 6.14) de valor binario que ilustra sobre si la extensión es o no *crítica*. En un determinado entorno, cuando una entidad recibe un certificado con una extensión calificada como crítica, para que la operación que realice apoyándose en ese certificado pueda ser considerada como válida, tiene que ser capaz obligatoriamente de interpretarla y de tenerla en cuenta. Por ejemplo, si dice que una determinada clave sólo puede ser usada para el acceso seguro a un sitio web, cualquier otro uso que quiera dársele es ilícito y, en determinadas circunstancias, es posible que esa operación pueda estar sujeta a responsabilidades penales. Por contra, si la extensión está calificada como *no crítica*, el hecho de que una entidad ligada a la CA no pueda, o no quiera, tener en cuenta el valor de esa extensión concreta, no le quita validez al uso del certificado. Es cierto que el marcar como crítica una determinada extensión denota un cierto grado de importancia para el valor que representa. No obstante, este calificativo no está especialmente pensado para marcar como importantes las extensiones

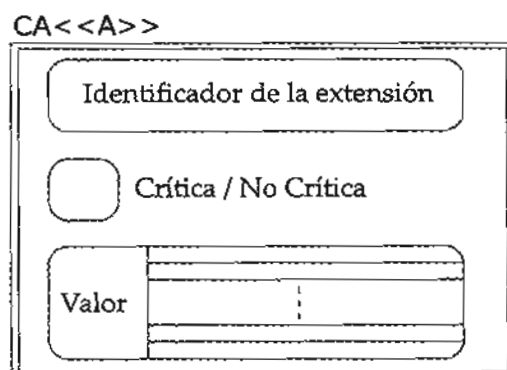


Figura 6.14. Estructura de cada una de las extensiones.

en un determinado dominio de certificación, ya que pueden serlo, y mucho, sin necesidad de ser consideradas como críticas.

El tercer campo que aparece en la Figura 6.14 representa el *valor* de la extensión propiamente dicha y es a su vez una estructura, más o menos compleja, compuesta por diversos campos descriptivos de los datos que aporta.

LOS NOMBRES DE LAS ENTIDADES EN LA X.509 V3

En los primeros borradores de la versión 3 del certificado todavía se arrastraba la concepción de nombramiento presente en las dos primeras versiones, consistente en considerar al DN, o nombre X.500, como única forma de identificar a una entidad. Por cuestión de compatibilidad entre las distintas versiones, en los campos que les son comunes estas restricciones se siguen manteniendo, aunque no así en las extensiones, donde las distintas entidades se pueden identificar por uno o más nombres contruidos conforme a distintos formatos.

Ya se comentó que en el punto de mira de los diseñadores del Servicio de Directorio X.500 estaba el hecho de que el sistema que se implementase abarcaría todo el planeta y que, de forma similar a las guías telefónicas existentes en ese momento, todos los usuarios de todas las aplicaciones telemáticas podrían reconocerse por ese único nombre distintivo. Como la realidad, para bien o para mal, no ha ido por esos derroteros, la unánime voluntad de usar el certificado X.509 como el formato más adecuado para la validación de las claves públicas se topaba con el escollo de tener que atribuir un DN a quienes en absoluto lo tenían y no poder utilizar otras formas de nombramiento utilizadas en las distintas aplicaciones telemáticas que están operativas.

El caso más llamativo es el del correo electrónico. Bien porque se use la dirección X.400 o, más comúnmente, una dirección RFC 822 (tal que, por ejemplo, `juan.garcia@alumnos.euitt.upm.es`), es el caso que la aplicación de correo electrónico seguro identifica al usuario de una forma y el certificado (v1 o v2) que maneja lo identifica de otra (a veces forzada debido a que tal usuario no tiene asignado previamente un nombre X.500). Otro caso llamativo puede ser que la entidad tenga asignado un determinado URL y se trate de una aplicación de securización del acceso a un sitio web.

Por todo ello, en la versión 3 del certificado X.509 se acepta, mediante las extensiones, que el usuario pueda identificarse de varias formas, de tal manera que se marcaría con cuál o cuáles de ellas la entidad propietaria o la CA emisora pueden ser identificadas. Este nuevo tipo de *nombre general* (*General Name*) puede adoptar valores pertenecientes a uno de los siguientes tipos:

- Dirección Internet de correo (RFC 822).
- Nombre Internet de dominio (DNS).
- Dirección de correo X.400.
- Nombre X.500.
- Nombre compuesto EDI.
- Identificador URI (*Uniform Resource Identifier*).
- Dirección IP.
- Otros nombres que sean definidos y registrados.

Como puede observarse, la especificación está abierta a aceptar cualquier otro tipo de nombramiento que se desee implantar en un dominio determinado. Lo más signi-

ficativo es que la entidad no tiene que verse obligada a elegir un solo tipo de nombramiento, sino que puede decidir incluir varios de ellos (todos los cuales la identifican de forma no ambigua) de cara a que las distintas entidades que usen ese certificado puedan localizarla por distintas vías, todas ellas compatibles entre sí.

EXTENSIONES DEL CERTIFICADO X.509 V3

Para simplificar, podemos dividir en varios grupos las extensiones definidas por los organismos de normalización. Exceptuando las extensiones relacionadas con la revocación (que se comentarán en el epígrafe 6.7), estos grupos serían:

- Relacionadas con claves y políticas.
- Relativas a los atributos de la CA emisora y de la entidad propietaria del certificado.
- Relativas a las restricciones sobre el camino de certificación.

A continuación se hace una somera descripción de cada una de ellas, limitándonos a reflejar la información que aportan. En casi todos los casos, las circunstancias que determinan el interés de su inclusión están en función de la política de seguridad que se defina en un determinado dominio, situaciones que se discutirán en el capítulo dedicado a Políticas de Seguridad.

Extensiones relacionadas con las claves y las políticas

1. *Identificador de clave de la autoridad emisora (Authority Key Identifier)*

Una CA debe actualizar de manera periódica su clave pública o puede verse obligada a hacerlo por circunstancias especiales relacionadas con ataques y disminución de la seguridad de su clave privada. En organizaciones complejas podría incluso usar claves distintas para firmar certificados expedidos bajo distintas políticas (situación poco aconsejable). En cualquier caso, esta extensión ayuda a la entidad que haga uso del certificado de forma que pueda identificar fiablemente la clave pública con cuya correspondiente clave privada se ha firmado el certificado. Para ello, entre las posibilidades que ofrece esta extensión está la de incluir un identificador y/o exhibir el certificado que otra CA de mayor jerarquía le ha expedido para garantizar la validez de su clave pública, lo que puede ser de suma utilidad para determinar el *camino de certificación* a que nos referíamos en el epígrafe anterior cuando estudiamos la problemática de varias CAs cooperando entre sí. Para nombrar a estas CAs se utiliza el tipo *General Name* antes descrito. Esta extensión no es crítica.

2. *Identificador de la clave de la entidad propietaria (Subject Key Identifier)*

Como en el caso anterior, la entidad para la que ha sido expedido el certificado puede cambiar de forma periódica su par de claves, con lo que esta extensión sirve para ayudar a conocer con puntualidad cuál es la clave certificada. Más interesante es la circunstancia de que un mismo usuario mantenga varios pares de claves distintas, para usos distintos, bajo políticas de seguridad distintas. En este caso la extensión nos ayudará a identificar sin ambigüedad cuál de ellas es la clave pública para la que se ha emitido el certificado. Esta extensión no es crítica.

3. *Uso de la clave (Key Usage)*

Para el caso que acabamos de comentar, cuando un usuario tiene distintas claves para usos distintos y, consecuentemente, certificados diferentes para usos diferentes, es importantísimo que en el certificado expedido se recoja cuál es el uso específico para el que ha sido emitido, permitiendo, por tanto, que una CA estipule que una clave certificada sólo puede ser utilizada para un determinado fin. Esta información puede ser de interés incluso en el caso de una entidad que sólo tiene un par de claves pero solicita distintos certificados para distintos usos. La extensión permite marcar uno o varios de los siguientes usos:

- Firma digital.
- No repudio.
- Cifrado de clave (como en la envoltura digital).
- Cifrado de datos.
- Concertación de clave (como en el caso de acordar una clave de sesión).
- Firma de certificados X.509.
- Firma de listas de certificados revocados (se describen en 6.7).

Esta extensión puede ser calificada como crítica o como no crítica, y la información que aporta es de suma trascendencia para la definición de políticas de seguridad, como más adelante se verá. Dependiendo de su criticidad y de cómo esté definida esa política, el uso de la clave certificada para usos distintos de los autorizados puede dar lugar, incluso, a responsabilidades penales (como no podía ser de otra forma, si lo que pretendemos es usar los servicios de seguridad para sustituir las transacciones convencionales por comunicaciones a través de protocolos telemáticos).

4. *Periodo de uso de la clave privada (Private-key Usage Period)*

Esta extensión está pensada para los casos en que la clave privada se usa para firmar documentos y la correspondiente clave pública certificada sirve a las entidades usuarias del certificado para la verificación de esas firmas. Normalmente, el periodo de utilización de una clave privada de firma será más corto que el de la correspondiente clave pública que sirve para verificarla. De esta forma, pueden limitarse los periodos de uso del firmado con vistas a protegerse de posibles pérdidas de seguridad en la custodia de dicha clave privada. Esto sirve para ayudar en la definición de la política de seguridad. Esta extensión no es crítica.

5. *Políticas de Certificación (Certificate Policies)*

Esta extensión sirve para indicar las políticas bajo las cuales es válido usar este certificado. Puede que en el dominio de seguridad en que el certificado se está usando estén definidas varias políticas diferenciadas, siendo necesario explicitar bajo cuáles de ellas es lícito usar el certificado. Esta extensión puede ser calificada como crítica o no crítica. Dependiendo de esta calificación, la puntualización que aquí se realiza tendrá mayor o menor grado de obligatoriedad en su cumplimiento.

6. *Correspondencias de políticas (Policy Mappings)*

Esta extensión sólo se usa en certificados cuya propietaria sea una CA emisora de certificados y, como su nombre indica, sirve para la nada baladí especificación de cuáles de las políticas de seguridad bajo las que están expedidos sus certificados son

compatibles (y en qué medida) con las políticas definidas en otros entornos. Esta extensión no es crítica.

Extensiones sobre atributos relativos a la entidad propietaria del certificado y a la CA emisora

Se han introducido determinadas extensiones para mejorar la información que se refleja en los campos heredados de la versión 1 acerca de las distintas entidades que participan en el entorno de certificación. Esta información sirve para ayudar a la entidad ligada a la CA en la adecuada identificación de la entidad propietaria de los certificados emitidos. Para ello se vale del mayor abanico de posibilidades que se ofrecen en la especificación de estos campos adicionales.

1. Nombre alternativo de la entidad propietaria (Subject Alternative Name)

Indica otros nombres alternativos, además del DN X.500, por los que puede ser identificada la entidad a la que se le ha expedido el certificado. El tipo de nombre permitido es del tipo *General Name* antes comentado, que fue introducido en la especificación del certificado X.509 al generar la versión 3. Esta extensión puede ser crítica o no crítica. Si en el campo del certificado reservado al nombre de la entidad propietaria del certificado no aparece su DN X.500, esta extensión debe ser crítica.

2. Nombre alternativo de la CA emisora (Issuer Alternative Name)

De forma análoga al caso anterior, permite nombrar de múltiples formas no ambiguas y compatibles entre sí a las CAs emisoras de certificados. Esta extensión puede ser crítica o no crítica.

3. Atributos de Directorio de la entidad propietaria (Subject Directory Attributes)

Sirve para añadir al certificado información adicional sobre la entidad titular del certificado y así obtener confianza en que el titular del certificado es la persona o cosa requerida. Pueden ser datos que no se reflejan en el nombre, como el cargo que ocupa el usuario dentro de la organización, dirección postal, número de teléfono, una imagen, etc. Esta información se incluye con formato de atributos de Directorio. En el apartado *Árbol de información del Directorio* del epígrafe 7.3 se describe someramente en qué consisten estos atributos. Por ahora, baste decir que el propio certificado es un atributo de Directorio. Esta extensión no es crítica.

4. Acceso a la información de la entidad propietaria (Subject Info Access)

En cierto modo es alternativa a la extensión anterior. Si en esa la información adicional se obtenía mediante atributos de Directorio, esta otra está pensada para localizar información adicional sobre la entidad titular del certificado en entornos en los que no está implantado el Servicio de Directorio. Para ello contiene una URI que describe la localización, el formato básico y el esquema de acceso a esa información. Esta extensión y la que se describe a continuación han sido propuestas por el IETF-PKIX.

5. Acceso a la información de la CA (*Authority Info Access*)

Es similar a la anterior pero destinada a facilitar el acceso a la información y a los servicios de la CA emisora del certificado. Asimismo, indica cómo y dónde conocer el estado del certificado, es decir, si está activo, o está en suspenso su uso por algún motivo de sospecha u oportunidad, o bien ha sido revocado definitivamente por riesgos graves en la custodia de la clave privada pareja de la pública que el certificado valida.

Extensiones relativas a restricciones sobre el camino de certificación

Cuando un grupo de CAs están organizadas de forma que cooperen entre sí, se podría pensar que cada una de ellas, en el subdominio de su competencia, podría delegar las tareas de certificación en otras tantas CAs dependientes de ella. Sucesivamente, esta misma operación de delegación-desdoblamiento se podría volver a producir en las nuevas «CAs terminales». En principio, desde un punto de vista puramente teórico, nada habría que objetar a esta forma de proceder porque cada nueva CA instalada en el sistema no crea ninguna inconsistencia con la CA de rango superior de la que procede, pero desde un punto de vista organizativo crea una distribución de competencias que a la larga puede desembocar en desorganización. Además, el camino de certificación aumenta cada vez que una CA se descompone en otras CAs, de forma que el número de certificados necesarios para establecer una relación de confianza se va incrementando. Por todo ello es conveniente definir unas extensiones que regulen y pongan coto a esta forma de proceder.

1. Restricciones básicas (*Basic Constraints*)

Indica si la entidad propietaria del certificado puede actuar como CA, o se trata, simplemente, de un usuario final. Además, de forma optativa, caso de tratarse de un certificado de CA, esta extensión permite delimitar la longitud del camino de certificación, especificando, mediante un número entero, el máximo número de saltos permitidos. Puede ser crítica o no crítica, aunque se recomienda lo primero para dejar bien acotado el derecho de los usuarios finales.

2. Restricciones sobre el nombre (*Name Constraints*)

Cuando se trata del certificado expedido a una CA, esta extensión permite restringir el espacio de nombres que pueden «colgar» de esta CA. Es decir, indica cuáles son los subárboles permitidos y cuáles los prohibidos, especificando la raíz y los números máximo y mínimo de CAs de que dicho subárbol puede constar. Se recomienda marcarla como crítica, aunque también puede ser no crítica.

3. Restricciones sobre la política (*Policy Constraints*)

Mediante esta extensión, una CA puede exigir de manera explícita la presencia de ciertas políticas (determinadas por un identificador específico) en los siguientes certificados de la cadena de certificación. Asimismo, puede explicitar que se prohíben las correspondencias (*mappings*) entre políticas en los siguientes certificados de la cadena.

6.5. NOTACIÓN ASN.1

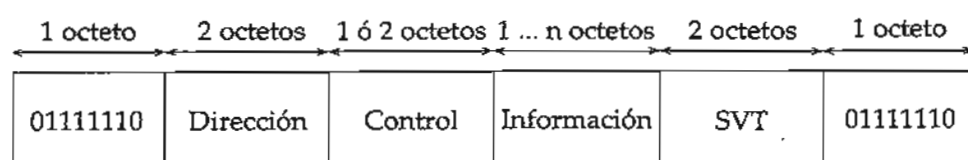
La *Notación de Sintaxis Abstracta uno*, conocida como ASN.1 por derivación de su nombre en inglés, apareció por primera vez en 1984 como Recomendación X.409, incluida dentro del conjunto de normas de la primera versión de correo electrónico OSI propuesta por el entonces CCITT (hoy ITU-T) bajo la referencia genérica X.400. Se propuso como un mecanismo útil para describir las PDUs que se intercambian en los protocolos que soportan dicha aplicación telemática. Posteriormente, se publicó como norma diferenciada⁵ ya bajo el nombre ASN.1 con el que hoy se conoce, y se generalizó su uso para definir cualquiera de las APDUs (PDUs del nivel de Aplicación) que fuesen apareciendo en la especificación de los distintos servicios ubicados en este nivel. El Servicio de Directorio (DS) fue uno de ellos y esa es la razón por la que el Certificado X.509 está especificado siguiendo las reglas marcadas en ASN.1, lo que conlleva que, para entender adecuadamente su descripción formal (objeto del siguiente apartado), sea necesario tener unas mínimas nociones de esta técnica de descripción.

Proporcionar una descripción resumida y simplificada de esta notación es, precisamente, el objetivo que se pretende cubrir en este apartado. En realidad, ASN.1 especifica dos cosas distintas y complementarias: de una parte es una notación formal que sirve para definir la sintaxis y la semántica de una PDU, es decir, los campos y subcampos de que consta y el significado de la información que cada uno de ellos alberga. De otra, nos proporciona las **reglas de representación** de esas estructuras de datos, es decir, nos sirve para determinar de forma no ambigua la ristra de unos y ceros (o lo que es lo mismo, el conjunto de valores hexadecimales) en que se convierte cuando la PDU es almacenada o transferida entre dos entidades comunicantes.

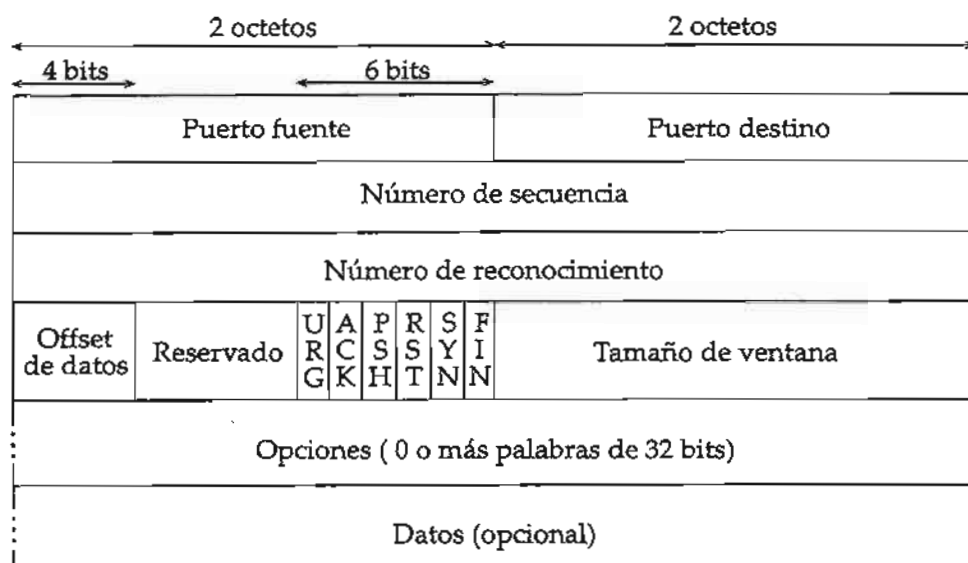
Esta forma de proceder puede ser considerada como una evolución en la representación del formato de las PDUs presentes en los distintos protocolos de las arquitecturas telemáticas organizadas jerárquicamente en niveles. En efecto, la técnica usada para representar una PDU en los niveles bajos de la arquitectura consiste en una gráfica constituida por un conjunto de cajitas ligadas entre sí en las que se señala su longitud en bits y el tipo de información que están destinadas a contener. A modo de recordatorio, en la Figura 6.15 se representa una trama de nivel 2 y la PDU definida en TCP.

Puede observarse que la sencillez y concreción con que está representada la primera no se corresponde con las de la segunda, en la que existen campos de longitud variable que, asimismo, pueden estar o no presentes. Además, para la correcta interpretación de la especificación es necesaria la correspondiente explicación textual que informa sobre los detalles, con las consiguientes imprecisiones interpretativas. En cualquier caso, esta forma de descripción es muy eficaz porque se suministra a un tiempo información tanto sobre la sintaxis y la semántica como sobre la representación en bits a que dan lugar. A medida que aumenta el número de campos optativos y que existe la posibilidad de que un mismo campo pueda ser monoevaluado o estar constituido por un número no fijo de valores (como es ya el caso de las PDUs del nivel OSI de Sesión), esta forma de representación mediante cajitas se hace cada vez más engorrosa e ineficaz. Por ejemplo, en el caso del correo electrónico, el campo destinado a la dirección del destinatario puede alojar una sola dirección o un conjunto elevado de ellas, siendo, además, unas más largas y detalladas que otras.

Ese fue el motivo por el que con el inicio de la especificación de aplicaciones telemáticas avanzadas (desde la perspectiva de 1984) se pensase en una técnica de descripción formal para la especificación de las complejas y multiformes PDUs del



a) Formato único de trama en LAP-D



b) Estructura de la PDU de TCP

Figura 6.15. Formato de dos PDUs de niveles bajos.

nivel de Aplicación. Algo después, con mayor intensidad a principios de la década de los noventa, coincidiendo con la campaña de voladura controlada de todo lo relativo a OSI, se desencadenó una fuerte polémica acerca de lo inoportuno de ASN.1, debido a la complejidad que conlleva, tanto la definición como las herramientas necesarias para su adecuado manejo. (Como suele ocurrir, fueron los neoconvertos, los que anteriormente apostaban todo por *sólo OSI*, los que de forma más virulenta y radicalizada proponían su eliminación.)

Se ponía como ejemplo la forma tan sencilla de describir los sencillos mensajes del SMTP (*Simple Mail Transfer Protocol*, el primer estándar de Internet para correo electrónico). Lo que ocurre es que cuando las cosas son complejas es necesario disponer de herramientas que permitan describir esa complejidad, y así, andando el tiempo, ASN.1 comenzó a ser usado en una gran variedad de protocolos, no sólo del ámbito OSI, sino también en el de las RFCs de Internet. De forma más significativa, se usa en todo lo relacionado con los certificados y con Seguridad en Redes. Por tanto, hoy en día es imprescindible tener un mínimo conocimiento de esta notación para poderse desenvolver con cierta soltura en el mundo de las aplicaciones telemáticas seguras⁶.

En la descripción que sigue⁷ se abordará primero la forma de especificar los tipos de datos y valores que pueden serles asignados, y después se comentarán las diferentes reglas de codificación (BER, DER, CER y PER). El método seguido no será una rigurosa y jerárquica descripción de todos los elementos que definen la notación, sino un conjunto de ejemplos comentados que puedan servir a los fines didácticos aquí propuestos. Quien necesite una descripción precisa y rigurosa, debe acudir a la fuente más segura y detallada: la norma.

TIPOS Y VALORES

Los tipos en ASN.1 son similares a los de un lenguaje de programación de alto nivel, aunque es importante tener en cuenta que ASN.1 no es, en absoluto, un lenguaje de programación. En efecto, con un lenguaje de programación, C por ejemplo, se puede describir, como parte de un programa, un tipo constituido por una estructura de datos, para luego definir una variable de ese tipo y asignarle valores. Posteriormente, el programa será compilado y la representación en bits de esa variable será, dentro de la memoria del computador, una ristra de octetos cuyos valores binarios se han establecido conforme a la sintaxis local de representación de datos definida en el Sistema Operativo bajo el que opera esa máquina. Ese mismo programa compilado en otra máquina distinta daría lugar a una representación binaria distinta. Hay dos pasos bien definidos y separados en el tiempo: la compilación (de la que se obtiene el programa ejecutable en lenguaje de máquina) y, posteriormente, la ejecución del programa. La situación aquí es bien diferente, para lo que nosotros queremos⁸, ASN.1 nos sirve «sólo» para definir el formato de las PDUs y para determinar de forma no ambigua el valor binario que tendrán las PDUs que se vayan generando, en vivo y en directo, en el transcurso de la ejecución dinámica de la aplicación telemática a la que pertenecen.

En la Figura 6.16 se muestra un ejemplo sencillo. En primer lugar, se aprecia que, a falta de detalles, la descripción no resulta totalmente ininteligible para una persona con cierta formación en lenguajes. La razón de ello es que, obviamente, los diseñadores de ASN.1 procuraron que así fuese. Como se indica en el ejemplo, el nombre de los tipos deben empezar por letra mayúscula y el de los campos que lo componen por letra minúscula. También se indica cuáles son los signos utilizados para la asignación y para los distintos comentarios. Estos últimos no tienen ningún efecto en la especificación y sólo sirven como ayuda al lector, que no es poco.

Imaginemos que se está definiendo una aplicación telemática para intercambio de datos de ciertos usuarios y los servicios que usan. Porque se desea que las máquinas remotas que se conecten puedan procesar automáticamente las informaciones recibi-

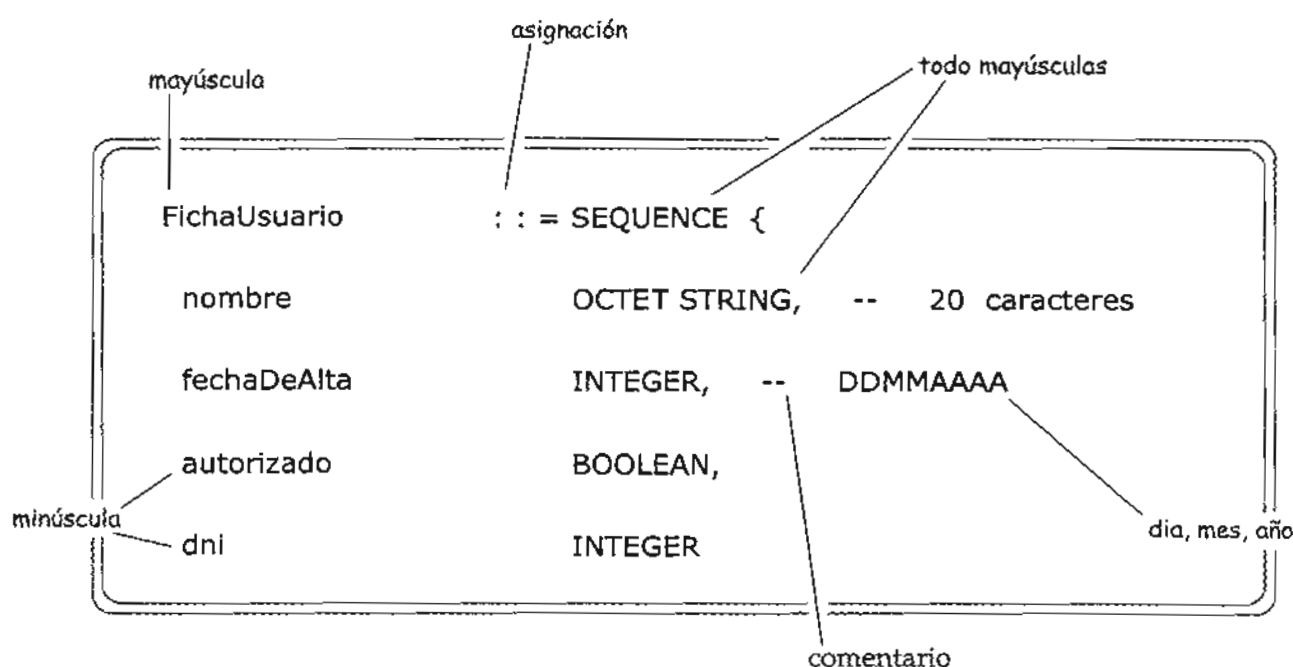


Figura 6.16. Ejemplo de especificación de una PDU sencilla.

das, es necesario definir la sintaxis y la semántica de las PDUs que se intercambien. Una primera necesidad es que la especificación de la aplicación sea legible para los telemáticos interesados en su implantación. En esa descripción existirán unas palabras-clave, definidas en la norma (que se escriben en mayúsculas) y otras, las etiquetas auxiliares, que sólo sirven de ayuda al lector, que han sido inventadas por el redactor de la PDU (en el ejemplo se han elegido en español, aunque sin eñes ni acentos) y que no generan dato alguno en la representación binaria de la PDU.

En cuanto a los **valores** que pueden ser asignados a la estructura antes definida, cabe decir que cada tipo definido puede tomar un conjunto de valores y que cada valor constituye un ejemplar de ese tipo, o sea, una PDU concreta. Es decir, lo que se ha definido en el ejemplo es *una plantilla* (al igual que los formatos de PDUs representados en la Figura 6.15) que tendrá que ser «rellenada» con valores concretos en sus distintos campos, y que dará lugar a ejemplares concretos de PDUs que puedan ser transmitidas a través de la red. Un valor concreto del tipo *FichaUsuario* sería el que hemos denominado «*este-usuario*». Podría ser:

```
este-usuario FichaUsuario ::= {
    nombre "Juan Abad Rojo",
    fechaDeAlta 25021999,
    autorizado TRUE,
    dni      9647541
}
```

Para entender su significado con mayor precisión, veremos a continuación los elementos más importantes. Los tipos más significativos que pueden aparecer en una especificación ASN.1 los podríamos clasificar de la forma siguiente:

- Simples: BOOLEAN, INTEGER, ENUMERATED, REAL, BIT STRING, OCTET STRING, NULL, Ristras de caracteres, Tiempos, OBJECT IDENTIFIER y OBJECT DESCRIPTOR.
- Estructurados: SEQUENCE, SEQUENCE OF, SET y SET OF.

TIPOS SIMPLES

BOOLEAN

Tiene únicamente dos valores, TRUE o FALSE. Por ejemplo, se puede definir:

```
Autorizado ::= BOOLEAN
```

Que se leería: «*Autorizado* tipo *BOOLEAN*».

Un valor concreto del tipo antes definido podría ser:

```
usuario Autorizado ::= TRUE
```

Que se leería: «*usuario*, tipo *Autorizado*, valor *TRUE*».

INTEGER

Representa un número entero arbitrariamente grande porque, a diferencia de lo que ocurre en un lenguaje de programación de alto nivel, aquí se trata sólo de un meca-

nismo para posibilitar una definición abstracta. Cuando se diseñe una aplicación telemática, que obviamente ha de residir en una máquina concreta, se pondrán los límites que se consideren necesarios al tamaño de los datos que contenga. Ejemplo:

```
Altura ::= INTEGER -- en metros
torre Altura ::= 51
```

ENUMERATED

Permite modelar tipos con varios valores identificados por sus nombres. Ejemplo:

```
Laborables ::= ENUMERATED {lunes (0), martes (1),
                             miércoles (2), jueves(3), viernes (4)}
```

Como «lunes» equivale a «0», la asignación de valor a este supuesto segmento de PDU se puede expresar de forma exactamente equivalente siguiendo una de las dos formas siguientes:

```
dia Laborables ::= lunes
dia Laborables ::= 0
```

REAL

Sirve para representar un número real tan preciso como sea necesario (puede tener un número ilimitado de dígitos) y se puede usar tanto en base 2 como en base 10. Ejemplo:

```
Radianes ::= REAL
pi Radianes ::= {31416, 10, -4}
```

que se interpreta como que el valor de «pi» es: $31416 \cdot 10^{-4} = 3,1416$

BIT STRING

Representa una ristra de bits de longitud indeterminada. La ausencia de limitaciones en el tamaño de la cadena obedece a las mismas causas antes comentadas. Ejemplo:

```
UnidadesFuncionales ::= BIT STRING {
    kernel (0), token (1), activity (2)}
```

Los valores se pueden expresar en binario, hexadecimal o mediante identificadores. Es decir, serían equivalentes las dos siguientes asignaciones de valores:

```
uf UnidadesFuncionales ::= '101'
uf UnidadesFuncionales ::= {kernel, activity}
```

OCTET STRING

Representa una ristra de octetos de longitud indeterminada. Ejemplo:

```
Nombre ::= OCTET STRING, -- 20 caracteres
cliente Nombre ::= "Juan Abad Rojo"
```

NULL

Se usa para indicar la ausencia de un elemento cuando sea necesario. Es decir, varias alternativas son posibles, pero ninguna se aplica. El valor único de este tipo es NULL. Ejemplo:

```
Nada ::= NULL
nadaDeNada Nada ::= NULL
```

Ristras de caracteres

Constituyen redefiniciones del OCTET STRING restringiendo los valores de cada octeto. Los usados con más frecuencia son: *NumericString* (dígitos decimales y espacio), *VisibleString* (ASCII internacional imprimible), *IA5String* (alfabeto internacional n° 5), *GrafiString* (todos los conjuntos de caracteres registrados más espacio), y *PrintableString* (letras latinas mayúsculas y minúsculas, dígitos decimales y () ' + - . , / : = ?). Ejemplo:

```
Nombre ::= VisibleString, -- 20 caracteres
cliente Nombre ::= "Juan Abad Rojo"
```

Tiempos

Son redefiniciones del *VisibleString*. Existen dos formatos: *UTCTime*, que sólo permite dos dígitos para consignar el año, y *GeneralizedTime*, que usa cuatro dígitos para asignación. Ejemplo:

```
Fecha ::= GeneralizedTime
fin-sXX Fecha ::= "19991231235959.9"
```

cuyo significado es: día 31 de diciembre de 1999 a las 23 horas 59 minutos y 59,9 segundos. Naturalmente, ASN.1 ni entra ni sale en la polémica de cuándo terminó el siglo xx.

IDENTIFICADORES DE OBJETOS

Existe en ASN.1 otro tipo simple, el **OBJECT IDENTIFIER**, que merece una atención especial por su significado y trascendencia. En una especificación ASN.1, por ejemplo del Certificado X.509, se puede hacer referencia a otros elementos u objetos que ya han sido descritos con anterioridad y se encuentran debidamente registrados y catalogados. Ello representa un ahorro de esfuerzo a la hora de redactar las especificaciones, al tiempo que aumenta la fiabilidad y facilita la actualización de lo definido.

Existe un conjunto de Autoridades de Registro [ISO 9834] que tienen capacidad para asignar identificadores a determinados objetos para que puedan ser referidos desde cualquier especificación. En realidad, un *OBJECT IDENTIFIER* consiste en un conjunto de números enteros asignados a un objeto concreto, de tal forma que ese valor es único dentro del espacio de identificadores de objeto existentes. Para conseguir esa unicidad, el procedimiento de asignación está regulado en las normas ASN.1,

donde se define una organización jerárquica en forma de árbol, de tal manera que cada autoridad es propietaria de un nodo y es responsable de los números que se asignan a las entidades u objetos de ella dependientes. En la Figura 6.17, a modo de ejemplo, se representa una parte del árbol de identificadores de objetos.

En el máximo nivel jerárquico, con la asignación de los números 0 al 2 respectivamente, se encuentran *itu-t*, *iso* y la alianza de ambas organizaciones: *joint-iso-itu-t*. A todos los efectos, para guardar la compatibilidad con las antiguas definiciones, *ccitt* es sinónimo de *itu-t*. Del nodo de *itu-t* salen cinco arcos, con cinco nuevos identificadores nominales y numéricos para otras tantas instancias de asignación de identificadores. En el nodo *iso* se especifican tres, mientras que los arcos situados debajo del nodo *joint-iso-itu-t* tienen valores acordados cada cierto tiempo por ambas organizaciones. Hasta este segundo nivel (y en otros niveles cuando se especifica), los identificadores pueden reflejarse tanto en forma de números como por los nombres asignados a cada uno de los arcos.

En el ejemplo de la figura se muestra, dentro de la rama de *joint-iso-itu-t*, el conjunto de normas X.500 (Servicio de Directorio, *ds*), que se identifica por el número 5. Situado inmediatamente debajo de él está el arco 1, que se corresponde con los

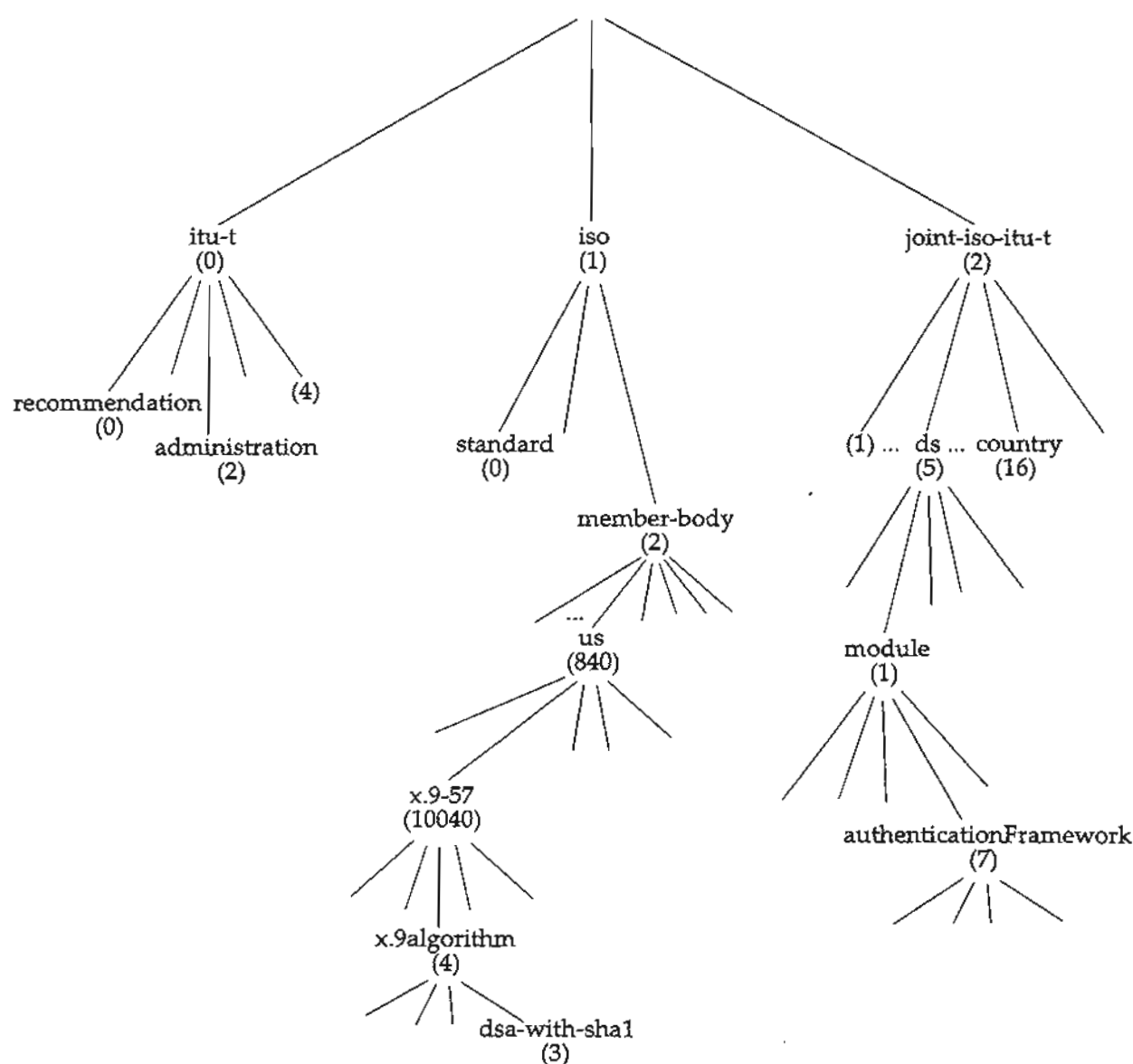


Figura 6.17. Árbol de identificación de objetos.

módulos X.500 definidos, y debajo de él está el arco que identifica al *Marco de Autenticación*, que es el nombre con que se conoce la norma X.509, que tiene asignado el número 7. Debido a que existe una correspondencia entre estos números y determinados nombres, una referencia a esta norma desde una especificación ASN.1 podría ser una de las siguientes:

```
{joint-iso-itu-t (2) ds(5) module(1) authenticationFramework(7) }
{joint-iso-itu-t ds(5) module(1) authenticationFramework(7) }
{ 2 5 1 7 }
```

Otro ejemplo representado en la figura es el del algoritmo de firma DSA en conjunción con otro algoritmo, el SHA-1 (ambos se describen en detalle en el Capítulo 8). En este caso, está registrado en *iso (1)* y, dentro de este organismo, en el arco *member-body (2)*. Este es un nodo debajo del cual se sitúan las autoridades nacionales de registro (naciones reconocidas como tal en los organismos de normalización). Es equivalente al {0 2} y al {2 16}. Siguiendo con el ejemplo, *us (840)* es el identificador para Estados Unidos y para el organismo delegado de ISO en ese país. El siguiente arco, *x.9-57 (10040)*, corresponde a una parte del organismo de estandarización ANSI X.9, y dentro de él está el arco que apunta a todos los algoritmos que tiene normalizados, *x.9algorithm (4)*. Finalmente, aparece el *dsa-with-sha1(3)*, que corresponde al algoritmo concreto que se identifica. En definitiva, el identificador de objeto para este algoritmo criptográfico es: {1 2 840 10040 4 3}.

El tipo ObjectDescriptor tiene menor importancia, ya que se trata simplemente de añadir un texto legible por el ser humano en el que se aclara el uso o se proporciona un nombre más detallado del objeto. Se recomienda que cuando una autoridad asigne valores de identificación a un objeto, asigne también valores de tipo «ObjectDescriptor» a ese mismo objeto. Por ejemplo, en la propia norma que especifica el ASN.1 se puede encontrar una cláusula que indica:

Valor de identificador de objeto

```
{joint-iso-itu-t asn1(1) specification(0) characterString(1) numericString(0) }
```

Valor de descriptor de objeto

“NumericString ASN.1 type”

TIPOS ESTRUCTURADOS

SEQUENCE y SEQUENCE OF

El tipo SEQUENCE representa una secuencia ordenada de tipos que pueden ser distintos entre sí. La definición mostrada en la Figura 6.16 es un ejemplo de ello:

```
Ficha      Usuario      ::= SEQUENCE {
           nombre      OCTET STRING, -- 20 caracteres
           fechaDeAlta  INTEGER, -- DDMMAAAA
           autorizado   BOOLEAN,
           dni          INTEGER
           }
```

Es menester recordar que las etiquetas auxiliares sólo sirven para ayudar al lector de la especificación a entender la estructura y significado de la PDU redactada. Lo que

realmente se codifica es el *tipo*, *longitud* y *valor* de cada uno de los campos. En este caso, primero aparecerá que se trata de una SEQUENCE de tantos octetos, que el primer campo es un OCTET STRING de otros tantos octetos, y así sucesivamente.

En este caso, *FichaUsuario* vamos a suponer que es una de las posibles PDUs de la aplicación, y que existen, por ejemplo, otras siete distintas relacionadas con otros tantos aspectos de la gestión de servicios. En el extremo receptor, la aplicación dispone de las «plantillas» de las siete PDUs y cuando le llegue a través de la red un ejemplar con valores concretos (una cadena de octetos con valores numéricos), tiene que ser capaz de reconocer de qué PDU se trata y de «rellenar», con los valores recibidos, los distintos campos del formato esquemático que posee. Exactamente igual que un programa que soporte IP tendrá que ser capaz de reconocer qué tipo de paquete recibe analizando los octetos iniciales de la ristra de octetos que le llega, la aplicación receptora tendrá que ser capaz de reconocer que la estructura recibida es una PDU del tipo *FichaUsuario*, analizando el tipo inicial y el de los campos de que consta.

El tipo SEQUENCE OF es un caso particular del caso anterior en el que todos los tipos de la secuencia ordenada son iguales. Por ejemplo, en el certificado X.509, el campo de extensiones se define como:

Extensions ::= SEQUENCE OF Extension

En el que, a su vez, se define Extension como una SEQUENCE compuesta por otros tres tipos distintos donde se recoge el identificador de cada extensión, el campo booleano de criticidad y el valor propiamente dicho, que es una cadena de octetos. Significa que todas las extensiones tienen una misma plantilla general, aunque internamente cada una mantiene una estructura propia.

SET y SET OF

El tipo SET representa un conjunto de tipos, sin orden preestablecido, que pueden ser distintos entre sí. Ejemplo:

```
Usuario ::= SET {
    nombre PrintableString,
    clave IA5String }
```

El tipo SET OF es un caso particular del anterior en el que todos los tipos del conjunto son iguales y pueden aparecer sin orden preestablecido. Ejemplo:

Usuarios ::= SET OF Usuario

Donde *Usuario* es el tipo definido en el ejemplo anterior como un SET. Cuando los tipos de los campos son iguales y, además, pueden venir desordenados, no hay manera de que la aplicación receptora sea capaz de distinguir cuál de ellos está recibiendo en cada momento, por lo que será necesario, como veremos más adelante, añadir algún mecanismo complementario de identificación que permita la correcta recomposición de la PDU.

OTROS ELEMENTOS DE LA NOTACIÓN

CHOICE

Indica que el segmento de PDU especificado puede ser, indistintamente, de uno, y sólo uno, de los tipos especificados. Ejemplo:

```
Identificador ::= CHOICE {
    dni NumericString,
    nif PrintableString}
```

Ello quiere decir que en el extremo emisor, la aplicación que componga el ejemplar de PDU a transferir puede elegir conformar el *Identificador* como una ristra de números o como una cadena de caracteres alfanuméricos. La aplicación receptora tiene que estar preparada para recibir y distinguir uno de estos dos tipos distintos cuando le llegue el segmento de PDU correspondiente.

Como se recordará, en el caso del certificado X.509, hay un campo que sirve para marcar su periodo de validez. Inicialmente estaba definido usando el tipo *UTCTime*, que sólo admite dos dígitos para la representación del año. En la década de los noventa, cuando se preveía la emisión de certificados con duración suficiente como para que siguiesen siendo válidos durante el presente siglo, para poder representar el año mediante cuatro dígitos, se llegó al acuerdo de modificar la definición (de la v3), de forma que el periodo de validez fuese del tipo *Time* que se definiría como sigue:

```
Time ::= CHOICE {
    utcTime          UTCTime,
    generalizedTime  GeneralizedTime}
```

Como el tipo CHOICE no se codifica ni transmite, la aplicación que tuviese un certificado antiguo donde el tiempo fuese del tipo *UTCTime*, siempre codificaría y transmitiría este tipo, que podrá ser entendido por la aplicación receptora aunque ésta tenga la definición nueva del campo de tiempos, ya que ese es uno de los dos posibles tipos que tiene que estar preparada para recibir. Con ello se aseguraría cierta compatibilidad. Por supuesto, si la entidad tiene el certificado en versión modificada y usa el tipo *GeneralizedTime* sólo podrá ser entendida por otra entidad que espere un certificado conforme con la nueva versión.

OPTIONAL y DEFAULT

Se trata de calificativos que se añaden a los campos definidos. OPTIONAL indica que el campo es opcional, esto es, que cuando la aplicación emisora genere el ejemplar de PDU a ser transferido, ese campo puede no aparecer. Ejemplo:

```
Identificador ::= SEQUENCE {
    dni NumericString DEFAULT "",
    letraNif PrintableString OPTIONAL }
```

DEFAULT significa que la entidad receptora debe asumir por defecto el valor indicado. Al generar el segmento de PDU mostrado en este ejemplo, la aplicación emisora puede no incluir ni transmitir el primer campo de la secuencia y la entidad receptora deberá asignar el valor especificado a ese campo no recibido. Si el valor que

se le quiere asignar es distinto del que se asigna por defecto, se comportaría como un campo cualquiera al que no se le ha calificado de esta forma.

IMPLICIT y Etiquetas (*tags*)

También denominadas *rótulos* en la versión en español que publica ITU-T, las etiquetas son una denotación que siempre va asociada con todo tipo ASN.1, de forma que a cada uno de ellos le corresponde un número identificador que es el que aparece en la representación binaria a la hora de codificar la PDU. Además de existir rótulos o etiquetas por defecto, se pueden añadir otras etiquetas mediante especificación explícita. En ASN.1 se especifican cuatro clases de etiquetas:

- UNIVERSAL: Son los tipos básicos, tanto simples como estructurados. Existe una tabla en la que la norma asigna valores numéricos concretos a cada uno de los tipos.
- APPLICATION: Usada para distinguir tipos en el ámbito del contexto de presentación.
- Específica del contexto (*context-specific*): Usada para distinguir tipos en una estructura.
- PRIVATE: Usada para distinguir tipos en el ámbito de una organización.

No hay diferencia formal entre la utilización de las etiquetas de estas tres últimas clases. Cuando se empleen rótulos de clase *aplicación*, podrán emplearse también, por lo general, rótulos de clase *privada* o *específica del contexto*, dependiendo de la elección y el estilo del usuario. Las etiquetas están pensadas para que la máquina receptora sea capaz de entender y discriminar los distintos campos de una PDU. Volvamos al ejemplo del principio en el que se han añadido algunos elementos calificativos:

```
FichaUsuario ::= [APPLICATION 4] IMPLICIT SEQUENCE {
    nombre          OCTET STRING, -- 20 caracteres
    fechaDeAlta     [1] IMPLICIT INTEGER OPTIONAL, -- DDMMAAAA
    autorizado      BOOLEAN DEFAULT TRUE,
    dni             [2] IMPLICIT INTEGER OPTIONAL
}
```

Como los tres últimos campos pueden o no aparecer debido a su carácter optativo o de valor por defecto, si una máquina recibiese un ejemplar de PDU con solo el primer campo y uno de los dos enteros, no sabría discriminar cuál de ellos es el que se ha transmitido, a no ser que entrase en impertinencias de analizar el número entero para saber si tiene más pinta de fecha que de número de identificación. La etiqueta añadida resuelve contundentemente esta indefinición. Análogamente, si en esta aplicación imaginaria existiesen, como ya se ha dicho, otras siete posibles PDUs y cuatro de ellas fuesen del tipo *SEQUENCE*, al recibir un ejemplar de PDU, la máquina tendría que analizar los campos internos para averiguar de cuál de ellas se trata, es decir, bajo qué «plantilla» tiene que reconstruirla. Añadiendo la etiqueta *[APPLICATION 4]* esta tarea se simplifica, al tiempo que se resuelven posibles inconsistencias debidas a que haya dos tipos de PDU tales que en algunos casos, recibido un ejemplar, no fuese posible discriminar a cuál de los dos tipos pertenece.

El calificativo *IMPLICIT* está pensado para ahorrar código binario a la hora de representar y transmitir la información. En efecto, al añadir una etiqueta tipo *APPLI-*

CATION o específica del contexto es necesario codificar tanto la etiqueta añadida como el tipo básico al que refuerza. Por contra, incorporando *IMPLICIT*, por ejemplo, en el último campo de la PDU que estamos comentando, no es necesario codificar ni transmitir la etiqueta del tipo universal *INTEGER*. Este elemento ya no se usa en las especificaciones más modernas porque para obtener códigos binarios más compactos se recurre a nuevas reglas de codificación, como es el caso del PER, que más adelante se comenta. No obstante, se ha incluido aquí porque especificaciones realizadas hace algún tiempo sí que utilizan el calificativo *IMPLICIT* en sus definiciones. Los dos nuevos campos que se añadieron al certificado X.509 en su versión 2 son un ejemplo de ello.

MÓDULOS, MACROS Y PARAMETRIZACIÓN

De forma semejante a lo que ocurre en otro tipo de lenguajes formales, en ASN.1 un *módulo* es una especificación en la que se definen tipos y valores encapsulados que pueden ser referidos externamente mediante un identificador. Por ejemplo, el marco en el que está recogida la definición del certificado en la Recomendación X.509 está definido como un módulo identificable como:

```
AuthenticationFramework {joint-iso-itu-t (2) ds(5) module(1) authenticationFramework(7) 2}
```

En general, de forma muy resumida y simplificada, podíamos decir que un módulo ASN.1 tiene una estructura parecida a la que se presenta en la Figura 6.18. En ella se usan paréntesis angulares < > para indicar el tipo de información que en esos sitios

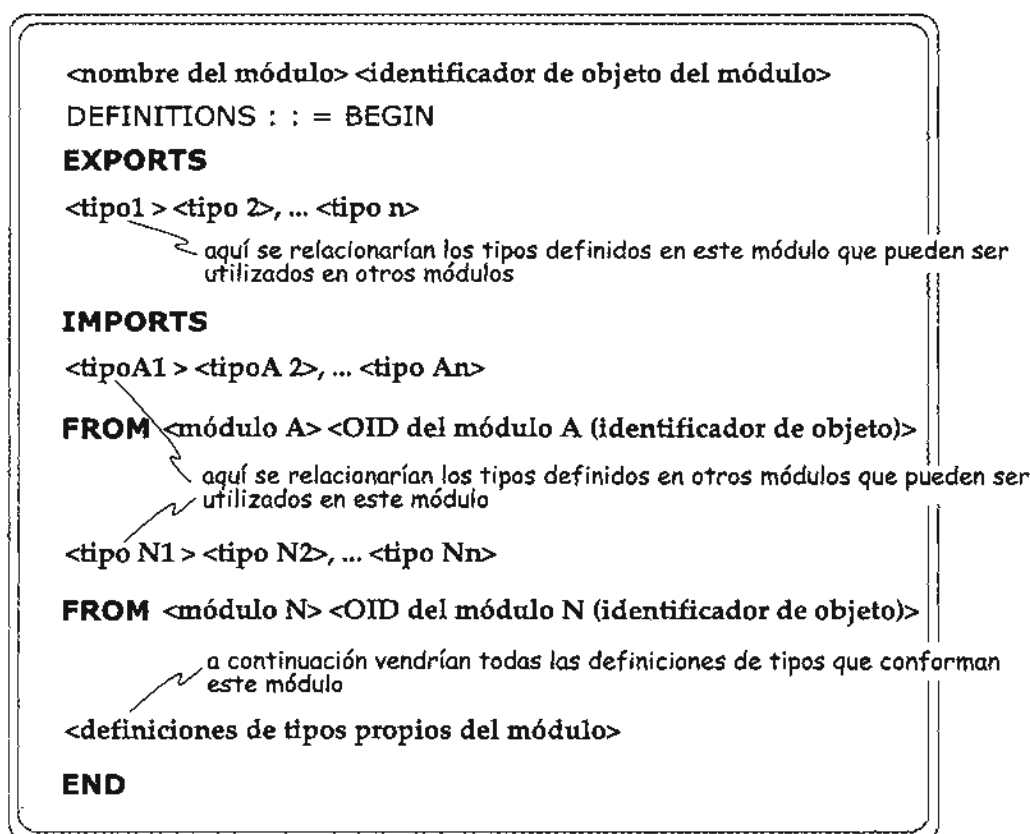


Figura 6.18. Estructura de un módulo ASN.1.

aparecería. No se corresponde con una notación formal ASN.1, sino que es un artificio didáctico para ayudar al lector a entender la estructura de un módulo, evitando la complejidad de lo especificado en la norma. Las palabras clave, escritas en mayúsculas, así como su orden de aparición, sí son, en cambio, propias de la definición de un módulo ASN.1 genérico.

Las *macros* eran una capacidad que permitía al usuario de ASN.1 definir estructuras para la especificación de objetos de información. La macro consistía en un conjunto de expresiones de tipo que podía ser invocado y heredado desde una especificación cualquiera al que, a veces, era necesario suministrarle algunos parámetros para que se convirtiera en un conjunto de tipos ASN.1 completamente definidos. En las nuevas versiones de la norma se ha suprimido su existencia y se recomienda que las especificaciones antiguas que la usaban sean corregidas, suprimiendo las macros, para posibilitar su compatibilidad con la nueva notación.

Las macros han sido sustituidas por tipos y valores *parametrizados* que introducen mejoras en lo tocante a legibilidad, precisión y flexibilidad de lo especificado. Como su nombre sugiere, un tipo parametrizado es una definición incompleta que genera una definición precisa de tipo cuando es referido (suministrando valores para sus parámetros) desde un lugar concreto de la especificación. En el apartado siguiente, al describir la especificación formal de certificado X.509, haremos referencia a varios tipos concretos de esta clase.

REGLAS DE CODIFICACIÓN

Como hemos venido insistiendo, cuando se genera un ejemplar concreto de PDU para ser almacenado o transferido a una entidad remota, es necesario materializarlo en un conjunto de unos y ceros susceptible de ser transmitido a través de un línea de comunicación. Esto es igualmente cierto cuando se trate de una aplicación definida mediante ASN.1 que cuando se trate de TCP o X.25 o cualquier otro protocolo. Por tanto, después de conocer las reglas ASN.1 mediante las cuales pueden definirse las PDUs, es necesario conocer las reglas de codificación mediante las que se obtiene la representación binaria de esa estructura concreta de datos. Esto equivale a decir, usando terminología OSI, que la aplicación de estas reglas de codificación produce una sintaxis de transferencia para estos valores. Existen cuatro posibles conjuntos de reglas para conseguirlo: BER, CER, DER y PER.

Codificación BER

Estas reglas de codificación básicas (*basic encoding rules*) tienen dos variantes. En una de ellas, que es la que nos interesa para los fines aquí perseguidos y que se denomina de *longitud definida*, la codificación se realiza (véase Figura 6.19) generando siempre tres campos con contenido numérico: *tipo*, *longitud* y *valor*. El campo *tipo* es un octeto en el que los bits 7 y 8 sirven para determinar la clase a la que pertenece la etiqueta, el bit 6 vale 1 cuando se trata de un tipo constructor, es decir, que «dentro» de lo que define hay otros tipos definidos. Caso de ser un tipo simple como *INTEGER*, *BOOLEAN*, etc., se denomina primitivo y lleva un 0 en el bit 6. En los cinco primeros bits se codifica el número binario del identificador del tipo, es decir, la etiqueta. Cuando el número que representa al identificador no cabe en cinco bits (es superior a $3E_{16}$), la norma especifica otra regla que no está recogida en el esquema de la figura.

NOMBRE DEL TIPO	ETIQUETA en hexadecimal
BOOLEAN	1
INTEGER	2
BITSTRING	3
OCTETSTRING	4
SEQUENCE	10
SEQUENCE OF	10
SET y SET OF	11
NumericString	12
IA5String	16
VisibleString	1A

Cuadro 6.1. Algunos identificadores de tipo UNIVERSAL.

- Las cadenas siempre tienen codificación primitiva.
- En el SET se mantiene el orden definido.
- El TRUE es siempre FF_{16} .
- En el BITSTRING los bits no usados van a cero.
- En los campos DEFAULT el valor por defecto nunca se transmite.

Codificación CER

La principal diferencia con la anterior es que las reglas de codificación canónica (*canonical encoding rules*) utilizan la forma de longitud indefinida en mayor medida que de la longitud definida. Por ello, las reglas DER son más adecuadas para los valores codificados pequeños, mientras que las CER son más adecuadas para valores grandes. Las restricciones con respecto al BER son que la longitud es definida, ocupando el mínimo número de octetos, en los tipos primitivos, y es indefinida en los tipos construidos. Además, las cadenas tienen codificación primitiva cuando ocupan 1.000 octetos como máximo y codificación construida cuando ocupan más de 1.000 octetos. Por otra parte, las reglas del SET, TRUE, BITSTRING y DEFAULT son idénticas a las del DER.

Codificación PER

La codificación empaquetada (*packet encoding rules*), para conseguir una codificación más compacta, trata de eliminar las redundancias existentes en la codificación básica, de forma que el número de octetos resultantes sea muchísimo menor. En este caso no se transmite el tipo, por lo que el orden de aparición de los distintos campos es mucho más crítico que en las anteriores codificaciones. Por ello, al no codificar el tipo, para

resolver los problemas que podrían plantearse con los campos opcionales en un tipo estructurado, aparece un campo «preámbulo» en el que se indican qué campos están presentes. Hay una modalidad en la que la alineación de los campos es a nivel de bit en lugar de ser el octeto la unidad mínima de representación.

Ejemplo de codificación con DER

Dado el tipo *Identificacion* definido como se indica, se trata de realizar la codificación, siguiendo las reglas DER, del ejemplar *miIdentificacion* cuyos valores se especifican. Las definiciones son:

```
Identificacion ::= SEQUENCE {
    nombre                [0] VisibleString
    claveNumerica         [1] IMPLICIT NumericString OPTIONAL
    palabraDePaso         [2] IMPLICIT IA5String OPTIONAL
    nivelSeguridad        [3] IMPLICIT INTEGER DEFAULT 0
}
miIdentificacion Identificacion ::= {
    nombre                "aplitel05"
    palabraDePaso         "miklave1"
    nivelSeguridad        2}
```

Siguiendo el esquema marcado en la Figura 6.19 (que también sirve para este caso), iremos deduciendo los valores numéricos que se recogen en la Figura 6.20 como resultado de la codificación. En primer lugar aparece *Identificacion* para el que se reservan los tres campos de tipo (*Identificacion*), longitud (*long1*) y valor. Por ser de tipo *SEQUENCE*, los bits 8 y 7 del octeto de tipo adquieren el valor 00 correspondientes a *UNIVERSAL*; el bit 6 vale 1 por tratarse de un tipo constructor que engloba a los restantes campos tipados de la estructura; y los cinco bits de menor peso valdrán 10000_2 correspondientes al 10_{16} del identificador de *SEQUENCE* (véase Cuadro 6.1). Es decir, el valor del octeto será $00110000_2 = 30_{16}$. En este momento no podemos conocer aún el valor del campo *long1*, que es igual al número de octetos de toda la estructura. El campo de valor está constituido por la secuencia de los valores de los tipos que engloba la *SEQUENCE*.

El primero de ellos corresponde a *nombre* con su etiqueta [0]. En este caso los bits 8 y 7 del octeto de tipo valen 10_2 , por ser específico del contexto. El bit 6 vale 1 porque al englobar al siguiente tipo (*VisibleString*), se considera como constructor, siendo 0 el identificador de etiqueta. Por tanto, el valor del octeto de tipo es $10100000_2 = A0_{16}$. El campo *long2* se corresponde con todos los datos englobados en *nombre*, por lo que su valor se conocerá algo más tarde. El campo valor está constituido por la siguiente terna tipo, longitud y valor. En este caso la codificación del tipo es la correspondiente a *VisibleString*, un tipo *UNIVERSAL*, primitivo, con etiqueta $1A_{16}$ (Cuadro 6.1), por lo que el octeto de tipo es $00011010_2 = 1A_{16}$. El campo de valor lo constituyen nueve octetos conteniendo cada uno de ellos el código de uno de los caracteres «aplitel05» (para que sea más inteligible la figura, no se refleja aquí el código binario de cada uno de ellos y se expresa con el conjunto de caracteres entrecomillado). Como consecuencia, el campo *long3* es un octeto de valor 09_{16} , lo que conlleva que *long2* valga $0B_{16}$, resultado de sumar 2 a 09_{16} , para tener en cuenta los otros dos octetos: el de tipo (cuyo contenido es $1A_{16}$) y el propio *long3* (con 09_{16} de contenido).

TIPO:						
<u>Identificacion</u>	<u>long1</u>	<u>valor</u>				
30	1A	Tipo:				
		<u>nombre</u>	<u>long2</u>	<u>valor</u>		
		A0	0B	VisibleString	long3	valor
				1A	09	"apitel05"
		Tipo:				
		<u>palabraDePaso</u>	<u>long4</u>	<u>valor</u>		
		82	08	"miklave1"		
		Tipo:				
		<u>nivelSeguridad</u>	<u>long5</u>	<u>valor</u>		
		83	01	02		

Figura 6.20. Ejemplo de codificación (datos en hexadecimal).

El campo *claveNumerica*, como es opcional y no tiene asignado ningún valor en la especificación de *milIdentificador*, no se codifica ni transmite.

En el campo *palabraDePaso*, sólo se codifica la etiqueta [2] (como primitiva y no como tipo constructor) porque el *IA5String* tiene el calificativo de IMPLICIT. Se da por supuesto que la entidad receptora conoce su existencia y, como ya se ha dicho, sólo sirve para reducir y compactar la codificación. Aunque se recomienda dejar de usarlo, se ha incluido aquí porque aparece en muchas especificaciones vigentes (por ejemplo en el certificado X.509). Siguiendo un procedimiento análogo al explicado en los dos casos anteriores, se obtienen los correspondientes valores para sus tres campos.

Por último, el campo *nivelSeguridad* tiene que ser codificado porque el valor que se le ha asignado es distinto de 0. Si fuese 0, con una codificación BER sería válido tanto codificarlo como no hacerlo; pero no así con codificación DER, donde el valor por defecto nunca se transmite.

Una vez completados los tres tipos «interiores», se puede calcular el valor del octeto *long1*, que resultará de sumar $0B_{16} + 08_{16} + 01 + 6 = 1A_{16}$. Los últimos seis octetos son los que se corresponden con los octetos que contienen *Long2*, *Long4* y *Long5*, más los respectivos octetos de identificación de tipo.

Es decir, el valor en hexadecimal de la ristra de octetos resultante (manteniendo en entrecomillado la representación de los caracteres) sería:

30 1A A0 0B 1A 09 "apitel05" 82 08 "miklave1" 83 01 02

PARA QUÉ SIRVE ASN.1

A modo de conclusiones, podríamos decir que la utilización que podemos hacer de ASN.1 y el grado de conocimiento que sobre él necesitamos tener depende, como casi todo en tecnología, de las circunstancias concretas en las que nos encontremos.

De una parte, de la misma forma que para tener un conocimiento aproximado del comportamiento de TCP, por ejemplo, es imprescindible ser capaz de entender la estructura de la PDU recogida en la Figura 6.15, cualquier persona que quiera ser capaz de tener acceso al contenido de normas recientes del nivel de Aplicación, o

simplemente a resúmenes de sus características más relevantes (como puede ser este libro) necesita entender, siquiera sea por encima, la estructura de las PDUs que allí se definen y entender la semántica de los distintos campos de que constan. Esta idea es válida para muchas áreas de la telemática, pero es especialmente cierta para temas relacionados con la Seguridad en Redes. Este nivel de conocimiento de ASN.1, que podríamos calificar de *«ser capaz de leer y entender aproximadamente una especificación»*, puede adquirirse con una lectura rápida y no excesivamente meditada del resumen que se presenta en los párrafos precedentes.

Un segundo nivel de conocimiento puede cifrarse en ser capaz de entender con absoluto rigor una especificación y ser capaz de redactar, usando ASN.1, una nueva PDU o estructura de datos a ser manejada en una determinada aplicación (estándar o no). Este es el nivel que requiere un ingeniero telemático que necesite trabajar en el desarrollo de sistemas que manejen este tipo de datos. Para llegar a ello, una vía plausible puede ser empezar por entender con puntualidad y precisión el resumen presentado en este epígrafe y, posteriormente, estudiar detenidamente sobre las normas aquellos puntos que requieran aclaración y detalle. No nos cansaremos de insistir en que, en este como en cualquier otro asunto, cuando llega la hora de la verdad, la mejor referencia es una versión actualizada de la norma. (Puestos a equivocarse, mejor es que nos equivoquemos nosotros que no que nos equivoquen otros.)

Otra situación, que puede presentarse en paralelo con la descrita en el párrafo anterior, es la relacionada con el manejo preciso de la representación de los datos resultante de la codificación de una determinada PDU. A pesar de la extremada sencillez del ejemplo de codificación que se muestra en la Figura 6.20, puede apreciarse que la labor de codificación-descodificación no es tarea fácil. El ejemplo mostrado (o su inverso, «dada una ristra de octetos recibidos, averiguar de qué PDU se trata y el valor de sus campos») pueden ser interesantes para un ejercicio de clase (o de examen), pero resulta evidente que es impracticable para el funcionamiento, en vivo y en directo, de una aplicación telemática. Para ello existen traductores automáticos que ayudan a las entidades comunicantes a interpretar y codificar esas estructuras de datos.

Igualmente, llegado el caso de ser necesario el uso de analizadores de protocolos a intercalar en medio de una línea de comunicación, también habrá herramientas conocedoras de la sintaxis y de la semántica de las PDUs allí manejadas que nos permitirán entender el comportamiento que se está produciendo y nos ayudarán a corregir los posibles fallos. No obstante, a veces (pocas) esas herramientas no entienden adecuadamente lo que está sucediendo, o no es posible intercalarlas en determinados sitios, y el ingeniero telemático no tiene más remedio que realizar una captura binaria pura, y echar mano de papel y lápiz para, reconstruyendo la PDU, intentar averiguar lo que allí está sucediendo. Para estos casos, ese ingeniero necesita tener un conocimiento bastante preciso de las reglas de codificación.

6.6. DEFINICIÓN FORMAL DEL CERTIFICADO X.509

Una vez conocida la notación formal ASN.1, trataremos de formalizar la estructura del certificado que se estudió en el epígrafe 6.4. Cabe decir que, como en otras ocasiones, no se pretende aquí ni sustituir ni suplir la norma, en este caso a la ISO/IEC 9594-8 o la correspondiente RFC 3282, que serán la fuente de referencia obligada de quienes necesiten una información actualizada para abordar problemas de diseño e implementación de aplicaciones telemáticas seguras (la recomendación equivalente ITU-T es la X.509). Debido a ello, se presenta aquí, con fines didácticos, una selección de los

elementos más significativos que constituyen la definición formal del certificado. No se incluye, por tanto, la especificación de todas y cada una de las extensiones que ya hemos descrito. En [Lope98] se puede encontrar la especificación ASN.1 de casi todas las extensiones acompañada de comentarios acerca de la evolución que han tenido a lo largo del tiempo los distintos campos de que constan, así como referencias a ciertas particularidades y diferencias que aparecen en otras definiciones del certificado, más concretamente en las del grupo IETF-PKIX de Internet.

La definición de la versión 3 del Certificado X.509, que, como ya se ha dicho, engloba a las otras dos anteriores, es la siguiente:

```
Certificate ::= SIGNED SEQUENCE {
version                [0] Version DEFAULT v1,
serialNumber           CertificateSerialNumber,
signature              AlgorithmIdentifier,
issuer                  Name,
validity                Validity,
subject                 Name,
subjectPublicKeyInfo    SubjectPublicKeyInfo,
issuerUniquelIdentifier [1] IMPLICIT UniquelIdentifier OPTIONAL,
                        -- si está presente, la versión tiene que ser v2
                        -- o v3
subjectUniquelIdentifier [2] IMPLICIT UniquelIdentifier OPTIONAL,
                        -- si está presente, la versión tiene que ser v2
                        -- o v3
extensions             [3] Extensions OPTIONAL
                        -- si está presente, la versión tiene que ser v3-- }
```

Obviamente, la especificación formal es única e independiente del idioma en que se edite la norma, pero los comentarios (las frases precedidas de --) pueden aparecer en español, tal y como se ha recogido en esta definición. Pasemos a comentarla pormenorizadamente.

Como no podía ser de otra forma, aparece la sintaxis de los campos cuyo significado y valores habíamos resumido en los comentarios a la Figura 6.13. Para empezar nos encontramos con el tipo SIGNED, que no está recogido en la descripción de ASN.1 presentada en el anterior apartado. Esto es así porque se trata de un tipo parametrizado definido explícitamente en la norma. Es el último eslabón de una serie de nuevos tipos que fue necesario crear cuando se formuló por primera vez la definición del certificado (aunque en la versión de 1988 estaban definidos como *macros*, formulación que fue posteriormente sustituida por la parametrización). Estos tipos, aunque estén definidos en el ámbito de la norma X.509, pueden ser utilizados desde cualquier especificación ASN.1. Esto es así porque en el módulo en que están definidos se ha indicado explícitamente (mediante la indicación EXPORT) que pueden ser importados desde cualquier módulo ASN.1. Veamos su definición:

TIPOS ENCRYPTED, HASHED, SIGNATURE Y SIGNED

Los mecanismos de seguridad que utilizan técnicas criptográficas necesitan de la definición de unos tipos especiales que posibiliten la especificación de tales mecanismos. Salvo el primero de ellos, las reglas de codificación utilizadas para estos nuevos tipos son las DER, que dejan mejor acotado el tamaño de los datos que han de ser transformados por los algoritmos.

ENCRYPTED

Su definición formal es la siguiente:

```

ENCRYPTED { ToBeEnciphered } ::= BIT STRING (CONSTRAINED BY {
    -- debe ser el resultado de aplicar un procedimiento de cifrado a los octetos
    -- con codificación BER de un valor – ToBeEnciphered
)}

```

La notación CONSTRAINED sirve para restringir o constreñir los valores de cualquier tipo a unas circunstancias de uso particulares. En este caso lo que ha de cifrarse debe ser el resultado de la codificación BER de lo que se desee cifrar. El tipo ENCRYPTED es, por tanto, un constructor. Si lo que se desea cifrar es, por ejemplo, una *password* compuesta por una cadena de caracteres, primero debe codificarse esa cadena mediante las reglas BER obteniendo como resultado la correspondiente terna *etiqueta de tipo, longitud y valor* (como en el ejemplo de la Figura 6.19). A continuación esa terna es la que se cifra con un algoritmo concreto, dando como resultado una cadena de bits. Finalmente, la codificación del constructor ENCRYPTED da lugar a una nueva terna *etiqueta de tipo, longitud y valor*, donde *valor* es la cadena de bits resultante del cifrado de la palabra de paso.

HASHED

Es un constructor diseñado para codificar el resultado de la operación resumen (*hashing*) de un mensaje (ya comentado en el Capítulo 5 cuando se introdujo el concepto de firma digital). El tamaño del resumen es fijo y dependiente del algoritmo utilizado. Su definición es:

```

HASHED { ToBeHashed } ::= OCTET STRING (CONSTRAINED BY {
    -- debe ser el resultado de aplicar un procedimiento hashing a los octetos
    -- con codificación DER de un valor – ToBeHashed
)}

```

Como en el caso anterior, el resultado final de su codificación es una terna *etiqueta de tipo, longitud y valor*, con la particularidad de que aquí el campo de valor es una ristra de octetos y que el mensaje al que se ha de aplicar el hashing debe estar previamente codificado conforme a las reglas DER.

SIGNATURE

Es un constructor que especifica la firma digital de un documento o pieza de información e incorpora en su definición otros dos constructores (el de cifrado y el de hashing). Además lleva una referencia explícita al algoritmo con el que se realiza la firma digital, ya que dependiendo de cuál sea éste, así será la operación que se realice con el cifrado de los datos y con el resumen del mensaje a firmar. Su definición es:

```

SIGNATURE { OfSignature } ::= SEQUENCE {
    algorithmIdentifier    AlgorithmIdentifier
    encrypted              ENCRYPTED {HASHED { OfSignature }}
}

```


ESPECIFICACIÓN DE LOS CAMPOS DEL CERTIFICADO

Si se compara la definición del certificado antes presentada (tanto en la forma utilizada por ISO como la utilizada por el PKIX WG) con la representada en la Figura 6.13 se observa que los distintos campos de la SEQUENCE se corresponden con los campos cuyo significado se explicó al comentar dicha figura. Cabe ahora comentar brevemente la especificación de esos campos que recoge la norma.

El campo **version** no ofrece duda alguna. Si se omite al darle valores a un ejemplar concreto de certificado, es que se trata de la v1. En la definición ASN.1 recogida en la norma, el correspondiente tipo **Version** aparece especificado como un tipo enumerado que puede tomar los valores 0, 1 o 2, según sea la versión:

```
Version ::= INTEGER { v1(0), v2(1), v3(2) }
```

También resulta inmediato entender el significado del segundo campo, **serial-Number**, que representa el número de serie que le deberá poner la CA a cada uno de los certificados que expida y está definido mediante el tipo **CertificateSerialNumber**:

CertificateSerialNumber ::= INTEGER

El campo **signature** está definido por el tipo **AlgorithmIdentifier**, que en la norma se define como una SEQUENCE en la que se incluye el identificador del algoritmo usado para la firma y unos parámetros adicionales opcionales:

```

AlgorithmIdentifier ::= SEQUENCE {
    algorithm          OBJECT IDENTIFIER,
    parameters        ANY DEFINED BY algorithm OPTIONAL }

```

Los campos **issuer** y **subject** reflejan, respectivamente, el nombre de la CA emisora y de la entidad a la que se expide el certificado. El tipo **Name** con el que se definen está aquí heredado de otra definición y es un DN (nombre distintivo X.500), compuesto a su vez por una secuencia de RDNs (*Relative Distinguished Name*), como ya se comentó en el epígrafe 6.4. El tipo de los nombres que aparecen en las extensiones puede ser, como ya se dijo en los comentarios a la Figura 6.14, más versátil.

El campo **validity** representa el periodo de validez del certificado y en la X.509 el tipo que lo define está recogido como:

```
Validity ::= SEQUENCE {
    notBefore      UTCTime,
    notAfter       UTCTime }
```

Sin embargo, la definición recogida por el PKIX WG (y en otras propuestas) es más genérica y bastante compatible con versiones anteriores por utilizar la notación CHOICE: al asignársele valores a un ejemplar concreto de certificado se elegirá una y sólo una de las opciones. De esta forma, según vimos en el epígrafe 6.5, es posible reflejar la fecha con dos dígitos para el año, UTCTime, o con cuatro dígitos, caso que se da si se emplea GeneralizedTime. Esta definición es la siguiente:

Validity ::= SEQUENCE {
 notBefore Time,
 notAfter Time }

```
Time ::= CHOICE {
    utcTime        UTCTime,
    generalTime    GeneralizedTime }
```

El campo **subjectPublicKeyInfo** representa la información sobre la clave pública de la entidad a la que se le expide el certificado. Constituye, por tanto, el objeto principal del certificado. El tipo que lo define se especifica en la norma como una secuencia constituida por dos componentes: el identificador de algoritmo (el tipo **AlgorithmIdentifier** es el mismo que el que define al campo **signature**) y el valor de la clave pública que se está certificando. Esta especificación es la siguiente:

```
SubjectPublicKeyInfo ::= SEQUENCE {
    algorithm        AlgorithmIdentifier
    subjectPublicKey  BIT STRING }
```

En cuanto a los dos siguientes campos, que fueron introducidos en la versión 2 (pero que no han tenido aceptación alguna), no merece la pena añadir más de lo que ya se dijo en el epígrafe 6.4 al describir las dos primeras versiones del certificado. Aunque estén en la foto, al estar definidos como **OPTIONAL**, si no son utilizados no aparecen en el ejemplar de certificado emitido. De hecho el PKIX WG impone que las CAs que pretendan ser conformes con su estándar no deben utilizar estos dos campos en los certificados que emitan.

El campo **extensions**, por su importancia, merece que lo veamos en un apartado específico.

EL CAMPO DE EXTENSIONES

La especificación formal de este campo, cuya semántica y utilización ya se discutió en el correspondiente apartado del epígrafe 6.4 y en los comentarios a la Figura 6.14, es como sigue:

El tipo con que la define ITU-T en la especificación completa del certificado es **Extensions** y es de carácter opcional. Es decir, si no se utiliza, el formato del certificado coincide con el de las versiones anteriores. En la especificación completa del certificado se define este tipo como:

```
Extension ::= SEQUENCE {
    extnId      EXTENSION.&id ({ExtensionSet}),
    critical    BOOLEAN DEFAULT FALSE,
    extnValue   OCTET STRING
    -- contiene una codificación DER de un valor de
    -- tipo &ExtnType
    -- para el objeto extension identificado por extnId --
```

Lo primero que aparece es algo que ya habíamos citado: el campo **extensions** está constituido por una secuencia de extensiones, todas del mismo tipo (**Extension**). Cada una de éstas, a su vez, se define mediante una secuencia de tres campos de tipo distinto. El primero de ellos contiene un identificador de objeto (OID, Object Identifier) que determina de forma no ambigua el tipo de extensión de que se trata, el segundo campo es el que marca la criticidad (TRUE o FALSE) de la extensión concreta, y el tercero contiene el valor propiamente dicho de la extensión.

El tipo **EXTENSION.&id** con el que está definido el primero de los campos, **extnId**, está especificado como un campo de una *clase objeto de información*, **CLASS**, que sirve para definir las características generales de un conjunto de objetos (en este caso, las extensiones). **WITH SYNTAX**, **SYNTAX** e **IDENTIFIED BY** son también palabras reservadas utilizadas en este tipo de esquemas de formato (este elemento de notación no se incluyó en el resumen de ASN.1 recogido en 6.5). La definición de esta clase es la siguiente:

```
EXTENSION ::= CLASS {
    &id          OBJECT IDENTIFIER UNIQUE,
    &ExtnType }
WITH SYNTAX {
    SYNTAX &ExtnType
    IDENTIFIED BY &id }
```

Las tres últimas líneas de la especificación marcan la forma mediante la cual deben definirse todas y cada una de las extensiones concretas. Según reza en el comentario del tercer campo del tipo **Extension** (con E mayúscula y el resto minúsculas) más arriba descrito, el tipo **&ExtnType** que aparece en la definición de la clase de objeto es el que marca la sintaxis concreta con que estará definida una extensión determinada.

(Para no complicar más la descripción, vamos a dejar sin explicar el significado del paréntesis ({ExtensionSet}) que aparece a la derecha del primer campo del tipo **Extension**. No es importante para el entendimiento y uso de las extensiones y además su definición detallada ha sido postergada.)

La cosa es tan sencilla como la describimos en los comentarios a las Figuras 6.13 y 6.14, aunque la descripción formal que acabamos de presentar resulta algo dura de entender por la reiterada aparición de la palabra «extension» en minúscula, en mayúsculas, en plural, en singular, etc., que produce cierto lío en la mente de quien lo lee (las especificaciones formales suelen ser claras y no ambiguas, pero poco amables). Para tratar de ver la poca complicación del asunto vamos a fijarnos en la definición de una extensión sencillita de entender, tanto por su sintaxis como por su semántica. La elegida es *Uso de la clave* (*Key Usage*) que presentamos con el número de orden 3 del primer grupo de extensiones descritas en el apartado «Extensiones del certificado X.509 v3» del epígrafe 6.4. Su definición ASN.1 es la siguiente:

```
keyUsage EXTENSION ::= {
    SYNTAX    KeyUsage
    IDENTIFIED BY id-ce-keyUsage }
KeyUsage ::= BIT STRING {
    digitalSignature      (0),
    nonRepudiation        (1),
    keyEncipherment       (2),
    dataEncipherment      (3),
    keyAgreement          (4),
    keyCertSign           (5),
    cRLSign               (6) }
```

Lo primero que vemos es que es del tipo **EXTENSION** la clase que define estos objetos de información. El primer campo dice que la sintaxis (véase la definición de la clase) está marcada por **KeyUsage**, que es del tipo **BIT STRING** uni-

versal, de forma que será una cadena de bits identificables por su número de orden (0,1, etc.) que permiten marcar, cuando su valor es 1, uno o varios de los usos previstos para la clave certificada. Es decir, este BIT STRING se corresponde con el **&ExtnType** definido en la clase, cuya codificación DER será la que ocupe el tercer campo (OCTET STRING) de la definición general de la extensión, el campo que, como decíamos más arriba, «contiene el valor propiamente dicho de la extensión».

El segundo campo (IDENTIFIED BY) es el que marca la identificación de objeto de esta extensión concreta, **id-ce-keyUsage**, que se corresponde con el **&id** de la clase y con el primer campo (**extnId**) de la definición general de la extensión.

En este apartado no se va a incluir la especificación ASN.1 de todas las extensiones que se describieron en el epígrafe 6.4. Lo que aquí se ha procurado es ayudar a entender la estructura formal que tienen y la significación de los distintos campos. Quienes necesiten la especificación detallada de ellas deberán acudir a la norma correspondiente. No obstante, a modo de ejemplo, vamos a comentar otras dos extensiones en cuya definición aparecen conceptos de interés.

La primera de estas extensiones es *Nombre alternativo de la entidad propietaria* (*Subject Alternative Name*), que ya presentamos en el apartado 6.4. Su descripción formal es la siguiente:

```
subjectAltName EXTENSION ::= {
    SYNTAX          GeneralNames
    IDENTIFIED BY    id-ce-subjectAltName }
```

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

```
GeneralName ::= CHOICE {
    otherName          [0]  INSTANCE OF OTHER-NAME,
    rfc822Name         [1]  IA5String,
    dNSName            [2]  IA5String,
    x400Address        [3]  ORAddress,
    directoryName      [4]  Name,
    ediPartyName       [5]  EDIPartyName,
    uniformResourceIdentifier [6] IA5String,
    iPAddress          [7]  OCTET STRING,
    registeredID       [8]  OBJECT IDENTIFIER }
```

OTHER-NAME ::= TYPE-IDENTIFIER

Puede observarse cómo los dos campos SYNTAX e IDENTIFIED BY tienen el mismo formato que en la definición de la extensión anterior, sólo que particularizados a la sintaxis de la SEQUENCE **GeneralNames** y al identificador concreto de esta extensión. El interés que presenta esta extensión consiste en que hace uso del tipo **GeneralName**, que sirve para elegir otro nombre distinto del DN o nombre X.500. Este tipo es usado desde otras muchas extensiones. En este caso la entidad propietaria del certificado puede tener uno o varios de estos nombres.

Este nuevo tipo fue introducido en la especificación del certificado X.509 al generar la versión 3, para resolver el problema de asignación de nombre a entidades no registradas en el Servicio de Directorio. Por esta razón, si en el campo del certificado reservado al nombre de la entidad propietaria del certificado no aparece su DN X.500, esta extensión debe ser crítica. En la definición de la extensión puede observarse que

el nombre alternativo no está limitado a la lista especificada, sino que con el campo **otherName** se deja la puerta abierta a la inclusión de otro tipo de nombre que pueda ser de interés en un dominio de seguridad concreto. (Para simplificar, en la especificación presentada se ha omitido la definición del tipo **EDIPartyName**, que está compuesto por una **SEQUENCE** con varios elementos.)

Otra extensión que vamos también a presentar a título de ejemplo es una relacionada con políticas de certificación. Se trata de la extensión *Correspondencias de políticas* (*Policy Mappings*):

```

policyMappings EXTENSION ::= {
    SYNTAX                                PolicyMappingsSyntax
    IDENTIFIED BY                        id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
    issuerDomainPolicy                    CertPolicyId,
    subjectDomainPolicy                  CertPolicyId }

CertPolicyId ::= OBJECT IDENTIFIER

```

Como ya se comentó, esta extensión sólo se usa en certificados cuya propietaria sea una CA emisora de certificados. Es una de las múltiples extensiones que hacen referencia a políticas de seguridad. Mediante las dos secuencias anidadas se especifican las políticas definidas en otros dominios que son compatibles con la de la CA propietaria del certificado donde aparece esta extensión. Puede observarse que las políticas de certificación están referenciadas mediante un identificador de objeto (**OBJECT IDENTIFIER**) que permite acceder de forma no ambigua a la información donde se describe dicha política. Otra cosa es el grado de formalidad que pueda existir en esa descripción, que será determinante a la hora de automatizar los procesos. Pero este aspecto lo veremos más adelante, cuando se aborde de forma genérica el tema de las Políticas de seguridad.

El PKIX Working Group, en la RFC 3282 *Certificate and CRL Profile* antes citada, utiliza una especificación mucho más sencilla para describir tanto el campo **extensions** del Certificado X.509 como las extensiones propiamente dichas. Quien entienda la especificación y los ejemplos que hemos explicado en los últimos párrafos no tendrá ninguna dificultad para entender de corrido la descripción de las extensiones que hace el PKIX WG. Aquí hemos optado por describir la especificación de ITU-T por ser más genérica y más difícil de entender, pretendiendo, así, servir de ayuda a quienes tengan que utilizar esa norma.

UN EJEMPLO DE CERTIFICADO X.509 V3

Un certificado concreto es simplemente una estructura de datos organizada en una serie de campos que, a su vez, están descompuestos en diversas partes, todo ello conforme a una sintaxis bien definida. En último extremo, esa estructura estará representada por un conjunto de bits (o si se prefiere, de octetos), que será lo que viaje por la línea de comunicación y lo que se almacene en la memoria de un computador o de una tarjeta inteligente. En la Figura 6.21 se muestra el volcado hexadecimal de un certificado generado por una determinada CA, en un momento preciso, a nombre de un titular claramente identificado, constituido por 760 octetos. Antes de cada fila se ha añadido un número decimal (para mayor sencillez) que indica el número de orden que ocupa el primero de los octetos de esa fila.

```

0  (300 82 02 f4 (301 82 02 b2 (a01a 03 02 011a 02) (021b 10 7f 00 00 01 00
20 00 02 7c 00 00 00 01 00 00 00 1b 14) (301c 0b 06 07 2a 86 48 ce 38
40 04 03 05 1c 00) (301d 81 8a 31 0b 30 09 06 03 55 04 06 13 02 45 53
60 31 0f 30 0d 06 03 55 04 08 13 06 4d 41 44 52 49 44 31 0f 30
80 0d 06 03 55 04 07 13 06 4d 41 44 52 49 44 31 0c 30 0a 06 03
100 55 04 0a 13 03 55 50 4d 31 0f 30 0d 06 03 55 04 0b 13 06 44
120 49 41 54 45 4c 31 12 30 10 06 03 55 04 03 13 09 44 49 41 54
140 45 4c 20 43 41 31 26 30 24 06 09 2a 86 48 86 f7 0d 01 09 01
160 16 17 77 65 62 6d 61 73 74 65 72 40 64 69 61 74 65 6c 2e 75
180 70 6d 2e 65 1d 73) (301e 1e 17 0d 30 31 30 38 32 35 30 38 33 32 32
200 34 5a 17 0d 30 34 30 38 32 36 30 38 31 33 31 32 1e 5a) (301f 81 ab
220 31 0b 30 09 06 03 55 04 06 13 02 45 53 31 0f 30 0d 06 03 55
240 04 08 13 06 4d 41 44 52 49 44 31 0f 30 0d 06 03 55 04 07 13
260 06 4d 41 44 52 49 44 31 0c 30 0a 06 03 55 04 0a 13 03 55 50
280 4d 31 0f 30 0d 06 03 55 04 0b 13 06 44 49 41 54 45 4c 31 18
300 30 16 06 03 55 04 03 13 0f 4a 75 73 74 6f 20 43 61 72 72 61
320 63 65 64 6f 31 26 30 24 06 09 2a 86 48 86 f7 0d 01 09 01 16
340 17 63 61 72 72 61 63 65 64 6f 40 64 69 61 74 65 6c 2e 75 70
360 6d 2e 65 73 31 19 30 17 06 0a 09 92 26 89 93 f2 2c 64 01 01
380 13 09 63 61 72 72 61 63 65 1f 64) (6f 1g 30 81 9f 30 0d 06 09 2a 86
400 48 86 f7 0d 01 01 01 05 00 03 81 8d 00 30 81 89 02 81 81 00
420 a6 b7 0d 04 ae f7 76 74 57 1b ef 73 36 de 67 64 f6 77 8f 50
440 39 0f 55 be 68 85 c1 35 5a a7 30 28 c8 0a f1 5f e9 6e 2e f6
460 ef f3 31 88 b3 8a 1e de b5 4d 4b f1 39 5b 59 19 2c 44 94 28
480 01 09 4f 85 4a 5f 36 b4 ea 75 bc 3b ee 2c 17 f6 25 c5 ec 9d
500 17 f6 57 cd 8e 7f ea aa 41 17 dd eb e9 12 7a c3 7b 47 5a 98
520 b3 33 f5 5b f3 03 2b 00 16 71 fe df 76 83 fd c4 5e a5 2a 4b
540 9b 23 50 a7 00 4c 4e 63 02 03 01 00 1g 01) (a3 1h 81 8e 30 81 8b 30
560 0b 06 03 55 1d 0f 04 04 03 02 04 90 30 42 06 03 55 1d 11 04
580 3b 30 39 81 17 63 61 72 72 61 63 65 64 6f 40 64 69 61 74 65
600 6c 2e 75 70 6d 2e 65 73 86 1e 68 74 74 70 3a 2f 2f 63 61 72
620 72 61 63 65 64 6f 2e 64 69 61 74 65 6c 2e 75 70 6d 2e 65 73
640 30 38 06 03 55 1d 21 01 01 ff 04 2e 30 2c 30 16 06 09 2a 86
660 48 86 f7 0d 01 09 10 06 09 2a 86 48 86 f7 0d 01 09 10 30 12
680 06 07 2a 81 07 70 02 08 07 06 07 2a 81 07 70 02 08 1h 07) (302 0b
700 06 07 2a 86 48 ce 38 04 03 05 2 00) (03 3 2f 00 30 2c 02 14 33 42
720 4b 2c c4 82 13 ac d6 d6 f5 5a a2 d6 c8 8f 94 37 63 73 02 14
740 08 f6 f2 a2 ba d1 d7 79 25 91 c3 0f 82 ec b8 19 3c 12 b9 51
760 51 3)

```

0	Certificate	1a	version	1e	validity
1	tbsCertificate	1b	serialNumber	1f	subject
2	signatureAlgorithm	1c	signature	1g	subjectPublicKeyInfo
3	signatureValue	1d	issuer	1h	extensions

Figura 6.21. Volcado hexadecimal de un certificado.

Es evidente que un certificado así representado es, a simple vista, un galimatías difícilmente interpretable por un ser humano corriente, lo cual no quiere decir que no sea susceptible de ser interpretado. De hecho, teniendo a mano la «plantilla» definitoria del certificado X.509 v3 y utilizando las escasas nociones que se presentan en el apartado «Reglas de codificación» del epígrafe 6.5 sería posible reconstruir, pieza por pieza, el valor de todos y cada uno de sus campos. No vamos a hacer eso aquí, pero sí vamos a ver (Figura 6.21) cómo se localizan algunos de ellos.

En la codificación del certificado se emplean las reglas DER, que, como antes se vio, son idénticas a las BER en cuanto a la resolución del problema que nos ocupa (obtener la especificación a partir de la representación binaria). En la Figura 6.21 se han marcado los distintos campos a partir de la definición del certificado usada en la RFC 3282, que más arriba recogíamos y que está definido como:

```
Certificate ::= SEQUENCE {
    tbsCertificate          TBSCertificate,
    signatureAlgorithm    AlgorithmIdentifier,
    signatureValue        BIT STRING }
```

Así, el primer octeto contiene un 30_{16} que se corresponde con la SEQUENCE inicial del certificado y se ha marcado con un «0», que en la leyenda de la parte inferior de la figura se indica que se corresponde con el *Certificate* inicial. El segundo octeto indica (véase Figura 6.19) que la longitud es superior a $7F_{16}$ y que la longitud está contenida en los dos octetos siguientes. Esta longitud es, por tanto, $02F4_{16}$, es decir, 756 octetos (los 760 menos los cuatro iniciales). El siguiente octeto (30_{16}) será, por tanto, el código del primero de los campos, *tbsCertificate* (marcado con «1» en la Figura 6.21) del tipo *TBSCertificate*, que define «el cuerpo» del certificado y está constituido por otra SEQUENCE (véase definición del certificado X.509 de la RFC 3282 que se presenta unas páginas más atrás) que ocupa en total $02B2_{16}$ octetos, es decir, 690 en decimal, que sumados al número 7 que es el orden que ocupa el último octeto ($B2_{16}$) del campo de longitud, dan 697, que será el número de orden del octeto donde termina este campo «1» (véase figura) y donde comenzará el campo *signatureAlgorithm*, marcado en la figura con un «2». Procediendo de forma similar se localizaría la ubicación del campo *signatureValue* que representa la firma del certificado por la CA.

A partir del octeto número 7 empezará el «cuerpo» del certificado, cuyo primer campo es *versión*, marcado con «1a» en la figura, que es una etiqueta [0] con campo de longitud 03 octetos, que engloba un entero de valor 02 (que indica que se trata de la versión 3 del certificado). Así, leyendo el campo de longitud y el octeto de código, con ayuda de la descripción que antes hacíamos de los distintos campos del certificado, podríamos obtener las marcas que representan los límites para los distintos campos, tal y como se recoge en la figura. Puede observarse que no aparecen los campos optativos *issuerUniqueID* y *subjectUniqueID*, por lo que el último campo de esa SEQUENCE es el de *extensions*, marcado con «1h».

La cosa no acabaría aquí. Para seguir leyendo el certificado habría que ser capaz de interpretar cada uno de los campos internamente y deducir los valores reales de los datos contenidos en el certificado. Aunque tediosa y pesada, esta es una tarea que es posible llevar a cabo disponiendo de la especificación completa del certificado que está recogida en la ISO/IEC 9594-8.

Es fácil ver que por este camino no llegaríamos muy lejos si lo que pretendemos es interpretar un certificado. Resulta evidente que lo que necesitamos es un programa que interprete ese código binario y nos lo muestre de manera inteligible. No obstante, la representación de la Figura 6.21 y los comentarios que le hemos dedicado en los párrafos precedentes nos sirven para reforzar la idea de que, en último extremo, un certificado no es sino una pieza de información constituida por una ristra de octetos. Las distintas entidades comunicantes y agentes telemáticos que manejan los certificados deberán disponer internamente de programas que sirvan para interpretar esa ristra de octetos (en base a la especificación formal del certificado) y poder conocer el valor de los distintos campos, así como ser capaces de verificar la firma de la CA.

Volviendo a la tarea, antes iniciada, relativa a la lectura del certificado representado en la Figura 6.21, se muestra a continuación (y hasta el final del presente epígrafe) el resultado que genera un programa auxiliar que sirve para interpretar certificados. Para mejor entendimiento, con ayuda de un editor de texto, hemos añadido, en cursiva y negrita, algunas anotaciones, con la misma nomenclatura utilizada en el pie de la Figura 6.21, que sirven para localizar los distintos campos del certificado en esa versión interpretada. Hagamos algunos comentarios sobre el formato ahí representado:

- En la columna de la izquierda aparece, en decimal, el número de orden del octeto en el que comienza cada campo. Así, el *Certificate* comienza en el octeto 0, el campo *versión* en el octeto 8, etc.
- La siguiente columna representa el código hexadecimal del tipo de dato con el que se representa cada uno de estos campos. Así, un 30_{16} para las dos primeras SEQUENCE, un $A0_{16}$ para la etiqueta [0], un 02 para el INTEGER, etc. (Puede comprobarse la coincidencia, respecto a esta información, con lo reflejado en la Figura 6.21.)
- En la tercera columna aparece (en decimal) el número neto de octetos mediante los cuales se representa cada uno de los elementos de la especificación. Los dos primeros son los 756 y 690 que antes habíamos deducido del volcado hexadecimal puro.
- A continuación, después de cada tipo, aparece el valor binario (hexadecimal) del campo concreto. Así por ejemplo, el valor del campo *serialNumber* son esos 16 (decimal) octetos, el primero de los cuales es $7F_{16}$ y el último 14_{16} (en la Figura 6.21 encontramos estos 16 octetos a partir del octeto número 15). Otro ejemplo: en la fila marcada por el octeto 56 aparece un código 13_{16} correspondiente a un *PrintableString* cuyo valor «binario» dice que es 'ES'. Con las comillas lo que quiere indicar es que esa es la interpretación del código ASCII que realmente aparece en la representación binaria; en la Figura 6.21 esos dos octetos son 45_{16} y 53_{16} ubicados a partir del octeto número 58. Otro tanto podríamos decir del 'MADRID' que aparece en la misma fila del octeto 69.
- Como se refleja en la fila marcada por el octeto 33, la CA firma el certificado usando los algoritmos DSA y SHA1.
- La identificación de la CA emisora se refleja, a partir del octeto 44, a través de 138 octetos, mediante una secuencia de SETs. De forma similar, a partir de la fila marcada por el octeto 217 y hasta la marcada por el octeto 391, se recoge la identidad del titular del certificado.
- En la fila marcada por el octeto 391 y siguientes se especifica que la clave pública del titular es RSA, y más adelante se refleja el valor binario (*e*, *n*) de dicha clave. El programa intérprete no nos ha desglosado ese valor señalando qué valor corresponde a cada una de las partes. El programa intérprete que resida en la entidad o agente telemático que maneje este certificado sí tendrá que conocer con precisión la sintaxis interna mediante la que está definida la clave pública RSA para poder extraer el valor concreto de cada uno de esos dos parámetros.
- En la fila marcada por el octeto 553 y siguientes se recoge que el campo *extensions* está constituido por una SEQUENCE de tres extensiones (*keyUsage*, *subjectAltName* y *policyMappings*), cuya especificación detallada vimos antes a modo de ejemplo. Pero el intérprete que ha impreso esta información no sabe o no «quiere» desglosarnos sus campos internos.

- Por último, a partir de los octetos 698 y 711 respectivamente, se recogen los campos que en la Figura 6.21 marcamos como «2» y «3»: el algoritmo de firma y el valor binario de ésta.

El certificado interpretado es el siguiente:

```

0 30 756: SEQUENCE { 0 Certificate
4 30 690:   SEQUENCE { 1 tbsCertificate
8 A0 3:     [0] { 1a version
10 02 1:      INTEGER 2
      :      }
13 02 16:    INTEGER 1b serialNumber
      :      7F 00 00 01 00 00 02 7C 00 00 00 01 00 00 00 14
31 30 11:    SEQUENCE { 1c signature
33 06 7:      OBJECT IDENTIFIER dsaWithSha1 (1 2 840 10040 4 3)
42 05 0:      NULL
      :      }
44 30 138:    SEQUENCE { 1d issuer
47 31 11:      SET {
49 30 9:        SEQUENCE {
51 06 3:          OBJECT IDENTIFIER countryName (2 5 4 6)
56 13 2:          PrintableString 'ES'
      :          }
      :      }
60 31 15:      SET {
62 30 13:        SEQUENCE {
64 06 3:          OBJECT IDENTIFIER stateOrProvinceName (2 5 4 8)
69 13 6:          PrintableString 'MADRID'
      :          }
      :      }
77 31 15:      SET {
79 30 13:        SEQUENCE {
81 06 3:          OBJECT IDENTIFIER localityName (2 5 4 7)
86 13 6:          PrintableString 'MADRID'
      :          }
      :      }
94 31 12:      SET {
96 30 10:        SEQUENCE {
98 06 3:          OBJECT IDENTIFIER organizationName (2 5 4 10)
103 13 3:          PrintableString 'UPM'
      :          }
      :      }
108 31 15:      SET {
110 30 13:        SEQUENCE {
112 06 3:          OBJECT IDENTIFIER organizationalUnitName (2 5 4
      :          11)
117 13 6:          PrintableString 'DIATEL'
      :          }
      :      }
125 31 18:      SET {
127 30 16:        SEQUENCE {
129 06 3:          OBJECT IDENTIFIER commonName (2 5 4 3)
134 13 9:          PrintableString 'DIATEL CA'
      :          }
      :      }
145 31 38:      SET {
147 30 36:        SEQUENCE {

```

```

149 06 9:          OBJECT IDENTIFIER emailAddress (1 2 840 113549 1
                      9 1)
160 16 23:         IA5String 'webmaster@diatel.upm.es'
                :
                :
                :
                }
                }
185 30 30:         SEQUENCE {          1e validity
187 17 13:         UTCTime '010825083224Z'
202 17 13:         UTCTime '040826081312Z'
                :
                }
217 30 171:        SEQUENCE {          If subject
220 31 11:        SET {
222 30 9:          SEQUENCE {
224 06 3:          OBJECT IDENTIFIER countryName (2 5 4 6)
229 13 2:          PrintableString 'ES'
                :
                :
                }
233 31 15:        SET {
235 30 13:        SEQUENCE {
237 06 3:          OBJECT IDENTIFIER stateOrProvinceName (2 5 4 8)
242 13 6:          PrintableString 'MADRID'
                :
                :
                }
250 31 15:        SET {
252 30 13:        SEQUENCE {
254 06 3:          OBJECT IDENTIFIER localityName (2 5 4 7)
259 13 6:          PrintableString 'MADRID'
                :
                :
                }
267 31 12:        SET {
269 30 10:        SEQUENCE {
271 06 3:          OBJECT IDENTIFIER organizationName (2 5 4 10)
276 13 3:          PrintableString 'UPM'
                :
                :
                }
281 31 15:        SET {
283 30 13:        SEQUENCE {
285 06 3:          OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
290 13 6:          PrintableString 'DIATEL'
                :
                :
                }
298 31 24:        SET {
300 30 22:        SEQUENCE {
302 06 3:          OBJECT IDENTIFIER commonName (2 5 4 3)
307 13 15:         PrintableString 'Justo Carracedo'
                :
                :
                }
324 31 38:        SET {
326 30 36:        SEQUENCE {
328 06 9:          OBJECT IDENTIFIER emailAddress (1 2 840 113549 1
                      9 1)
339 16 23:         IA5String 'carracedo@diatel.upm.es'
                :
                :
                }
364 31 25:        SET {
366 30 23:        SEQUENCE {

```



```

368 06 10:      OBJECT IDENTIFIER userID (0 9 2342 19200300 100
                  1 1)
380 13  9:      PrintableString 'carracedo'
                  }
                  }
                  }
391 30 159:     SEQUENCE { Ig subjectPublicKeyInfo
394 30 13:       SEQUENCE {
396 06  9:         OBJECT IDENTIFIER rsaEncryption (1 2 840 113549 1 1 1)
407 05  0:         NULL
                  }
409 03 141:     BIT STRING 0 unused bits
                  30 81 89 02 81 81 00 A6 B7 0D 04 AE F7 76 74 57
                  1B EF 73 36 DE 67 64 F6 77 8F 50 39 0F 55 BE 68
                  85 C1 35 5A A7 30 28 C8 0A F1 5F E9 6E 2E F6 EF
                  F3 31 88 B3 8A 1E DE B5 4D 4B F1 39 5B 59 19 2C
                  44 94 28 01 09 4F 85 4A 5F 36 B4 EA 75 BC 3B EE
                  2C 17 F6 25 C5 EC 9D 17 F6 57 CD 8E 7F EA AA 41
                  17 DD EB E9 12 7A C3 7B 47 5A 98 B3 33 F5 5B F3
                  03 2B 00 16 71 FE DF 76 83 FD C4 5E A5 2A 4B 9B
                  [ y otros 12 octetos más....]
                  }
553 A3 142:     [3] { Ih extensions
556 30 139:       SEQUENCE {
559 30 11:         SEQUENCE {
561 06  3:           OBJECT IDENTIFIER keyUsage (2 5 29 15)
566 04  4:           OCTET STRING
                    03 02 04 90
                    }
572 30 66:         SEQUENCE {
574 06  3:           OBJECT IDENTIFIER subjectAltName (2 5 29 17)
579 04 59:           OCTET STRING
                    30 39 81 17 63 61 72 72 61 63 65 64 6F 40 64 69
                    61 74 65 6C 2E 75 70 6D 2E 65 73 86 1E 68 74 74
                    70 3A 2F 2F 63 61 72 72 61 63 65 64 6F 2E 64 69
                    61 74 65 6C 2E 75 70 6D 2E 65 73
                    }
640 30 56:         SEQUENCE {
642 06  3:           OBJECT IDENTIFIER policyMappings (2 5 29 33)
647 01  1:           BOOLEAN TRUE
650 04 46:           OCTET STRING
                    30 2C 30 16 06 09 2A 86 48 86 F7 0D 01 09 10 06
                    09 2A 86 48 86 F7 0D 01 09 10 30 12 06 07 2A 81
                    07 70 02 08 07 06 07 2A 81 07 70 02 08 07
                    }
                    }
                    }
                    }
698 30 11:     SEQUENCE { 2 signatureAlgorithm
700 06  7:       OBJECT IDENTIFIER dsaWithSha1 (1 2 840 10040 4 3)
709 05  0:       NULL
                  }
711 03 47:     BIT STRING 0 unused bits 3 signatureValue
                  30 2C 02 14 33 42 4B 2C C4 82 13 AC D6 D6 F5 5A
                  A2 D6 C8 8F 94 37 63 73 02 14 08 F6 F2 A2 BA D1
                  D7 79 25 91 C3 0F 82 EC B8 19 3C 12 B9 51
                  }

```

6.7. REVOCACIÓN Y SUSPENSIÓN DE CERTIFICADOS

La inclusión del concepto de certificado en las estructuras de gestión y distribución de claves públicas representa **casi** un cierre perfecto en el proceso de securización de las comunicaciones, tanto en lo que se refiere a la confidencialidad e integridad de los datos transferidos como a la autenticación de su origen, además de otros servicios de seguridad que pueden prestarse cuando las exigencias de los usuarios así lo requieran. Pero le falta el «casi». El certificado es una pieza de información segura en sí misma, por lo que deja de tener relevancia el cómo y el desde dónde se distribuya. Las *entidades usuarias del certificado* (*relying parties*), es decir, aquellas que confían en los certificados que emite una CA porque confían en la probidad de sus gestores y tienen seguridad en la validez de la clave pública de esa CA, estas entidades, decimos, podrían olvidarse de más averiguaciones durante todo el periodo de vigencia del certificado una vez obtenido éste. Pero no pueden: hay que contar con los arrepentimientos.

Desafortunadamente, existen circunstancias que provocan la invalidez de un certificado aunque no haya concluido el periodo de validez que aparece explícitamente reflejado en uno de sus campos. Algunas de estas causas son de tipo organizativo, como el hecho de que un determinado usuario ha dejado de pertenecer a alguna organización, pero las más graves son aquellas que se producen cuando el propietario de un certificado teme que la clave privada, pareja de la pública certificada, no está tan segura como debía a causa de un descuido en su custodia o ante el temor a los efectos de un ataque que haya sufrido. En estos casos hay que proceder a generar un nuevo par de claves, declarar no válido el anterior certificado y poner los medios para que las entidades que hipotéticamente puedan usarlo tengan conocimiento de tan infausta circunstancia.

Llegado este caso, lo más contundente y eficaz es *revocar* el certificado, es decir, dejarlo inválido para los restos. Esta situación se hace patente mediante las *Listas de Revocación de Certificados*, CRLs, que más adelante se describen. Para complicar más las cosas, se dan circunstancias en las que esos temores existentes son más bien de carácter dubitativo o circunstancial, de tal manera que sería interesante *suspender* la validez del certificado durante un tiempo, al cabo del cual decidir si revocarlo definitivamente o restituirle su validez. En dominios de seguridad cerrados y poco numerosos no resulta difícil manejar una y otra situación. Por contra, cuando nos encontramos en entornos muy abiertos, donde concurren organizaciones distintas y donde es muy elevado el número de entidades a las que se les ha expedido un certificado, si ya es difícil manejar el concepto de revocación, más difícil aún es tener debidamente en cuenta la situación de suspensión temporal.

La entidad con capacidad para revocar o suspender un certificado es la CA que lo generó. El certificado revocado o suspendido puede pertenecer a una entidad usuaria final o a otra CA cuya clave haya sido certificada. En este último caso, los efectos de la revocación se expanden hacia todos los certificados que hayan sido emitidos por ella. Las causas por las que puede ser revocado un certificado están recogidas en una extensión que más adelante comentaremos (Extensión *reasonCode*) y son las siguientes:

- **Clave de entidad propietaria comprometida⁹** (*Key Compromise*): Es lo que ocurre cuando la entidad final sabe o sospecha que la clave privada, pareja de la clave pública certificada, es conocida por otros usuarios. El uso de tarjetas inteligentes a prueba de ataques que realicen internamente las operaciones de

cifrado con la clave privada, de forma que ésta nunca abandone el chip que la contiene, representa una ayuda considerable a la hora de custodiar adecuadamente la clave mantenida en secreto. La pérdida de la tarjeta (a pesar de estar protegida ante intrusos mediante mecanismos de control de acceso) puede ser, no obstante, causa para procurar que sea generado un nuevo par de claves y revocado el certificado que validaba a la que ha quedado comprometida. En todos los casos, las decisiones de este tipo deben venir marcadas en la política de seguridad definida en el dominio.

- **Clave de CA comprometida** (*CA Compromise*): En ese caso la revocación afecta al certificado de una CA al suponer en riesgo la seguridad de su clave privada.
- **Cambio de afiliación** (*Affiliation Change*): En función de cómo esté definida la política de certificación, cuando cambian algunos datos relevantes del usuario o *entidad suscrita* a nombre de la cual está expedido el certificado, el certificado en cuestión debe ser revocado. Se supone que esa misma u otra CA le expedirá un nuevo certificado conforme a sus nuevos parámetros y claves (si es que se genera un nuevo par). Es fácil deducir que la importancia de esta causa de revocación es mucho menor que la derivada del compromiso de las claves.
- **Certificado reemplazado** (*Superseded*): La CA puede, por motivos muy justificados, tomar la decisión de reemplazar por otro nuevo el certificado que ha expedido a una determinada entidad final antes de que finalice el periodo de vigencia con el que fue emitido.
- **Cese de operación** (*Cessation of Operation*): Si por cualquier circunstancia, la entidad propietaria del certificado ha cesado de operar en las condiciones bajo las cuales fue emitido el certificado, éste pierde sentido y resulta innecesario que continúe vigente.
- **Borrado de la CRL** (*Remove from CRL*): Es un indicador de que el certificado, que antes se encontraba incluido en una CRL, ha sido borrado de ella. Eso no significa que haya recuperado el carácter de activo, porque puede ser que haya sido incluido en otro tipo de CRL distinta (por ejemplo una *Delta --CRL*, que más adelante se comenta).
- **Suspensión de Certificado** (*Certificate Suspension o Certificate Hold*): El certificado puede haber quedado suspendido durante un tiempo hasta que se aclaren ciertas sospechas. En este caso no se han definido parámetros que sirvan para discriminar cuál fue la causa de la suspensión. Si en una determinada lista de certificados inválidos, el certificado estaba calificado como suspendido y en la siguiente aparece como **borrado de la CRL**, puede significar, según esté definida la política de seguridad en el dominio, que ese certificado ha recuperado su situación de activo. Como veremos más adelante, la autoridad emisora de la CRL puede proporcionar información complementaria que ayude al usuario final a tomar las decisiones adecuadas.

En todo este asunto hay dos aspectos a considerar: cómo dejar reflejada la revocación y cómo hacerla llegar hasta las entidades que hacen uso de esos certificados. El primero de ellos se aborda mediante las listas de certificados revocados, CRLs, y el segundo es un problema de gestión y distribución de esa información. En líneas generales, la dificultad que todo esto conlleva es que cuando vayamos a usar un certificado para, por ejemplo, validar una firma digital, tenemos que tener garantías de que la clave estaba vigente en el momento en el que la firma fue realizada, para lo cual tenemos que tener cierta certeza de que el certificado de esa clave no ha sido revo-

cado. Si encontramos una CRL donde esté incluido, queda resuelta nuestra duda en sentido adverso y rechazaríamos por inválido tal documento. Pero si no encontramos una CRL donde esté incluido, sólo sabemos que no aparece la revocación en el sitio que nosotros creemos que debe aparecer, pero esto no conlleva la certeza plena de que esa revocación no se haya producido, o que se haya producido instantes después de que la lista en cuestión haya sido generada. A estas y otras cuestiones trataremos de dar respuesta en los apartados que siguen.

FORMATO DE LAS LISTAS DE CERTIFICADOS REVOCADOS (CRLs)

Cuando, en 1998, se aprobó la primera versión del certificado X.509, se incluyó también una primera versión de la definición de las CRLs. Posteriormente, con la definición de la tercera versión del certificado y la inclusión del campo de extensiones, se realiza también una nueva definición de esta pieza de información, apareciendo así la versión 2 de la CRL que también hace uso del *campo de extensiones* para matizar las circunstancias en que se ha producido la revocación. En la norma ISO/IEC 9594-8 se recoge la versión 2 de la CRL como sigue:

```

CertificateList ::= SIGNED SEQUENCE {
    version                               Version OPTIONAL,
                                           -- Si está presente, la versión debe ser v2
    signature                             AlgorithmIdentifier,
    issuer                                 Name,
    thisUpdate                             UTCTime,
    nextUpdate                             UTCTime OPTIONAL,
    revokedCertificates                   SEQUENCE OF SEQUENCE {
        userCertificate                   CertificateSerialNumber,
        revocationDate                     UTCTime,
        crlEntryExtensions                 Extensions OPTIONAL}OPTIONAL,
    crlExtensions                        [0] Extensions OPTIONAL}
}

```

Es decir, se trata de una secuencia de siete campos firmada por la autoridad que la expide (valores en claro acompañados de la firma del documento completo). Los cinco primeros nos resultan, después de haber estudiado la especificación formal del certificado, fáciles de entender. En primer lugar aparece el campo *version*, que no figuraba en la primera versión de la CRL, por lo que es optativo en esta segunda. En los siguientes campos se recogen el algoritmo de firma utilizado, el nombre distintivo de la autoridad que ha generado la CRL y los campos *thisUpdate* y *nextUpdate*, que sirven para indicar la fecha y hora de emisión de la presente CRL y la fecha prevista para la siguiente actualización. Los tipos en los que se encuadran estos campos son los mismos que se usan, como ya vimos, en la especificación del certificado.

El campo *issuer* recoge el nombre de la autoridad que ha emitido la CRL. En un principio, las CRLs se pensaron para que cada CA generase sus propias listas con los certificados por ella generados que con posterioridad había sido necesario revocar. Posteriormente, con la aparición de la segunda versión, este concepto se revisó para hacer más flexible la cuestión de los puntos de distribución, permitiéndose que una determinada autoridad pudiese generar listas con certificados revocados por otras CAs.

Luego viene el campo más significativo, *revokedCertificates*, consistente en una SEQUENCE que define la lista de certificados propiamente dicha: cada uno de sus elementos contiene mediante una nueva SEQUENCE los datos de uno de los certifi-

cados revocados. Para cada uno de estos certificados revocados es necesario indicar el algoritmo con que fue firmado, el nombre de la CA que lo generó (que, según lo expuesto en el párrafo anterior, no tiene por qué coincidir con el de la autoridad que emite la CRL), el número de serie con que fue generado y la fecha y hora de su revocación. Opcionalmente, pueden añadirse extensiones.

Por último, hay un campo optativo, *crlExtensions*, que sirve para incluir en la CRL cuantas extensiones se considere conveniente.

PEM, el estándar de Internet para correo seguro, adoptó con ligeros cambios la primera versión de las CRLs, en base a la cual se ha definido la segunda versión de CRL para la comunidad Internet, que coincide, con ligeras variaciones, con la definida por ITU-T e ISO. En el documento antes citado *Certificate and CRL Profile* [HPFS02] del PKIX WG se recoge una especificación de la CRL que se diferencia de la más arriba comentada sólo en la forma de presentarla (no usando el tipo SIGNED, de forma similar a la definición que hacen del Certificado X.509) y sustituyendo el tipo UTCTime por el tipo Time (que permite elegir entre UTCTime y GeneralizedTime) que usan, según vimos en el epígrafe 6. 6, para el campo *validity* del certificado.

EXTENSIONES ESPECÍFICAS RELACIONADAS CON LAS CRLS

Las **extensiones** pueden añadirse, siempre de forma optativa, tanto a la lista en su conjunto, campo *crlExtensions*, como a la secuencia descriptiva de cada uno de los certificados revocados, campos *crlEntryExtensions*. En ambos casos, desde un punto de vista meramente formal, la definición de tipo, *Extensions*, es exactamente la misma, y coincide igualmente con la formulada en la especificación del certificado X.509. Ello significa que cualquiera de las extensiones definidas para el certificado podrían ser utilizadas aquí si los responsables de emitir las CRLs lo consideran conveniente. De hecho, las CRLs utilizan frecuentemente la extensión que permite asignar nombres alternativos a las autoridades emisoras y todas las extensiones con información relativa al uso de las claves y a las políticas.

No obstante, se ha definido un grupo de extensiones específicamente pensadas para ayudar a las entidades finales a hacer el más provechoso uso de la información facilitada a través de estas listas. En los párrafos que siguen se irán comentando todas estas extensiones, continuando con el mismo criterio utilizado en la descripción de las extensiones del certificado: mostrar solamente, a modo de ejemplo, la descripción formal de algunas de ellas.

Cuatro de estas extensiones específicas son las que afectan a cuestiones generales de las listas. Ninguna de ellas es crítica. Son las siguientes:

1. *Número de CRL (crlNumber)*

Sirve para crear un número secuencial creciente que sirva para identificar la lista entre las emitidas por una misma autoridad, conforme a la política de seguridad definida, dirigida a un determinado grupo de entidades usuarias de certificados. Este número puede ayudarles a conocer cuál fue la anterior CRL de la serie y establecer, en su caso, las comparaciones necesarias.

2. *Código de motivo (reasonCode)*

Recoge, según ya se comentó, la causa por la que el certificado ha sido invalidado, siendo una de ellas haberlo situado en estado de «suspendido» (*hold*). Su especificación formal es la siguiente:

```

reasonCode EXTENSION ::= {
    SYNTAX
    IDENTIFIED BY
    CRLReason
    id-ce-reasonCode }

```

```

CRLReason ::= ENUMERATED {
    unspecified (0),
    keyCompromise (1),
    cACompromise (2),
    affiliationChanged (3),
    superseded (4),
    cessationOfOperation (5),
    certificateHold (6),
    removeFromCRL (8) }

```

Como puede comprobarse, «el aspecto» formal de su definición es totalmente coincidente con el de las extensiones del certificado cuya especificación hemos mostrado en el anterior epígrafe: los campos que indican la sintaxis concreta de la extensión y su identificador.

El significado de cada uno de los motivos ya fue discutido más arriba. Puede observarse que se ha dejado hueco para un motivo no especificado en la norma que puede ser usado dentro de un determinado dominio de seguridad para añadir una nueva causa, que sería reconocida por todos los usuarios de certificados que estuviesen al tanto de la política allí definida.

Caso de que la invalidación del certificado sea temporal, a la espera de una decisión posterior (*certificateHold*), la información proporcionada por esta extensión es excesivamente genérica, no aportando pistas sobre las causas de esa suspensión y sobre las circunstancias en las que la suspensión será levantada. Para poder solucionar esta imprecisión, la extensión que se describe a continuación permite añadir información que ayude a los usuarios en la toma de decisiones una vez se haya presentado este caso.

3. Código de instrucciones (*instructionCode*)

Aunque el nombre que se le ha asignado no ayuda a sugerir su funcionalidad, esta extensión aporta el *object identifier* de un documento donde se dan instrucciones a los usuarios para cómo actuar en caso de que un certificado haya sido suspendido. Esas instrucciones habrán sido formuladas por la autoridad de seguridad responsable de la política de seguridad en el entorno en el que estaba siendo utilizado el certificado suspendido.

4. Fecha de invalidez (*invalidityDate*)

Esta extensión refleja la fecha y hora en que se detectó que la clave privada pareja de la clave pública certificada había quedado comprometida. Obviamente, debe ser anterior a la fecha de emisión de la CRL. Lo ideal sería que entre una y otra hora mediase el menor lapso de tiempo posible para evitar sorpresas desagradables. Pero esto no siempre es posible. De ello trata el siguiente capítulo.

Como decíamos al principio de este epígrafe, el hecho de que los certificados puedan ser revocados dentro del periodo de vigencia para el que fueron emitidos representa un grave problema que presenta serias dificultades para ser gestionado adecuadamente. La existencia de las CRLs, su estructura y las extensiones que es posible añadirle ayudan a la resolución de ese problema. En el próximo capítulo se volverá a insistir en los

procedimientos que pueden servir de ayuda para conocer, en el contexto de las infraestructuras de certificación, el estado de validez de los certificados.

NOTAS AL CAPÍTULO 6

¹ En el contexto de la seguridad que se genera en torno a la certificación digital, una entidad comunicante es una *relying party* o una *relying entity* de una determinada CA cuando confía en los certificados emitidos por esa CA, es decir, es una *entidad que se fia* de esa CA. A la hora de encontrar un nombre en español que equivalga al término inglés *relying party*, se podría elegir el de *parte ligada* o *entidad que se fia*. (La acepción que da el Webster's Dictionary al término *rely* es la de «depender confiadamente» de algo o alguien, y la considera derivada del latín *religere* (atar, ligar a algo).) No obstante, nosotros usaremos preferentemente el término *entidad usuaria del certificado*, ya que en algunas normas, por ejemplo la RFC 2527 [ChFo99], el término *relying party* se considera sinónimo de *certificate user*, en referencia a las entidades que reconocen los certificados emitidos en un determinado dominio.

² Como ya se ha dicho, ITU-T publica sus normas en varios idiomas de amplia difusión mundial, entre ellos el español. En la versión española de la Recomendación X.500 se traduce el término «distinguished» por su correspondiente literal «distinguido». Sin pretender descalificar la validez de esa traducción, nosotros optamos (desde los primeros borradores de mediados de los ochenta producidos dentro del entonces CCITT) por traducir ese término por el español «distintivo» al entender que esta denominación es la que mejor se ajusta a la noción de *identificador único* del usuario (sea éste un ser humano o un sistema automático), que permite distinguirlo y reconocerlo de manera unívoca y no ambigua dentro del conjunto de nombres de usuarios de Directorio.

³ Cuando en 1991 la IAB Privacy Task Force de Internet hace pública las RFCs 1113, 1114 y 1115 en las que se define PEM, el correo seguro de Internet, se acepta el formato del certificado X.509 v1 y se abre una lista de distribución, la *pem-dev@tis.com*, en donde se debate, entre otras cosas, sobre el uso del certificado y sobre la insuficiencia de la versión 1. Posteriormente, esta lista de distribución sirvió para opinar sobre las mejoras (insuficientes) de la versión 2 y se recogieron propuestas para lo que, andando el tiempo, sería la versión 3.

⁴ En el otoño de 1994, el grupo de investigación que coordina el autor de este texto presentó una propuesta de organización de múltiples CAs a un congreso organizado por TERENA (agrupación de las redes académicas y de investigación europeas). En él se proponía el uso de una nueva versión con extensiones para la gestión de políticas de certificación (basándose en un mensaje difundido por Warwick Ford en la lista *pem-dev* y en una contribución de Canadá para la decimoprimer reunión del Grupo de Trabajo 4, del subcomité 21 del ISO/IEC JTC1, celebrada en Southampton). No sólo fue rechazado el artículo, sino que se recibió una verdadera filípica amonestatoria de parte del comité de programa. No obstante, a finales de 1996 fue aprobada la nueva versión incluyendo el campo *extensions*. Como dijo el filósofo, cuando a la realidad se la expulsa por la puerta, se cuela por la ventana.

⁵ Dentro del CCITT, la primera versión de ASN.1 apareció en las recomendaciones X.208 y X.209. Posteriormente, ITU-T las ha reordenado y se recogen bajo las referencias ITU-T X.680 a X.683 y X.690 a X.691, separando las que se refieren a la especificación abstracta propiamente dicha y las que definen las reglas de codificación. El mismo conjunto de normas, dentro de ISO se denomina ISO 8824-1 a 8824-4 y 8825-1 a 8825-2.

⁶ En el ámbito de las RFCs de Internet existe otra sintaxis, denominada XDR, que también permite la representación abstracta de estructuras de datos. La diferencia estriba en que en XDR las reglas de codificación son fijas, es decir, a cada tipo de datos definido le corresponde una representación concreta. La sintaxis definida en XDR es semejante a C y en principio fue definida por SUN para que el intercambio de parámetros en las RPCs (llamadas a procesos remotos) fuese independiente de la máquina. No obstante, puede usarse XDR para definir las PDUs de cualquier protocolo y de hecho se usa en la definición de las PDUs del protocolo de seguridad SSL.

⁷ En gran parte, la información aquí seleccionada procede de unas transparencias manuscritas, realizadas por el autor en 1989 para la impartición de clases, que fueron corregidas y mejoradas en varias ocasiones. La última de ellas, realizada por Manuel Álvarez Sol apoyándose en las últimas versiones de las normas, actualizando definiciones e incluyendo nuevos ejemplos, es la que ha servido de referencia directa.

⁸ ASN.1 es una notación suficientemente genérica y abstracta como para ser de utilidad en la descripción de estructuras de datos de lo más diverso. No obstante, para los fines que aquí perseguimos, nos interesa su faceta como mecanismo que sirve para definir PDUs que viajan por la red en forma de una ristra de octetos que tienen que ser interpretados correctamente en su recepción. Es por ello que, aunque en la norma de ASN.1 no se nombra ni una sola vez el término PDU, toda la descripción y los ejemplos que aquí se incluyen giran en torno a esta, para nosotros, crucial utilización.

⁹ Se ha utilizado aquí «clave comprometida» como traducción del inglés «key compromise» (literalmente «compromiso de clave»), ya que aunque en el lenguaje actual en español la utilización que se hace del adjetivo «comprometido» está casi siempre en relación con la acepción de «estar en compromiso» con algo (con el significado de tener contraída alguna obligación), también significa «que está en riesgo, apuro o situación dificultosa». De hecho, este último entrecomillado es el que recoge el Diccionario de la Real Academia al definir el adjetivo, como segunda opción, después de indicar su carácter de participio del verbo «comprometer».

CAPÍTULO

7

INFRAESTRUCTURAS DE SEGURIDAD (PKIs Y TTPs)

En el capítulo anterior se ha puesto de manifiesto la importancia que tiene el uso de algoritmos y técnicas de criptografía asimétrica para la provisión de los servicios de seguridad. También hemos definido el concepto de certificado de la clave pública y estudiado con cierto detalle la estructura interna del certificado X.509 y de las listas de revocación de certificados (CRLs). Hemos podido comprobar que el certificado es una **pieza clave** para la obtención de la seguridad en redes telemáticas. Si tan importante es el certificado, no menos han de serlo las garantías y medidas de seguridad necesarias para la implantación de la tercera parte de confianza (TTP) que lo genera: la Autoridad de Certificación (CA).

Cuando el entorno telemático en el que se usan los certificados crece en extensión geográfica y complejidad organizativa resulta necesario (como también hemos adelantado en el capítulo anterior) el concurso de diversas CAs para garantizar la provisión de estas credenciales. Además de CAs, es necesaria la existencia de otras TTPs, tales que autoridades de registro, agentes autorizados para emitir listas de certificados revocados, repositorios para el almacenamiento y recuperación de certificados, etc. Una red de entidades y agentes de estas características constituyen la infraestructura de certificación de ese entorno.

Además de éstas, existen otras TTPs (que también deben ser consideradas como parte integrante de una infraestructura de seguridad) que basan su funcionamiento en criptosistemas de clave pública y que son de uso imprescindible para la provisión del servicio de no-repudio y el reforzamiento del servicio de autenticación.

En este capítulo trataremos de fijar el concepto de infraestructura de seguridad y de definir lo que se entiende por infraestructura de clave pública, PKI. En primer lugar se estudian las distintas configuraciones que se presentan para garantizar la certificación, y posteriormente se aborda la definición y servicios de otras TTPs.

7.1. QUÉ SE ENTIENDE POR INFRAESTRUCTURA DE SEGURIDAD

Los servicios de seguridad en las redes telemáticas representan un valor añadido sobre los servicios convencionales que son provistos en los entornos donde estas redes

telemáticas están operativas. Por tanto, para implantar protocolos de seguridad será necesario que previamente exista una *infraestructura* de comunicaciones. Con ello queremos significar que existan unas redes sobre las que esté implantado, al menos, un bloque de transporte (por ejemplo, basado en TCP) en el que las entidades comunicantes puedan acceder y ser accedidas a través de puntos de acceso al servicio representados por sus respectivas direcciones. Lo denominamos *infraestructura* porque está presente en todo el entorno de comunicación y pertenece a todas y cada una de las entidades comunicantes, pero no es pertenencia exclusiva de ninguna de ellas en particular.

Si pensamos ahora en un entorno de comunicación en el que estén implantados servicios de seguridad, lo más normal es que existan determinadas aplicaciones telemáticas genéricas (correo electrónico, EDI, páginas web, transferencia de ficheros, etcétera) o aplicaciones de uso muy específico (por ejemplo, una aplicación de voto electrónico), de forma que en todas ellas la eficacia del protocolo y del servicio concreto que prestan está reforzada por la presencia de un protocolo de seguridad.

(Ya hemos venido dejando constancia de que la inclusión de servicios de seguridad en los servicios telemáticos convencionales, más que un refuerzo optativo es casi una **medida de obligado cumplimiento** si lo que se desea es que esos servicios digitales puedan tener las mismas garantías que las ofrecidas por los servicios de comunicación tradicionales que se apoyan en la información representada en papel.)

Para la implantación de esos protocolos de seguridad sobre una aplicación telemática concreta será necesario el concurso de determinados agentes telemáticos (específicos del protocolo de seguridad o compartidos con el protocolo convencional) que, en cualquier caso, son de uso exclusivo para las entidades comunicantes que están participando en la provisión global de ese servicio. Estos agentes *no* puede decirse que pertenezcan a una *infraestructura* genérica.

Lo que ocurre es que además de esos agentes específicos de cada aplicación existirán otros agentes telemáticos, generalmente TTPs, cuya utilización será imprescindible para la adecuada provisión del servicio. Estas TTPs *pertenecen a todas y cada una de las aplicaciones telemáticas presentes en ese entorno, pero no pertenecen de manera exclusiva a ninguna de ellas en particular*; a eso es a lo que denominaremos una *infraestructura de seguridad*.

Intentemos matizar algo más el concepto. Para no pensar en vacío, imaginemos un entorno de comunicación en el que existan varias CAs que generan certificados, una o varias entidades que generan y guardan listas de certificados revocados y uno o más servidores donde estén almacenados los certificados para que puedan ser consultados por las entidades que se comunican entre sí. Este conjunto de agentes telemáticos (excluidas las entidades comunicantes) constituye, de por sí, una *infraestructura de seguridad*. Es fácil comprobar que cumple con la definición antes formulada porque todos estos agentes son de uso común para todas las entidades comunicantes, las cuales, a su vez, intercambiarán entre sí PDUs conforme a lo establecido en cada una de las aplicaciones telemáticas seguras en que estén involucradas.

Además de lo que acabamos de decir, los elementos constitutivos de una *infraestructura* deben tener las siguientes características:

- a) Los agentes de la *infraestructura* deben constituir un sistema cuyo ámbito de acción se extiende a todo el entorno de seguridad, de forma que dichos agentes puedan estar omnipresentes en todos los intercambios de información que en él puedan llevarse a cabo. Para ello, todos sus agentes han de disponer de un punto de acceso (dirección) perfectamente conocido en todo el entorno, a

- través del cual pueda establecerse comunicación con ellos (que puede consistir en una conexión virtual o simplemente en el envío *connectionless* de una solicitud de servicio con respuesta diferida).
- b) Los agentes de este sistema (la infraestructura) son accesibles por todas las entidades comunicantes de manera separada, de forma que cualquier proceso de aplicación residente en una entidad o cualquier objeto de la aplicación pueda solicitar el concurso de los agentes.
 - c) El comportamiento de los agentes de la infraestructura es genérico e independiente de la aplicación telemática concreta que le solicite sus servicios, de aquí su independencia e «imparcialidad».
 - d) Para que la información que proporcionen tenga efecto, estos agentes deben contar con algún tipo de confianza generalizada por parte de las entidades comunicantes que los utilizan.

Cabría calificar a la infraestructura y a sus agentes en virtud del tipo de criptografía que utilizan. Aunque en algún caso concreto hagan uso de algún mecanismo basado en criptografía simétrica, la robustez de los servicios y de la información que facilitan está basada, casi en su totalidad, en criptografía de clave pública, por lo que, aunque teóricamente fuese posible, carece de interés tecnológico hablar de infraestructuras de clave secreta. O lo que es lo mismo, si decimos que una **infraestructura de seguridad de clave pública** es aquella *infraestructura de seguridad cuyos servicios están implementados y son proporcionados utilizando mecanismos basados en criptografía de clave pública*, no estaríamos añadiendo ningún distinguo significativo sobre el concepto antes analizado de *infraestructura de seguridad*, ya que, sin necesidad de explicarlo, podemos dar por supuesto que los mecanismos utilizados se apoyan en criptografía asimétrica y son de utilidad para aplicaciones que usan servicios de seguridad basados en criptografía asimétrica.

Sí podríamos particularizar el concepto si nos refiriésemos a una **infraestructura de certificación** como aquella *infraestructura de seguridad en la que los agentes y servidores que la constituyen están orientados a la generación, gestión y uso de los certificados de la clave pública*. Con esta acepción hemos hecho uso de este término antes de ahora.

Y entonces, ¿qué es una **PKI**? Sus siglas se corresponden con *Public Key Infrastructure* y en la literatura en la que se cita, con gran frecuencia suele dársele el mismo significado de lo que antes hemos definido como *infraestructura de certificación*. Este es el caso del IETF en las distintas normas generadas dentro del PKIX Working Group.

Lo que ocurre es que con esa limitación se queda algo corta y no serviría para, por ejemplo, la obtención de marcas de tiempo, tan importantes para garantizar la validez de una firma digital. Por ello, algunos autores extienden también el ámbito de las PKIs para que incluyan las TTPs que proporcionan estos y otros servicios.

Además de las TTPs y servidores necesarios para la certificación y de las TTPs que sirven para proporcionar marcas de tiempo, existen, o pueden existir, otras TTPs que cumplan con las características que hemos considerado que debe reunir una *infraestructura de seguridad*.

Entre éstas se encuentran las AAs (*Attribute Authorities*) que emiten *Attribute Certificates* (ACs). En el documento del PKIX WG titulado *Internet X.509 Public Key Infrastructure: Roadmap* [ADT02] se define también otra infraestructura, la *Privilege Management Infrastructure* (PMI), como una colección de agentes que manejan y gestionan estos *certificados de atributos*. Es decir, para los autores se trata de otra infraestructura que está dentro del contexto de la PKI pero que no es exactamente la

PKI. En otras propuestas aparecen también definidas otras infraestructuras relacionadas con agentes que operan con claves públicas.

Llegados a este punto, podemos optar por una de las siguientes alternativas:

- Hablar de *PKI* en sentido restringido a lo concerniente a la certificación y de *PKI extendida* para significar que incluye otras TTPs y otras infraestructuras como *PMI*.
- Hablar de *PKI* en sentido genérico y equivalente a lo que antes hemos definido como *infraestructura de seguridad*.
- Hablar de *infraestructura de certificación* con el significado antes definido y de *infraestructura de seguridad*, a palo seco, para el concepto genérico, dando por supuesto que es *de clave pública* y teniendo presente que incluye lo que es la *infraestructura de certificación* y la infraestructura *PMI*, además de toda TTP que responda a las características antes enunciadas.

Las dos primeras opciones nos obligarían a matizar, a cada paso, para deshacer la ambigüedad que un lector poco advertido podría encontrar si no se reflejasen las restricciones o generalizaciones pertinentes. Por ello, y a pesar de que hemos incluido en el título genérico de este capítulo el acrónimo PKI, en lo sucesivo, cuando se trate de concretar, vamos a optar por la tercera de las posibilidades, que es suficientemente precisa y de formulación no excesivamente larga. En cambio, utilizaremos el término PKI cuando esa concreción no sea significativa.

COMPONENTES DE UNA INFRAESTRUCTURA DE SEGURIDAD

Consecuentemente con la acepción global que hemos adoptado acerca de lo que es una infraestructura de seguridad, dentro de ella podemos distinguir los siguientes componentes:

- *Autoridades de Certificación*, CAs (*Certification Authorities*), para la generación de certificados.
- *Autoridades de Registro*, RAs (*Registration Authorities*), para facilitar la identificación y el registro previo de las entidades del dominio antes de serles emitido un certificado.
- *Autoridades de Repositorio* para el almacenamiento y recuperación de certificados y CRLs (por ejemplo, el Directorio).
- *Autoridades para la generación y distribución de CRLs*.

Todas ellas involucradas en la *infraestructura de certificación*, a las que podemos añadir otras TTPs tales como:

- *Autoridades de Sellado de Tiempo*, TSAs (*Time-Stamping Authorities*). Son TTPs que proporcionan «pruebas de existencia» para un determinado documento en un determinado instante de tiempo.
- *Autoridades de Atributos*, AAs (*Attribute Authorities*). Son TTPs encargadas de la generación de los *Certificados de Atributos*, ACs (*Attribute Certificates*), que son unas piezas de información que contienen un conjunto de atributos para una entidad comunicante y alguna información adicional, que están firmadas digitalmente por la AA que las generó.
- TTPs para la provisión del servicio de no-repudio.
- TTPs para la provisión del servicio de anonimato.

Por encima de todas estas autoridades y agentes se encuentra la *política de seguridad* que han de seguir todas las comunicaciones seguras que se lleven a cabo dentro del entorno. A veces esa política está formulada y documentada al margen de la infraestructura, pero con frecuencia ello se refleja en agentes telemáticos que informan de las condiciones impuestas por cada política, por lo que podríamos añadir:

- Autoridades de emisión y publicación de políticas de seguridad.

Consecuentemente con ello, en este capítulo, dedicado a las *infraestructuras de seguridad*, abordaremos en primer lugar todo lo concerniente a la *infraestructura de certificación* para pasar luego a describir brevemente la *infraestructura PMI* constituida por las AAs y otros agentes que manejan los *certificados de atributos* (ACs). A continuación se describirá la funcionalidad de otra TTP muy significativa: la TSA, que sirve para el sellado de tiempo. En capítulos sucesivos se seguirá estudiando el comportamiento de otras TTPs y elementos que también forman parte constituyente de una infraestructura de seguridad.

El *servicio de autenticación* suele apoyarse fuertemente en una TTP, la Autoridad de Certificación (CA). Dada la importancia de este servicio y la conexión directa que tiene con el mecanismo de firma digital, no se estudiará en este capítulo, sino que será abordado en un capítulo aparte y específico (el siguiente). El *servicio de integridad* se provee también mediante el mecanismo de firma digital o mediante un cifrado simple con la clave privada, por lo que podemos considerar que la problemática que plantea su provisión queda abordada al mismo tiempo que la relativa al *servicio de autenticación*. En cuanto al *servicio de confidencialidad* ya quedó suficientemente tratado en los Capítulos 4 y 5 en su doble variante de ser provisto mediante solo criptografía simétrica o mediante el uso combinado de criptografía de clave secreta y de clave pública. En este último caso, el que presenta mayor interés, el quid de la cuestión reside en la confianza que adquieran las entidades comunicantes acerca de la validez de las claves públicas, cosa que se resuelve mediante una adecuada *infraestructura de certificación*, tal y como veremos en el presente capítulo.

También se dedica un capítulo aparte (el noveno) a estudiar el *servicio de no repudio* y el *servicio de control de acceso*. Estos servicios están muy orientados a cada aplicación concreta y se apoyan en TTPs especializadas. Para implantar los permisos y autorizaciones necesarias, el *servicio de control de acceso* se apoya en AAs y en otros agentes que gestionan los *certificados de atributos* (ACs). En la provisión del *servicio de no repudio* pueden intervenir diversos componentes de una infraestructura de seguridad (entre ellos los que sirven para garantizar la validez de la firma y los que componen una *infraestructura PMI*), pero existen también TTPs especializadas en este servicio, sobre todo para lo referente a obtener garantías en la entrega de mensajes.

Para completar el arco de los seis servicios de seguridad en redes que describimos en el capítulo primero, al *servicio de anonimato* se le dedicará también un capítulo aparte, el undécimo, ya que aunque para su provisión son también necesarias TTPs convencionales, dada su especificidad requiere de agentes especializados, por lo que conviene dedicarle un tratamiento separado.

De lo comentado en los párrafos precedentes se deduce la constante presencia que las infraestructuras de seguridad tienen para la implantación de protocolos telemáticos seguros. Además de anunciar el tratamiento que de ello haremos en los siguientes capítulos, hemos tratado de reflejar aquí los elementos más significativos que constituyen una infraestructura de seguridad, así como las principales características que definen a un sistema de este tipo. En el siguiente apartado abordaremos resumidamen-

te las características que debe tener una infraestructura de seguridad derivadas de la política vigente en ese entorno telemático.

CARACTERÍSTICAS DE LA INFRAESTRUCTURA DERIVADAS DE LA POLÍTICA DE SEGURIDAD

Como antes decíamos, por encima del comportamiento de las distintas autoridades y TTPs se encuentra la política de seguridad que ha de guiar todas las comunicaciones que se lleven a cabo dentro del entorno, porque la política concreta que se siga determinará, también, las responsabilidades en que incurren las TTPs por el hecho de expedir las credenciales o evidencias que les son propias.

Antes de continuar, conviene que recordemos la diferencia que existe entre dos términos que hasta aquí hemos venido utilizando de manera algo indefinida: se trata de *entorno* y *dominio*. El primero de ellos tiene un significado genérico que abarca cualquier segmento del ciberespacio en el que está establecida una determinada relación entre entidades (por ejemplo, una determinada infraestructura de seguridad), mientras que *dominio de seguridad* se refiere exactamente al ámbito en el que está definida una determinada política de seguridad. Con frecuencia, ambos conceptos coincidirán, pero puede darse el caso de que en un mismo entorno puedan coexistir varios dominios.

Pero hay más: una determinada entidad puede poseer, incluso para un mismo par de claves, certificados expedidos bajo distintas políticas, por lo que puede operar en distintos dominios dependiendo de cuál sea la aplicación telemática en que esté involucrada y del certificado que esté presentando para garantizar la validez de su clave pública.

Teniendo en cuenta los condicionantes que impone la política de seguridad, podríamos añadir otras nuevas características (o requisitos) que, además de las ya expuestas, cabe esperar de una infraestructura de seguridad. Éstas son:

- a) La infraestructura deberá soportar **múltiples y variadas políticas** de seguridad para permitir a las entidades comunicantes que hacen uso de ella la utilización de los recursos requeridos en cada caso (unas políticas serán más exigentes que otras en cuanto al número y garantía de las credenciales que sea necesario exhibir).
- b) La infraestructura deberá ser gestionada de tal forma que pueda interoperar coordinadamente con **otras infraestructuras** (administradas por otras personas u organizaciones) para posibilitar el establecimiento de comunicaciones en ámbitos geográficos y administrativos muy amplios. Por ejemplo, para que un profesor pueda acceder, con las suficientes garantías, a la base de datos donde se guardan y modifican las calificaciones académicas de sus alumnos no se requieren las mismas garantías que para enviar un correo seguro a otro colega de una universidad ubicada en un país distinto del suyo; ni se usará, muy probablemente, una misma infraestructura.
- c) Como consecuencia de la política bajo la que trabajen, todos los agentes telemáticos y TTPs que operen dentro de una infraestructura deberán tener presente (y darlo a conocer fehacientemente a sus hipotéticos usuarios) las **responsabilidades legales y económicas** que se derivan de la utilización de las credenciales o evidencias que generan. Tanto en lo relativo a lo que los usuarios están autorizados a hacer con ellas como en lo que concierne a la veracidad y robustez de los datos que estas TTPs proporcionan.

Debido a esta omnipresencia de la política de seguridad en todas las operaciones que se lleven a cabo bajo el paraguas de las infraestructuras, en todas las descripciones que en lo sucesivo hagamos, tanto del comportamiento intrínseco de los agentes y TTPs como de los protocolos y mecanismos necesarios para la provisión de servicios de seguridad, será necesario hacer referencia a los condicionantes que dicha política impone. No obstante, dada la importancia que tiene, se dedicará un capítulo aparte (el décimo) para abordar este trascendental asunto.

7.2. INFRAESTRUCTURAS DE CERTIFICACIÓN. GENERALIDADES

Conforme habíamos fijado más arriba, una *infraestructura de certificación* es aquella infraestructura de seguridad (o parte de ella) en la que los agentes y servidores que la constituyen están orientados a la generación, gestión y uso de los certificados de la clave pública. En la Figura 7.1 se representa la relación entre los distintos elementos que componen una infraestructura de este tipo conforme se recoge en el documento *Public Key Infrastructure. Certificate and CRL Profile* [HFPS02] del PKIX WG. De una parte se contempla el ámbito de las entidades usuarias de la infraestructura y de otra el conjunto de TTPs que se encargan de la gestión de la PKI.

El **Repositorio** de Certificados y CRLs aparece como una entidad compartida entre ambas partes de la infraestructura, tanto a efectos de gestión como en lo relativo a la realización de operaciones de almacenamiento y consulta. Más adelante, en este mismo capítulo (epígrafes 7.3 y 7.7), se analizan algunos aspectos relacionados con estas tareas de almacenamiento y recuperación de información relacionada con los certificados de la clave pública.

La **Autoridad de Registro**, RA, será la encargada de establecer una ligazón fiable entre las claves públicas y la identidad de las entidades candidatas a obtener certificados. Cuando una entidad o usuario desea recibir un certificado firmado por una determinada CA, tendrá que someterse primero a un proceso de registro. La RA confirmará la identidad de la entidad peticionaria, comprobará que su nombre, organización y otros atributos se ajustan a lo requerido para configurar un certificado, y verificará la existencia y protección de la clave privada pareja de la clave pública que se demanda certificar. Se trata, no obstante, de una entidad opcional, ya que su comportamiento puede ser asumido por la propia CA, pero resulta más adecuado separar estas funciones administrativas de las propiamente relacionadas con la generación de certificados. En realidad, poco más cabe decir acerca de la estructura y funcionamiento de la RA. En el Capítulo 10, cuando se aborde lo relativo a los compromisos contractuales que adquieren tanto la entidad solicitante como la CA proveedora del servicio, se volverá a insistir en los condicionantes que determinan el proceso de registro.

Directamente ligado a la RA se encuentra el **conjunto de CAs** que dan soporte a los usuarios de la infraestructura. Aunque en la figura se han representado dos, el número y organización de las distintas CAs presentes dependerá, en cada caso, de la extensión y complejidad del dominio de seguridad de que se trate.

La funcionalidad y organización de las CAs constituyen el «núcleo duro» de cualquier infraestructura de seguridad. En el capítulo anterior, concretamente dentro del epígrafe 6.3, abordamos ya el asunto de la existencia de varias CAs organizadas entre sí para proporcionar los certificados requeridos en un entorno con cierta complejidad. Concretamente, dentro de los apartados «*Varias CAs organizadas jerárquicamente cooperando entre sí*», «*Varias CAs de igual jerarquía cooperando entre sí*»

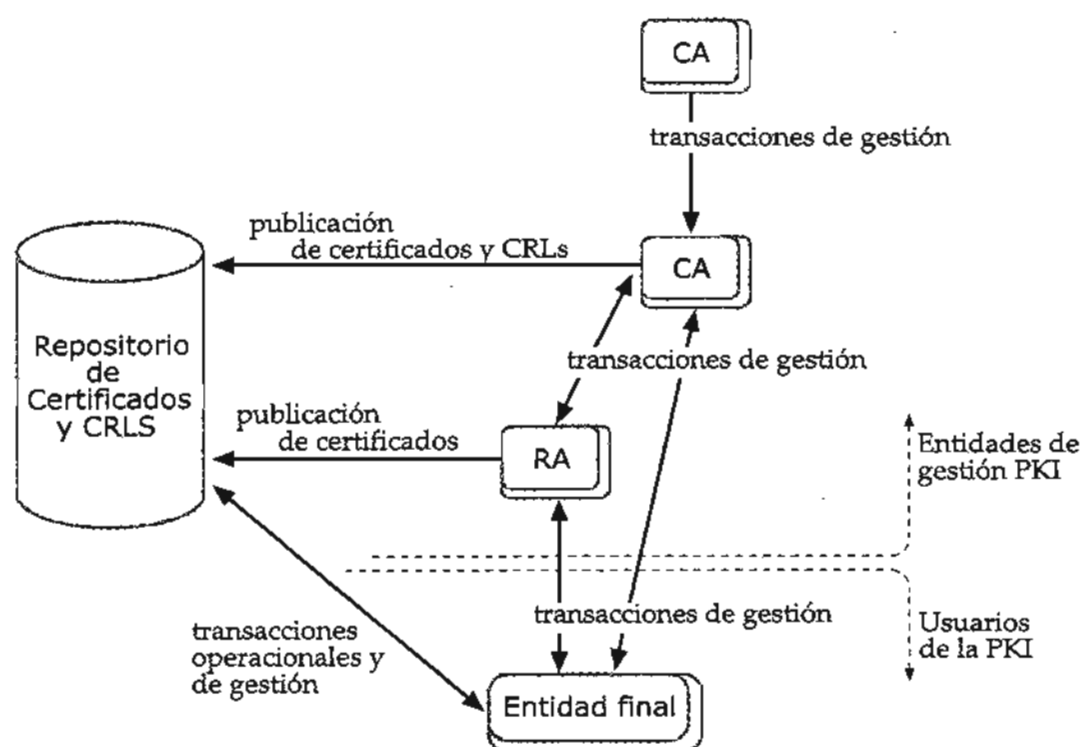


Figura 7.1. Componentes de una infraestructura de certificación (arquitectura PKIX).

y «*Otras formas de organizar la certificación*» se recogen resumidamente las ideas más significativas de estas arquitecturas. Este adelanto se hizo necesario para que se entendieran adecuadamente, entre otros conceptos, el significado del certificado cruzado y las extensiones del certificado X.509 relacionadas con el camino de certificación. En buena ley, esos tres apartados deberían formar parte del presente epígrafe, por lo que se recomienda una relectura previa de ellos para entender adecuadamente lo que se describe a continuación. Por esa misma razón, haremos aquí referencia a las figuras en que se apoyan los razonamientos allí expuestos.

En la descripción que sigue llevaremos un orden inverso al utilizado en esos apartados del capítulo anterior. Comenzaremos por las estructuras más anárquicas y terminaremos por las que obedecen a una fuerte jerarquización, que son, con diferencia, las más frecuentemente utilizadas.

CERTIFICACIÓN SIN CAS

La aplicación de correo electrónico PGP (*Pretty Good Privacy*), que podríamos traducir por algo así como «privacidad bastante buena» [Zime94], propuso en sus inicios (posteriormente han surgido otras versiones que aceptan otras alternativas) un sistema de confianza en las claves públicas que no incluía la existencia de autoridades de certificación. No es nuestro objetivo describir aquí con precisión y fidelidad cómo se realiza la gestión de claves en ese sistema, sino tomarlo como modelo para discutir esta alternativa de infraestructura distribuida, que es casi una «no-infraestructura».

Lo primero que hay que decir de este tipo de modelos descentralizados es que, a pesar de la debilidad que presentan ante determinados riesgos (como más adelante veremos), siguen siendo sumamente interesantes y aportan soluciones muy autónomas y muy poco costosas.

Al no existir CAs, cualquier usuario puede convertirse en autoridad y emitir un certificado de la clave pública de otro usuario. En realidad, el certificado es poca cosa: solamente contiene la identificación de la persona a la que se le expide, la clave pública que se certifica, la fecha de emisión y el nombre y firma digital de quien lo expide. El certificado es emitido, en primera instancia, por el propio titular de la clave certificada (es un autocertificado) y puede ser firmado posteriormente por todos aquellos usuarios que deseen avalar la validez de esa clave.

Cada usuario guardará en su propia máquina una colección de claves públicas (certificados emitidos) de aquellos otros usuarios con los que habitualmente se comunica. Supongamos que la usuaria B tiene una copia fiable de la clave pública del usuario A y se fía de las personas a las que A recomienda o avale: todos aquellos usuarios que envíen a B mensajes firmados y le presenten un certificado firmado por A serán reconocidos por B como válidos. Es decir, la receptora de los mensajes sólo reconoce al firmante que haya sido avalado por alguien de su confianza. A la colección de certificados (claves avaladas) que posee B se le llama «anillo de claves».

La pequeña red de confianza que se ha formado en torno al usuario A, constituida por los usuarios que en él confían y los usuarios avalados por él, puede irse extendiendo de forma ilimitada si se confía en los nuevos usuarios que sean recomendados por otros usuarios ya avalados previamente por alguien de confianza. Se puede ir creando de forma encadenada una malla o entramado de confianza (*web of trust*, en terminología del autor de PGP) a través de la cual buscar *caminos de certificación* hasta encontrar puntos de confianza. Dado que esa malla no tiene una estructura predefinida, sino que se va construyendo de forma algo anárquica, pueden existir múltiples caminos de certificación y no todos ellos con el mismo grado de confianza.

En realidad, se trata de proyectar en el ciberespacio un esquema de amistad y confianza que se produce frecuentemente en el mundo real, aunque en su reproducción virtual las reglas han de ser más estrictas para que gran parte de ellas puedan ser cumplidas de forma automática por las entidades de comunicación.

El proceso de aplicación residente en la máquina de B, ante la recepción de una información cuya fiabilidad se apoya en una determinada clave pública, puede estar diseñado para tomar una decisión acerca de la aceptación (o no) del mensaje basándose en la información de que dispone en su anillo de claves. Ello es debido a que las claves certificadas guardadas en el anillo pueden estar calificadas (en función del grado de confianza) como no-fiables, dudosas, semifiables o fiables. Si la operación se apoya en una clave fiable es aceptada sin más, mientras que un certificado semifiable no es suficiente, por sí solo, para dar por válida la clave certificada y se necesita el concurso de otro semifiable (o varios, según se configure) para aceptar su validez. Si se ha calificado de dudosa, el programa no podrá adoptar una decisión de forma automática y deberá consultar directamente a la usuaria B de lo que ha de hacer.

Independientemente de cómo esté o no previsto hacerlo en PGP, lo interesante de este tipo de configuraciones es la libertad que dejan a la comunidad de usuarios que se constituye para crear sus propias reglas, tanto para la extensión de la malla como para dar por válido un certificado. Los mecanismos de distribución de claves públicas pueden complementarse usando medios no telemáticos o incluso a través de correo electrónico no seguro complementado con una llamada telefónica a través de la cual, y a viva voz, transmitir una *huella* o resumen (deducido mediante un algoritmo predeterminado) de la clave antes enviada.

Estas configuraciones tienen bastante interés cuando un grupo de personas¹ decide constituir un dominio de seguridad en el que no dependen de ninguna CA, bien porque

no se fíen de ninguna en particular, bien porque quieran quedarse al margen de posibles intervenciones de poderes externos al grupo, reforzando así la privacidad. Lo que ocurre es que, cuanto más reducido sea el rigor a la hora de «registrar» a un nuevo miembro, más fácil será que se infiltre alguien con intereses divergentes de los de aquellos que constituyeron el grupo inicialmente.

Este riesgo de infiltración (que alguien consiga, maliciosamente, ser avalado por algún miembro de confianza) será tanto mayor cuanto más extenso y complejo sea el entramado de confianza y mayor liberalidad exista a la hora de posibilitar su expansión. Una vez que un usuario malicioso se introduzca en la malla, podría dar entrada a un número indefinido de nuevos miembros maliciosos y corromper el dominio en cuanto a los fines iniciales para los que se constituyó: la privacidad sería bastante mala.

Una crítica que suele hacerse a este tipo de configuraciones es la falta de rigor en la custodia tanto de las claves privadas como de las claves públicas, ya que, en la práctica, este tipo de aplicaciones utilizan como respaldo los propios almacenes de la máquina donde se trabaja (muy vulnerables) y se usan disquetes como testigos de seguridad para guardar las claves privadas. Esta crítica no es del todo acertada, porque estos hábitos no son intrínsecos de las configuraciones sin CAs, sino que podrían darse en igual medida en cualquiera de las otras configuraciones donde existen CAs instaladas. Nada impide que se diseñe un escenario sin CAs en el que las claves privadas se guarden en tarjetas inteligentes y se cuiden más los almacenes de claves públicas.

Lo que sí es cierto es que en este tipo de soluciones la responsabilidad está muy diluida y que resulta muy difícil generar políticas mediante las cuales alguien quede obligado por lo que firma o por lo que avala. Al no contar con CAs reconocidas públicamente, no pueden contar con certificados reconocidos, por lo que es muy difícil que los compromisos que puedan adquirirse en su entorno lleguen a tener algún tipo de valor jurídico.

Otro problema gordísimo es el relacionado con las revocaciones que puedan ser necesarias si alguno de sus miembros cree que ha sido comprometida la seguridad de su clave. Si ya de por sí, incluso en una estructura muy jerarquizada, el problema de la revocación es un asunto envenenado, en una configuración como la que estamos analizando, en dominios muy extensos, se convierte en algo sin solución (que pueda tener el calificativo de tal). El que el usuario afectado emita un mensaje anunciando el compromiso de su clave para que sea difundido entre las entidades que la tenían previamente almacenada como válida no aporta garantías de nada porque no existe constancia alguna de cuáles son esas entidades. La única salida consiste en conformarse con crear comunidades muy reducidas en las que la revocación llegue a todos sus miembros (por si acaso).

VARIAS CAs DE IGUAL JERARQUÍA

En la Figura 7.2 se representa de nuevo la estructura de comunicación recogida en la Figura 6.11, aunque sin reflejar en ella los distintos certificados cruzados que constituyen la malla de certificación. Además, en este caso, la relación definida por el establecimiento de certificados cruzados se representa simplemente por una línea con doble flecha que une las CAs que tienen establecidos entre sí certificados cruzados. En el fondo, un certificado cruzado no es más que el establecimiento de dos certificados directos, cada uno en una dirección.

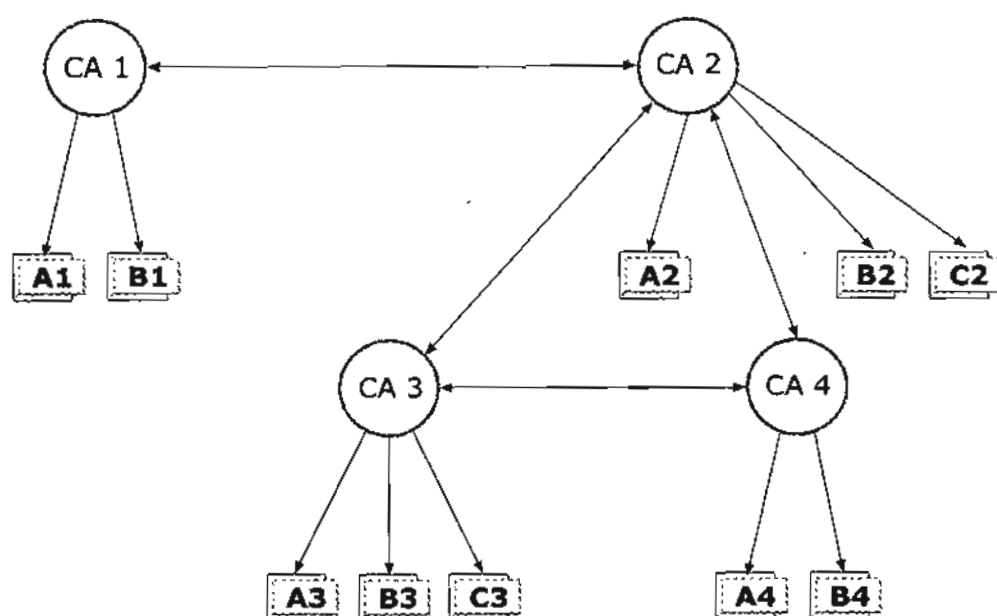


Figura 7.2. Organización de varias CAs de igual jerarquía.

Este modelo es bastante interesante y ofrece una solución sencilla para el caso en que un conjunto de organizaciones autónomas quieran «confederarse» para constituir un dominio de seguridad global en el que cada una de las organizaciones seguiría siendo un subdominio con bastante autonomía.

En este modelo simplificado damos por supuesto que las políticas establecidas en los distintos subdominios son totalmente equivalentes entre sí (o incluso es la misma para todos). De la misma forma, en esta simplificación se supone que cuando una CA reconoce a otra mediante un certificado, reconoce a todas las entidades adscritas a ella, sin excepción de ninguna clase.

A pesar de su interés, es un modelo poco estudiado porque para dominios con un número muy reducido de CAs, los problemas de determinación del camino de certificación son triviales, ya que la estructura de certificación puede ser conocida, sin mayor problema, por todas las entidades que participan en las comunicaciones que se establezcan en el dominio. Por contra, si el número de CAs presentes es muy elevado el problema se convierte en muy difícil debido a que podrían existir varios caminos posibles que deberían ser evaluados para determinar cuál es el óptimo. Además, podrían aparecer bucles de certificación que introducirían cierta inconsistencia. Por eso no es un modelo frecuentemente seguido cuando existe un número grande de CAs. No obstante, dentro de los modelos flexibles que presentamos en el epígrafe 7.6, esta configuración será una de las posibles que puedan establecerse.

La cosa se complica algo más si las políticas vigentes en cada una de ellas no son totalmente compatibles entre sí y si en el reconocimiento mutuo que se establece entre las CAs se quieren hacer restricciones en relación con algunos aspectos de esas políticas, o con respecto a los nombres o direcciones de algunas de las entidades comunicantes que «cuelgan» de las CAs. Este tipo de restricciones es posible imponerlas haciendo uso del campo de extensiones del certificado X.509. Más adelante abordaremos este asunto.

MÚLTIPLES CAs ORGANIZADAS JERÁRQUICAMENTE

Excluyendo las organizaciones deliberadamente paritarias, como es el caso de los entramados que se establecen en una relación de confianza como la que hemos descrito en el caso de PGP, en casi todos los restantes casos aparece, de una u otra forma, la conveniencia de una organización jerárquica de la estructura de certificación. Salvo en algunos casos de relaciones personales, para garantizar con cierto rigor la autenticidad de las entidades comunicantes y la firmeza de los compromisos establecidos en la comunicación (no repudio), la solución jerárquica es la más sencilla de manejar y la que ofrece mejores resultados. Esto es tanto más cierto cuanto más extenso y plural sea el dominio de seguridad en el que se asienta.

En la Figura 7.3 se representa de forma esquemática una estructura de certificación muy similar a la que ya se mostró en la Figura 6.10, con la particularidad de que las dobles flechas indican una certificación de doble dirección: la CA de mayor jerarquía ha emitido un certificado *directo* a favor de la de menor jerarquía y ésta emite un certificado *inverso* en reconocimiento de su dependencia y de la confianza que deposita en esa CA de rango superior.

Es conveniente percatarse de que en este tipo de estructuras, el **control** de la certificación se realiza *de arriba abajo*, de tal manera que partiendo de la CA de mayor jerarquía, la que ocupa la raíz del árbol², cada nuevo escalón jerárquico aparece porque la CA que ocupa una determinada posición decide delegar (si la política de seguridad ahí vigente se lo permite) en otras CAs de rango inferior para que puedan, a su vez, emitir certificados a favor de otras CAs o de entidades finales de comunicación. Por contra, cuando dos entidades comunicantes desean establecer una relación

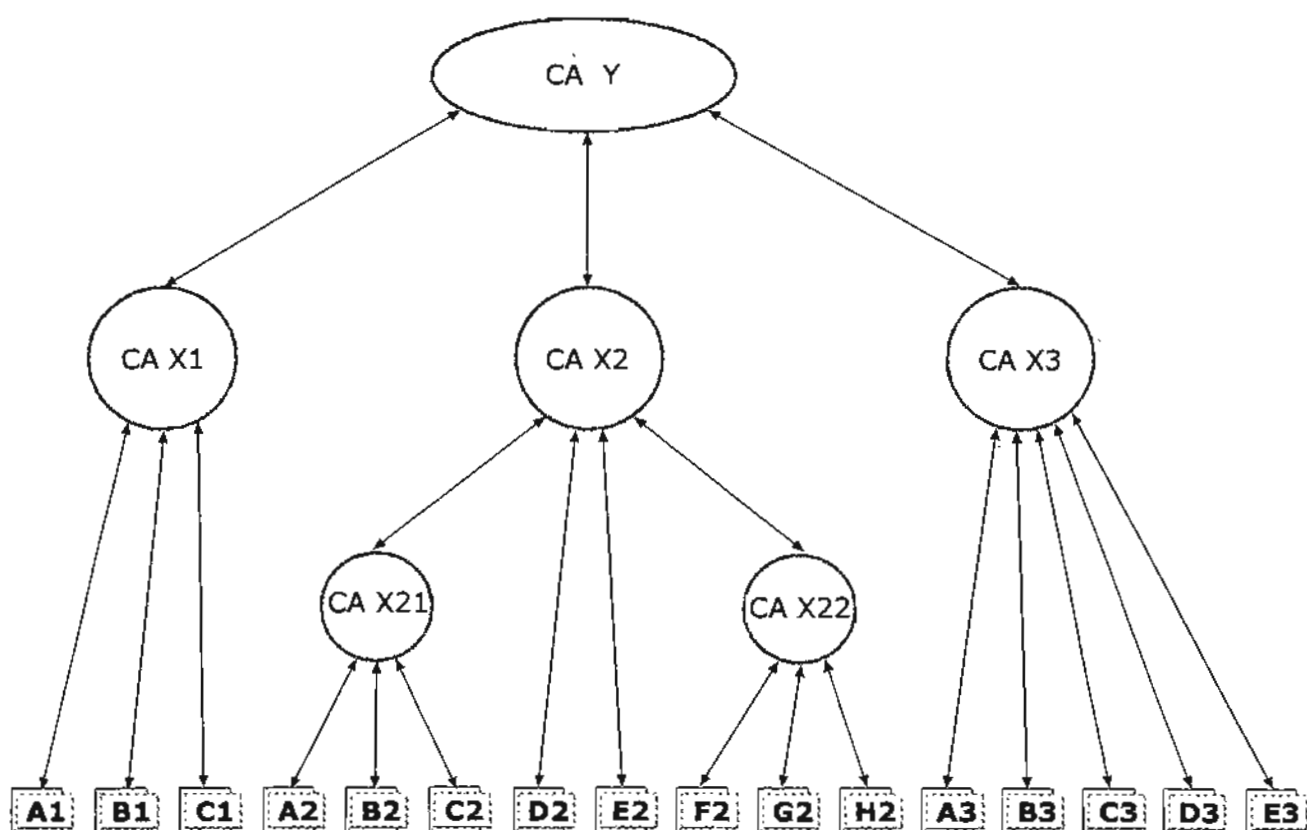


Figura 7.3. Organización jerárquica de CAs con certificados directos e inversos.

de confianza basada en los conjuntos de certificados que cada una de ellas exhibe, esa **relación de confianza** se establece, por lo general, *de abajo arriba*. Es decir, A2 en quien confía plenamente es en sí misma (si esta frase no es cierta, entonces, apaga y vámonos) y lo habitual es que confíe bastante en la CA X21, de la que depende directamente. La confianza que A2 tiene en la CA X2 viene dada por la confianza que en ella tiene depositada X21. Y así sucesivamente, mientras se escalan los distintos niveles jerárquicos de la estructura.

Esta estructura de certificación (con certificados directos e inversos) tiene la gran ventaja de que en ella es fácil encontrar y validar el *camino de certificación* cuando dos entidades desean establecer una relación de confianza. En efecto, a partir de los distintos certificados que cada una de las entidades posee, mediante un proceso iterativo de análisis de los certificados inversos que presentan las distintas CAs que van apareciendo en el escalafón de certificación, se llegará hasta una CA que sea de confianza para ambas partes. Si esa averiguación la está realizando un programa residente, por ejemplo, en la entidad A1 cuando quiere establecer un camino de certificación con A2, ese análisis iterativo acaba cuando se alcanza la CA Y que también es de confianza para A1.

Tiene el inconveniente, debido a esa misma duplicidad de certificados, de que la carga de certificados que tiene que soportar es muy pesada, sobre todo para estructuras que soporten un número muy elevado de CAs y de entidades de comunicación. Además, los siempre tediosos procesos de revocación se complicarían aún más si existen certificados inversos.

Una simplificación de este modelo consiste en suprimir los certificados inversos, Figura 7.4, lo que acentúa aún más el carácter jerárquico de esta estructura. Aunque formalmente es parecidísimo al modelo anterior, presenta mayores dificultades a la hora de calcular el *camino de certificación*.

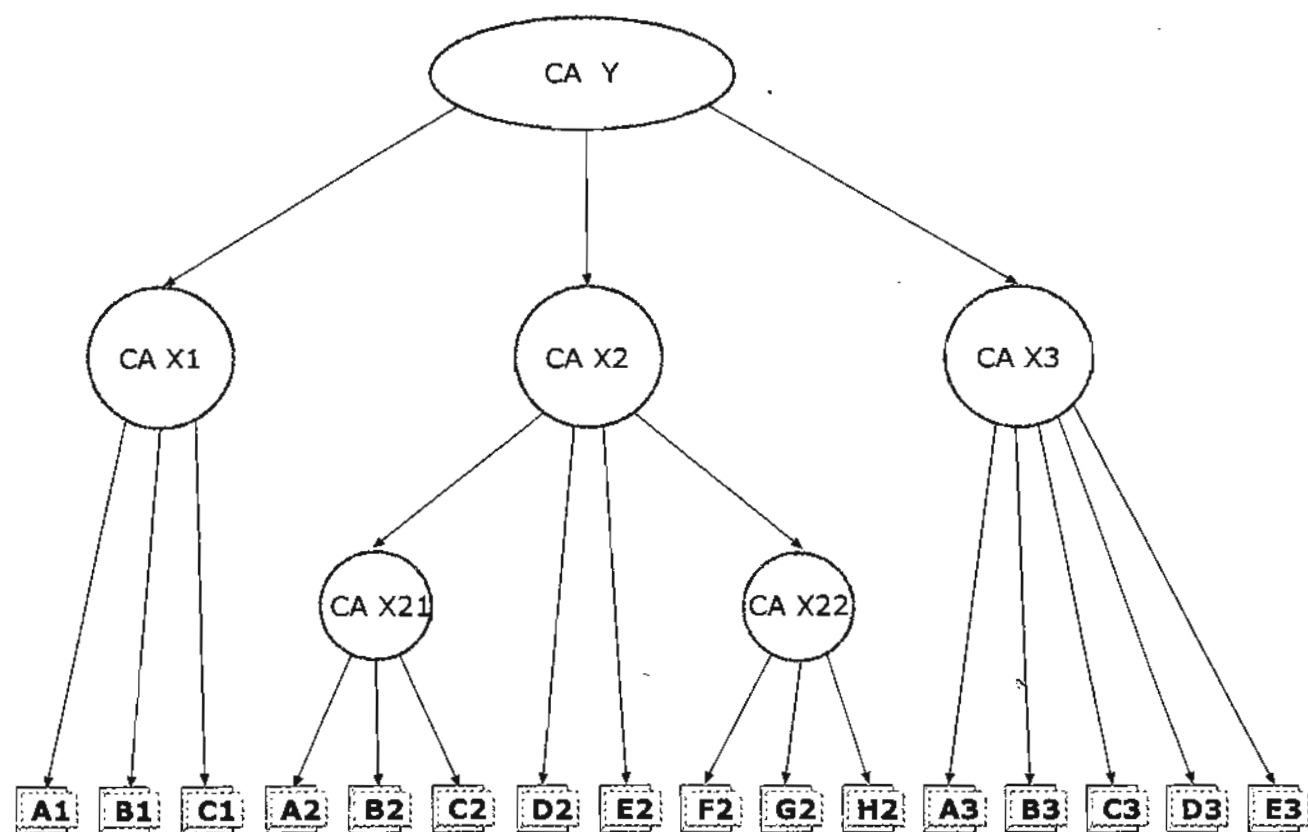


Figura 7.4. Organización jerárquica de CAs con sólo certificados directos.

Una solución drástica para obviar este problema consiste en que cada una de las entidades comunicantes posea permanentemente una colección completa de todos los certificados de todas las CAs que se encuentran en la ruta que une esa entidad con la CA de máxima jerarquía que ocupa la raíz del árbol. Cuando dos entidades comunicantes quieren establecer una relación de confianza, no tienen más que comparar esas dos tiras de certificados para encontrar si en ellas aparece en la estructura un punto común de confianza intermedio (por ejemplo, en la Figura 7.4, la CA X2 en la relación entre A2 y F2) o si, por el contrario, la confianza común reside en la raíz del árbol (la CA Y en el caso de la relación entre A1 y A2).

Esta solución para localizar el camino de certificación, que hemos calificado como drástica, sólo sirve para entornos no muy extensos en los que se puede implantar una política de almacenamiento y revocación de certificados muy rigurosa (este es el caso, por ejemplo, de la estructura adoptada por el Departamento de Defensa de Estados Unidos).

Cuando el dominio es más amplio y plural, una solución como la antes descrita es impracticable. Una situación de este tipo es la que se presenta para el establecimiento de correo electrónico seguro entre personas encuadradas en instituciones pertenecientes a distintos países (como es el caso de PEM, que más adelante comentaremos). En estos casos será necesario recurrir a procedimientos más trabajosos y menos inmediatos que el anterior.

Uno de ellos, que puede servir también para estructuras no estrictamente jerárquicas, consiste en disponer de **servidores** que «conozcan» el *árbol global de certificación* y que estén disponibles para que cualquier entidad comunicante pueda preguntar por la situación en el árbol de una determinada CA: es decir, qué otras CAs han emitido certificados a su favor y a qué otras CAs le ha generado ella certificados. Con sólo la primera de estas informaciones se puede averiguar el camino de certificación mediante un proceso iterativo similar al que describimos cuando se trataba de una estructura jerárquica en la que están presentes certificados directos e inversos. Con la información de qué otras CAs han sido certificadas por ella, el programa residente en la entidad interrogante adquiere datos que le permiten acelerar ese proceso de averiguamiento, tanto para resolver el problema del caso inmediato en el que se encuentre involucrado como para sucesivas búsquedas.

LIGERAS VARIACIONES DEL MODELO JERÁRQUICO PURO

Aun aceptando las características principales propias de las estructuras jerárquicas, existen algunos elementos que hacen que una organización de CAs de este tipo no resulte viable en muchas ocasiones, debido a lo cual se hace necesario buscar alternativas consistentes en introducir ligeras variaciones en el modelo jerárquico puro.

Una de las dificultades que surgen cuando se trata de establecer una estructura jerárquica tiene que ver con la existencia de una sola CA en el máximo nivel jerárquico. Si el dominio de seguridad abarca una sola organización política o administrativa, la cosa no presenta mayor problema, pero si forman parte de él varias empresas o instituciones distintas, todas ellas con las mismas ínfulas, puede resultar problemático decidir bajo qué organización debe estar la CA raíz. Un ejemplo típico lo constituye una infraestructura de seguridad en la que se integren infraestructuras pertenecientes a diversos países, como puede ser el caso de una organización de CAs para dar soporte a una aplicación de correo seguro de ámbito internacional.

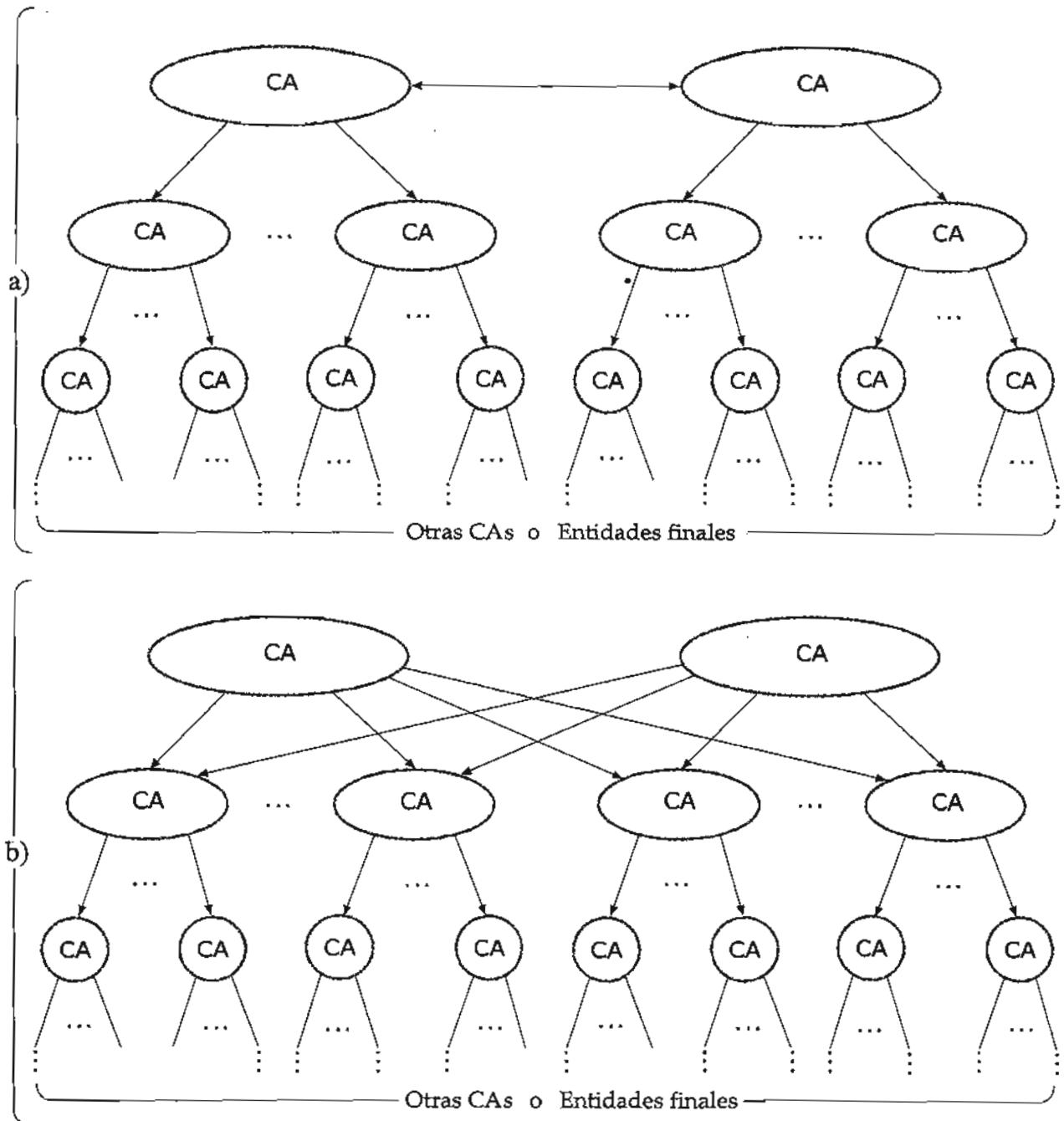


Figura 7.5. Dos formas de interconectar dos estructuras jerárquicas.

Una manera de salvar este escollo es desdoblar la CA de mayor jerarquía en tantas otras como organizaciones paritarias puedan existir. En la Figura 7.5 se presentan dos posibles soluciones³. Para simplificar el dibujo se ha optado por representar solamente dos estructuras jerárquicas interconectadas, pero las soluciones son igualmente válidas para cualquier número (no excesivo) de ellas.

En la opción a) lo que se genera es un certificado cruzado que relaciona las dos CAs de máximo nivel, mientras que en la opción b) cada una de las CAs de mayor rango reconocen, mediante un certificado, a las CAs existentes en el siguiente nivel jerárquico, que son las CAs donde se pueden definir los principales condicionantes de la política de seguridad.

El problema que presenta una solución tal que la recogida en la Figura 7.5 a) es que no es fácil de aplicar cuando aumenta el número de organizaciones paritarias que

se desea relacionar entre sí, ya que el número de certificados cruzados necesarios para interconectar todas las CAs de mayor jerarquía aumenta de forma alarmante, lo que hace impracticable una solución de ese tipo. Una evolución y mejora de esta estructura es la que se recoge en la Figura 7.6. En ella aparece una CA «neutral» que actúa como puente intermediario entre las distintas CAs raíz que encabezan cada uno de los árboles de certificación. En este caso, sólo es necesario contar con un certificado cruzado entre la CA puente y las restantes CAs raíz.

Por otra parte, en otro orden de cosas, en aquellas organizaciones jerárquicas donde existen muchas CAs y en las que participan un número elevado de entidades de comunicación, suele constatar con frecuencia que los caminos de certificación resultan **demasiado largos**.

Un primer efecto de la existencia de demasiadas CAs en el camino de certificación es que se produce una debilitación de la confianza a medida que aumenta el número de etapas o pasos en los que esa confianza es delegada. Como decíamos más arriba, una entidad, aparte de sí misma, en quien más confía es en la CA que le ha expedido su certificado, de forma que la confianza que tiene en la siguiente CA que aparece en su camino de certificación viene dada por la confianza que en ella tiene depositada «su» CA. No es lo mismo que alguien de tu confianza te relate un hecho que él o ella afirma haber presenciado, que si esa misma afirmación se basa en que ese alguien afirma que lo presenció otra persona en quien confía plenamente otra persona que es a su vez de la máxima confianza de otra persona en la que nuestro interlocutor confía: cuanto más larga sea la cadena, menos fiable resulta el testimonio. Es totalmente cierto que este ejemplo no es comparable con lo que ocurre en una cadena de certificación, ya que aquí ese depósito de confianza del que hablamos se materializa en un certificado emitido bajo una política y bajo unas responsabilidades que contrae quien

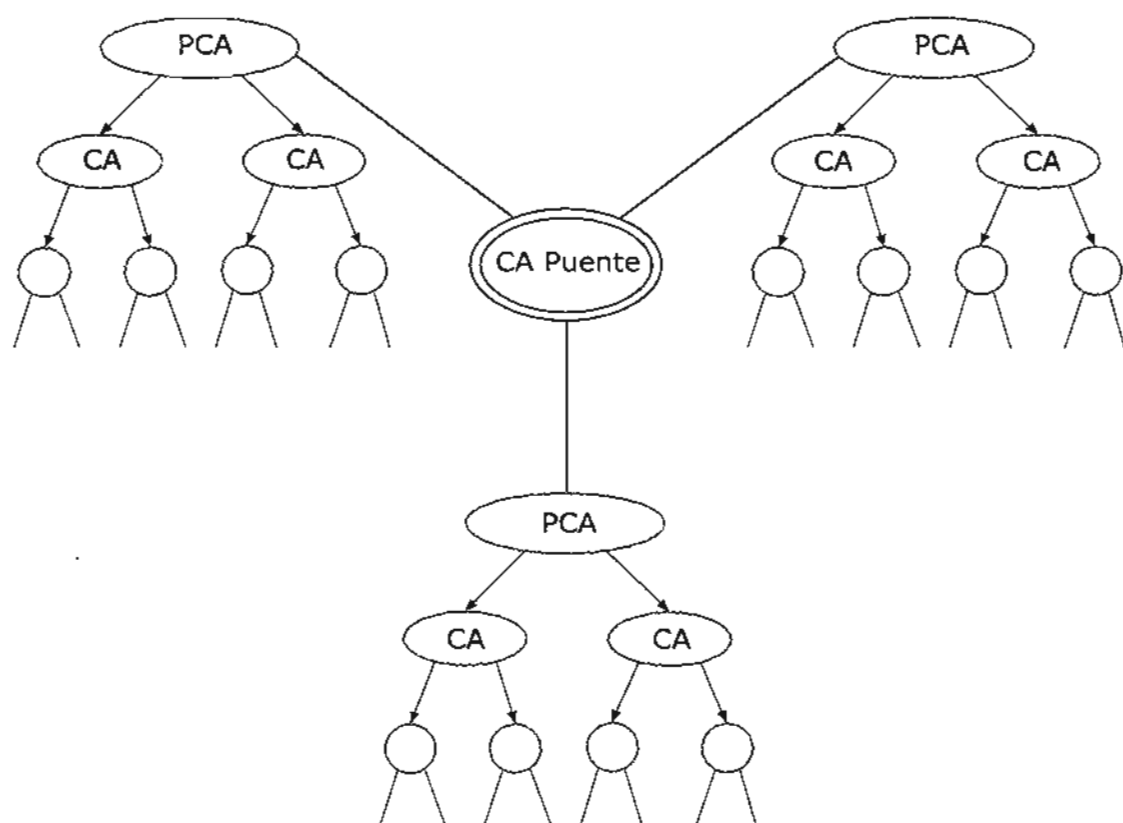


Figura 7.6. Estructuras jerárquicas interconectadas mediante una CA puente.

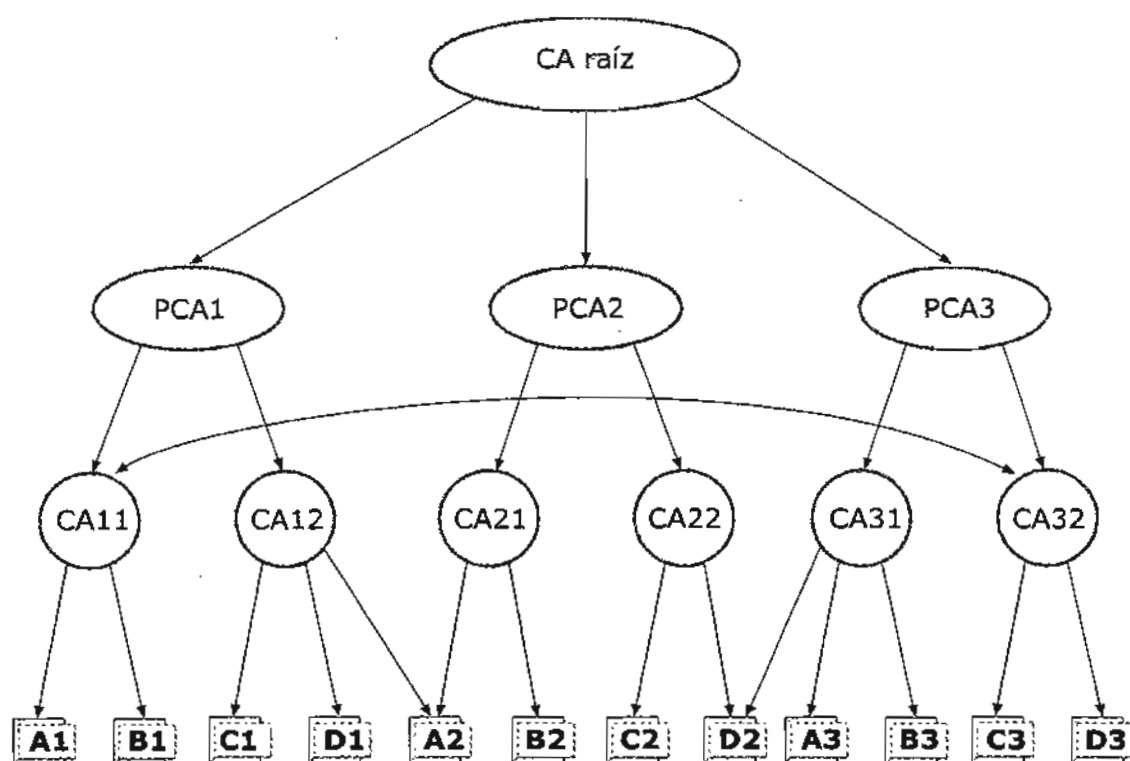


Figura 7.7. Estructura jerárquica con «atajos».

lo firma, pero también es cierto que la confianza sigue conservando un componente subjetivo bastante fuerte y que a medida que interviene un mayor número de actores aumentan los riesgos de errores o de descuidos que pueden redundar en una inadecuada certificación.

Aparte de la disminución de la confianza que conlleva el alargamiento de esos caminos, para muchas aplicaciones, tener que dar un rodeo demasiado grande en la búsqueda de certificados que respalden la confianza entre las partes resulta poco útil para los fines allí perseguidos.

Una solución a este problema puede encontrarse alterando el carácter estrictamente jerárquico de la estructura y estableciendo ciertos «atajos» que simplifiquen el camino necesario para obtener confianza en las claves públicas puestas en juego. En la Figura 7.7 se representa una situación de este tipo en la que se introducen dos alteraciones: de una parte, el establecimiento de certificados cruzados entre dos autoridades de certificación (CA11 y CA32); y de otra, que dos entidades (A2 y D2) comunicantes adquieran certificados no sólo de «su» CA natural, sino de otras CAs ubicadas en otros subárboles de la estructura.

Aunque a simple vista la cosa parece sencilla e inocente, pueden producirse problemas de competencia de políticas que necesitan ser analizados más detenidamente. No obstante, este tipo de soluciones permite adaptar la estructura de certificación a las necesidades concretas de cada grupo de usuarios, por lo que será tomada en cuenta de manera especial a la hora de diseñar modelos flexibles y eficaces (epígrafe 7.6).

7.3. EL DIRECTORIO X.500 COMO REPOSITORIO EN LAS INFRAESTRUCTURAS DE CERTIFICACIÓN

Como ya se ha dicho anteriormente, cuando los comités de ISO y de CCITT (posteriormente ITU-T) encargados de especificar los servicios y protocolos de seguridad en

redes telemáticas contemplaron la necesidad de definir el certificado digital y las CRLs, pensaron que el sitio adecuado para almacenar estas piezas de información sería el Servicio de Directorio, DS (*Directory Service*), más conocido como Directorio X.500, conforme al nombre de serie de la Recomendación de ITU-T en que está definido. Por esa razón fueron los grupos de trabajo encargados del conjunto de normas X.500 los que se responsabilizaron de la especificación de esas piezas de información que tan capital importancia tienen en la plasmación de los servicios de seguridad. La Recomendación X.500 está a su vez constituida por un conjunto de nueve normas (una de las cuales es la X.509). En ISO⁴ el Directorio se recoge bajo la denominación genérica ISO/IEC 9594.

El Directorio X.500 adquiere así un papel relevante como repositorio para el almacenamiento y recuperación de certificados y CRLs. Pero no es sólo eso: el hecho de presentar una organización jerárquica para el almacenamiento de información relevante acerca de las distintas entidades comunicantes lo convierten en referencia obligada a la hora de la definición estructurada de los nombres mediante los cuales reconocer de manera no ambigua a dichas entidades.

Inicialmente, el Directorio fue concebido como un sistema distribuido de rango planetario que soportaría una base de datos destinada a contener toda la información de las entidades comunicantes que fuese relevante a los efectos de las comunicaciones telemáticas. En aquel momento se pensaba que el Directorio, cuando se implantase, se extendería por todos los países de forma distribuida pero unitaria, actuando como un paraguas que cubriría todas las necesidades de almacenamiento de información relativa a las entidades comunicantes de todas las aplicaciones telemáticas.

Si bien es evidente que esa implantación universal de «El» Directorio no se ha producido, no es menos cierto que algunas infraestructuras de seguridad se basan en la presencia distribuida de «un» Directorio, además de que la estructura interna del Directorio X.500 ha sido tomada siempre como referencia a la hora de definir las distintas infraestructuras de certificación. Además, en la casi totalidad de los productos comerciales que sirven para la instalación de una Autoridad de Certificación se incluye una base de datos para el almacenamiento y recuperación de certificados (y otros datos relevantes) en la que la información está organizada conforme a la norma X.500, y se accede a ella mediante protocolos X.500.

En los párrafos que siguen se hará una descripción resumida de esta aplicación telemática considerándola soportada por un número indeterminado de máquinas distribuidas a todo lo ancho del dominio de seguridad. En el buen entendido de que es perfectamente particularizable para el caso en el que resida en un único sistema informático.

ESQUEMA GENERAL DEL DIRECTORIO

El sistema de Directorio consiste en una colección de equipos o sistemas abiertos (en la significación del término que se recoge en el Modelo OSI) ampliamente distribuidos, que cooperan entre sí para mantener una base de datos de rango universal, denominada Base de Información del Directorio o **DIB** (*Directory Information Base*).

En las versiones iniciales en español del CCITT se denominaba Guía a este servicio. Es fácil de adivinar que los antecedentes que pesaron sobre sus diseñadores se hallaban en la guía telefónica (en la situación en la que se encontraba este servicio en 1988) y en los servidores de nombres instalados en las redes. Una guía telefónica local es una base de datos (máxime si está informatizada), actualizada y gestionada por responsables locales del servicio, que puede ser consultada por los distintos usuarios

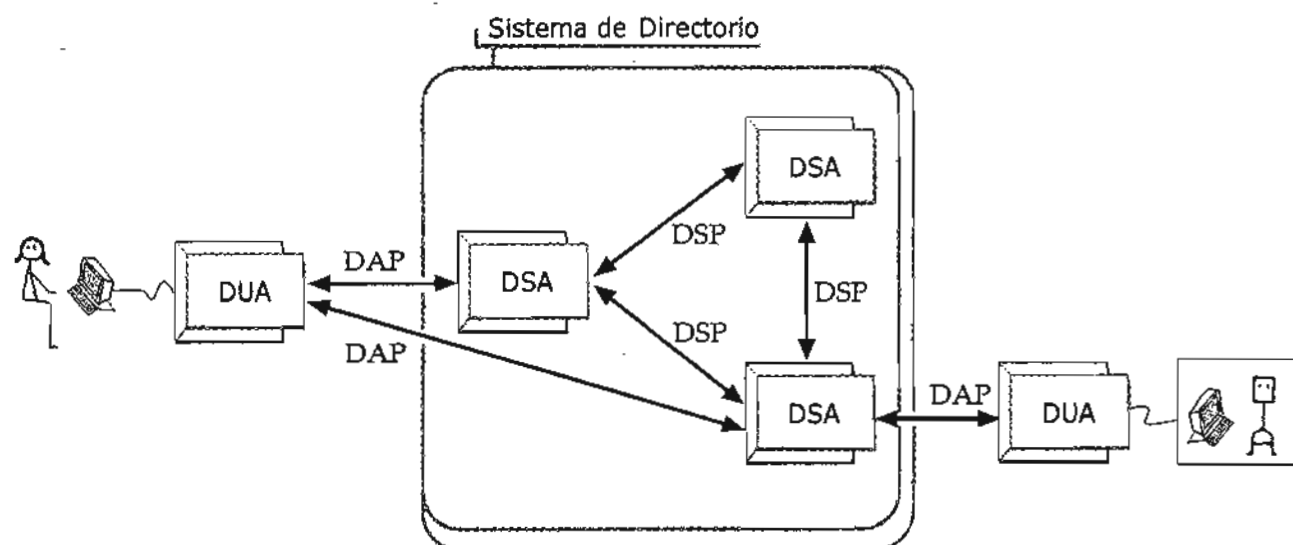


Figura 7.8. Esquema global del Servicio de Directorio.

para conocer el número de teléfono asignado a una determinada persona o entidad. El conjunto de todas las guías locales (el concepto de «local» dependerá de cada administración del servicio) a lo ancho de todo el mundo constituye de hecho una base de datos distribuida de rango internacional en la que, de una forma u otra, existen mecanismos para acceder desde uno de los «dominios» locales a la información contenida en cualquiera de los otros.

Continuando con el referente de la guía telefónica distribuida, se da la circunstancia de que si bien la información global es única, cada administración local «manda» sobre la parcela de esa información que cae bajo su responsabilidad. En cada una de estas parcelas es donde se encuentra la información local perfectamente actualizada, aunque pueden existir copias de esa información repartidas por otros lugares. Dependiendo del mecanismo que se elija para la obtención de esa copia (por ejemplo, poseer un ejemplar impreso de la guía de otra ciudad o disponer del volcado de la base de datos informatizada) así será el grado de «frescura» y actualización de la copia que se posee.

Estos mismos presupuestos son válidos para tener una idea global aproximada de lo que se pretendía con el Directorio: si la guía era un servicio imprescindible para hacer un uso adecuado del servicio telefónico, el Directorio contendría toda la información relevante de todas las entidades comunicantes (sólo en aquello que afecta a la comunicación) de todas las aplicaciones telemáticas que se pudiesen implantar en el ámbito de la interconexión de sistemas abiertos.

La Figura 7.8 recoge una representación del Servicio de Directorio constituido por el Sistema de Directorio⁵ propiamente dicho y por un conjunto de Agentes de Usuario de Directorio (DUAs), cada uno de los cuales sirve de interfaz con el sistema a un solo usuario. El Sistema de Directorio está constituido, a su vez, por un número indeterminado de Agentes del Sistema de Directorio (DSAs), cada uno de los cuales almacena una parte de la Base de Información (DIB) global que mantiene informaciones relativas a los sujetos que intervienen en las comunicaciones telemáticas. Los DSAs pueden encontrarse distribuidos geográficamente de forma que cada uno de ellos esté gestionado por un administrador distinto.

La cooperación entre los distintos DSAs para mantener la información global se lleva a cabo a través del Protocolo de Sistema de Directorio (DSP, *Directory System Protocol*) que regula el diálogo mediante el cual los DSAs se intercambian informa-

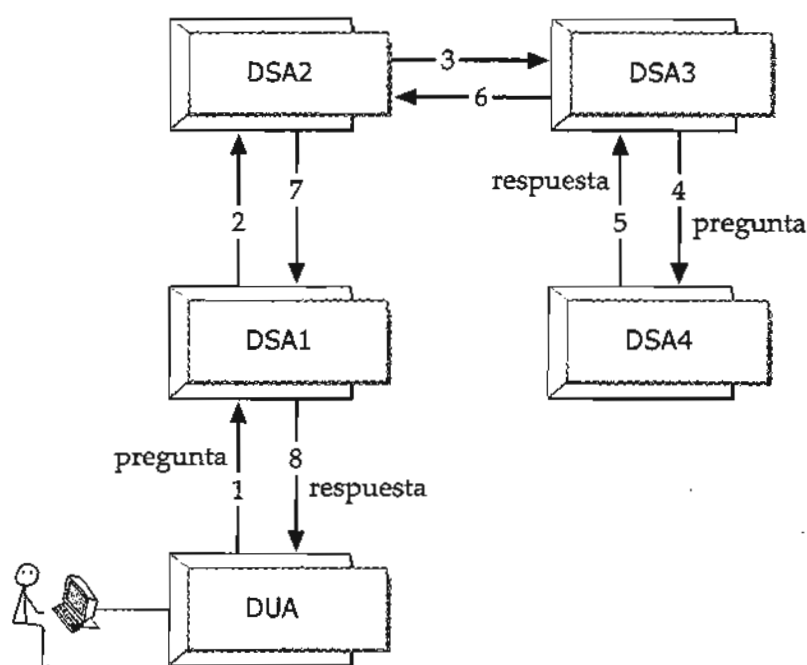


Figura 7.9. Cooperación entre DSAs mediante «encadenamiento».

ción. Existen especificaciones detalladas para regular las distintas posibilidades que ofrece el Directorio para que unos DSAs mantengan copia (con un cierto nivel de actualización) de la información contenida en otros DSAs, así como de los distintos mecanismos existentes para que los DSAs cooperen entre sí en la búsqueda de una información que haya solicitado un usuario.

Ante una petición de servicio, un DSA puede seguir varias formas distintas de operación. Una de ellas es llevar a cabo la operación solicitada de manera local, situación que sólo se producirá si los datos relacionados con esa información se encuentran en el fragmento de la DIB que está bajo su incumbencia. Si el DSA no posee esa información puede recurrir a pedir ayuda a otros DSAs hasta agotar las posibilidades de encontrarla, o bien responder al DUA del usuario dándole una pista sobre en cuál o cuáles otros DSAs puede encontrarse la información requerida. Naturalmente, la primera de las dos opciones resulta ser la que ofrece mejor servicio al usuario, aunque requiere algoritmos de cooperación mutua entre DSAs más evolucionados. En la Figura 7.9 se representa uno de los posibles mecanismos que han sido propuestos para llevar a cabo esa cooperación. Se denomina de «encadenado» y consiste en que el DSA que ignora la respuesta a una pregunta que le ha sido formulada se la pasa a otro DSA donde supone que puede encontrarse esa respuesta. Al final de la cadena, la respuesta, caso de que haya sido encontrada, viaja en orden inverso hasta llegar al usuario que originó la pregunta.

La solución más adecuada para propiciar que un DSA esté en condiciones de informar acerca de la información contenida en otro DSA es conseguir que posea copia actualizada de aquellos otros «trozos» del árbol que con mayor frecuencia los usuarios deseen consultar. Para replicar y actualizar la información entre varios DSAs, la norma X.500 especifica un protocolo de *sombreado* (*shadowing*) y otros protocolos de gestión que, por razones de oportunidad, no vamos a comentar aquí.

Obviamente, cuando sólo existe una máquina que contiene toda la información del Directorio (situación que, como antes hemos dicho, se presenta en multitud de implementaciones de una CA aislada) solamente existe un DSA que almacena todo el conjunto de informaciones disponibles, por lo que no cabe hablar ni de cooperación entre DSAs ni de Protocolo del Sistema de Directorio.

En cualquier caso, cada usuario de Directorio, Figura 7.8, tiene asociado un Agente de Usuario de Directorio (DUA) a través del cual realiza las preguntas y obtiene las respuestas. Además, el usuario a través del DUA podrá almacenar determinada información en el Directorio (en los casos en que le esté permitido). El protocolo mediante el cual el DUA accede a un DSA para llevar a cabo una de las operaciones permitidas se denomina DAP (*Directory Access Protocol*). (Más adelante se presenta una breve reseña de cuáles son esas operaciones.)

Las implementaciones existentes utilizan mayoritariamente una simplificación de este protocolo: es el denominado LDAP (*Lightweight Directory Access Protocol*), definido por el IETF, que aunque no contempla algunas de las funcionalidades definidas en la norma X.500, se apoya en una infraestructura de comunicación muy simple (utiliza TCP/IP), y lleva a cabo la representación interna de la información mediante cadenas de caracteres, evitando las complicadas estructuras en que se basa el DAP. Se estima que LDAP presenta aproximadamente el 90 por 100 de las funcionalidades del DAP, pero sus exigencias de recursos se reducen al 10 por 100. Por ese motivo, el LDAP es fácilmente instalable en máquinas dotadas de pocos recursos, lo que ha propiciado su aceptación masiva.

En la Figura 7.8 también se refleja cómo un DUA puede acceder al Servicio de Directorio comunicándose con cualquiera de los DSAs en él existentes, aunque lo normal es que por cuestiones organizativas y de proximidad le sea más cómodo considerarse «adscrito» a un DSA concreto al que habitualmente dirigirse para realizar todas las consultas, con la esperanza de que aquellas que ese DSA no pueda resolver por sí mismo las resuelva mediante la cooperación con los restantes DSAs.

ÁRBOL DE INFORMACIÓN DEL DIRECTORIO (DIT)

La información de la DIB se encuentra agrupada en *entradas* o *inserciones*, cada una de las cuales consiste en una colección de informaciones sobre un *objeto* y representa a una entidad que puede ser una persona, una organización o un elemento automático. Estas entradas están organizadas jerárquicamente en forma de árbol, que es conocido como Árbol de Información del Directorio (DIT). Este árbol, cuya raíz es única, se halla repartido entre los distintos DSAs, cada uno de los cuales posee, por así decir, un trozo de rama «en propiedad» e información relativa a cómo y dónde están las informaciones que él no posee. El DIT representa, por tanto, la estructura lógica de la información contenida en el Directorio, en tanto que los DSAs serán agentes telemáticos (cada uno de ellos ubicado físicamente en una máquina), que mantienen una parte concreta de la base de datos, y que son accesibles por todos los DUAs de las entidades comunicantes.

En la Figura 7.10 se representa la estructura del Árbol de Información del Directorio. En la concepción «universal» del Directorio, la raíz es común para todos los países y las entradas ligadas directamente a ella (indicador «C», del inglés *country*) representan países u organizaciones internacionales de amplia implantación (como puede ser el caso de ISO). La *entrada objeto* (representada mediante un rectángulo en la figura) es la que realmente contiene información, mientras que la *entrada alias* (representada por un triángulo) lo que contiene es un puntero dirigido hacia una entrada objeto que es donde reside realmente la información. La opción del *alias* se utiliza para que, en caso de que una misma entidad se encuentre, por razones organizativas, en dos ramas distintas del árbol, no tenga que mantener obligatoriamente su información en dos sitios de forma simultánea (por ejemplo, una persona puede per-

OPERACIONES SOBRE EL DIRECTORIO

Todos los servicios son proporcionados por el Directorio en respuesta a solicitudes formuladas por los DUAs, que son los agentes que representan a los usuarios. Existen peticiones que sirven para interrogar al Directorio y otras que permiten modificarlo.

Los servicios de interrogación son:

- *Lectura (read)*: Una petición de lectura se dirige a una entrada o inserción determinada y da como resultado la devolución de los valores de los atributos especificados en la pregunta (que pueden ser todos los que constituyen la entrada o solamente algunos concretos). No presente en LDAP.
- *Comparación (compare)*: Es una pregunta dirigida a un atributo concreto de una entrada y provoca que el Directorio verifique si el valor suministrado en la pregunta concuerda con el valor almacenado de dicho atributo.
- *Listado (list)*: Es una petición que hace que el Directorio devuelva la lista de los subordinados inmediatos de una determinada entrada del DIT. No presente en LDAP.
- *Búsqueda (search)*: Una operación de búsqueda da como resultado que el Directorio devuelva información de todas las entradas dentro de cierta porción del DIT que satisfagan el *filtro* especificado en la pregunta. La información devuelta comprende los valores de algunos o de todos los atributos de dichas entradas (según se haya formulado la interrogación). Un *filtro* expresa una o más condiciones que la inserción debe satisfacer para ser devuelta como resultado y comprende un conjunto de reglas de concordancia (*matching rules*) para ser aplicadas a los valores de los atributos especificados en la búsqueda.

Los servicios de modificación son:

- *Adición de entrada (add entry)*: Es una operación que sirve para añadir una nueva hoja (una entrada objeto o una entrada alias) en determinado punto del DIT.
- *Supresión de entrada (remove entry)*: Sirve para borrar una entrada del árbol de información del Directorio.
- *Modificación de la entrada (modify entry)*: Es una operación que sirve para introducir cambios en la información almacenada correspondiente a una determinada entrada. Los cambios permitidos son la adición, eliminación o reemplazo de atributos o de valores de atributo.
- *Modificación del nombre X.500 (rename o modify DN)*: Sirve para modificar el *Nombre Distintivo* (DN) de una entrada. Esta modificación tiene por efecto (como puede deducirse de la descripción que sobre la estructura del nombre X.500 se hace en el siguiente apartado) mover de sitio la entrada dentro del DIT. Si la entrada tiene subordinados, todos ellos son denominados de nuevo y desplazados, consecuentemente, dentro del DIT.

Los servicios relacionados con la conexión al Directorio son:

- *Conexión (bind)*: Permite a un cliente iniciar una sesión.
- *Desconexión (unbind)*: Permite cerrar la sesión iniciada.
- *Abandono (abandon)*: Es una petición que indica que el usuario ya no está interesado en que el Directorio continúe procesando una petición anterior.

Para administrar adecuadamente estas operaciones es preciso regular convenientemente los derechos de acceso de los distintos usuarios. Sobre todo, las operaciones que modifican el contenido del Directorio pueden causar disfunciones notables si son llevadas a cabo por usuarios no autorizados para ello. En el Directorio existen mecanismos muy elaborados que permiten definir un *servicio de control de acceso* muy refinado mediante el que se pueden proteger tanto entradas completas como determinados atributos e, incluso, valores concretos de atributo. No obstante, a falta de estos mecanismos de protección, siempre se puede garantizar la integridad del Directorio permitiendo solamente al gestor del mismo la realización de operaciones que den lugar a la modificación del contenido o de la estructura de la información almacenada.

NOMBRES X.500

Cada *entrada* del DIT debe ser reconocida de forma única por el Directorio, por lo que debe tener un nombre único y distintivo en todo el entorno telemático. Si tenemos en cuenta que el Directorio pretende abarcar a todos los países y servir de apoyo a todas las aplicaciones telemáticas, la estructura de nombres debe estar compuesta de elementos fácilmente deducibles a partir de la ubicación lógica y administrativa de la entidad comunicante (persona o sistema automático) de que se trate.

Como ya adelantamos en el capítulo anterior (epígrafe 6.4), el *nombre X.500*, que es utilizado en los campos del certificado X.509, está compuesto por campos jerárquicamente organizados y se denomina *Nombre Distintivo* o DN (*Distinguished Name*). Los elementos o campos de que consta el DN se derivan directamente de la ubicación de la entidad nombrada dentro de la estructura del DIT y cada uno de estos campos diferenciadores se denomina RDN (*Relative Distinguished Name*).

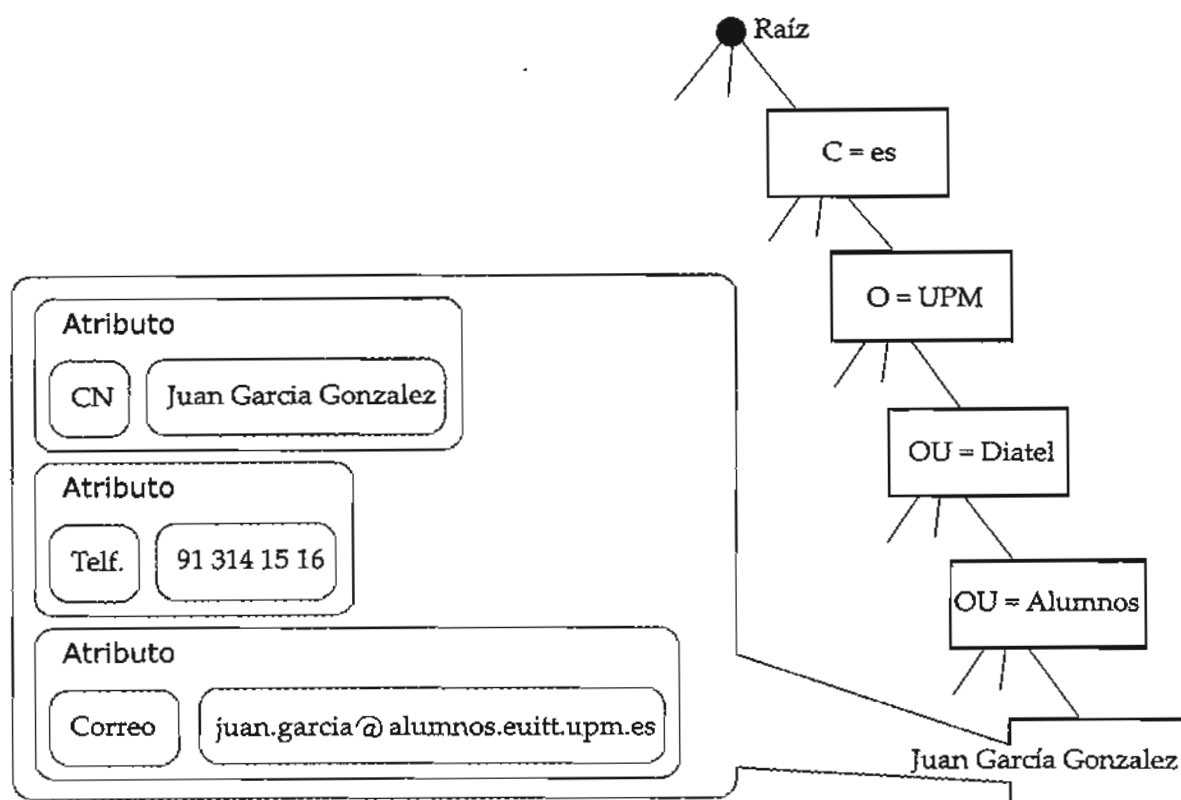


Figura 7.11. Ejemplo de ubicación de una entidad comunicante en el DIT.

Retomemos el ejemplo que en dicho epígrafe utilizamos. Suponíamos la existencia de un alumno, Juan García González, que está matriculado en asignaturas del departamento Diatel de la Universidad Politécnica de Madrid. En la Figura 7.11 se representa la hipotética ubicación en el DIT de la entrada a él correspondiente. El RDN de mayor jerarquía correspondería al país en el que está ubicada la organización a la que pertenece la entidad comunicante nombrada. El identificador es «C» (del inglés *country*) y en este caso sería C = es. En segundo lugar del escalafón jerárquico aparece la entrada correspondiente a la *Organización*, que aquí suponemos que se corresponde con la Universidad Politécnica de Madrid cuyo nombre relativo es: O = UPM.

A continuación puede aparecer un número indeterminado de campos correspondientes a diversas unidades organizacionales jerárquicamente dependientes de la institución catalogada como «organización». El identificador de estos campos es «OU», del inglés *Organization Unit*. En este ejemplo hemos supuesto que existen solamente dos de estas unidades organizativas: el Departamento y la que agrupa al conjunto de alumnos, cada una de las cuales tiene su correspondiente entrada dentro del DIT. Los valores de estos dos RDNs serían: OU = Diatel y OU = Alumnos.

Por último, aparece la entrada correspondiente al alumno Juan García González, que estará constituida, a su vez, por un conjunto de atributos y sus correspondientes valores. Para simplificar el ejemplo hemos supuesto que solamente son tres los atributos que contiene esa entrada: el nombre, el número de teléfono y la dirección de correo electrónico. (A los efectos de la incorporación de ese Directorio como repositorio en una hipotética infraestructura de certificación, un atributo de capital importancia sería el destinado a contener los certificados X.509 que hubiesen sido expedidos a nombre de ese usuario.) El RDN que contiene el nombre propio (*Common Name*) del usuario es CN = Juan García González. Como resultado de todo ello, el conjunto de los distintos RDNs es el DN o nombre X.500 completo:

{ C = es, O = UPM, OU = Diatel, OU = Alumnos, CN = Juan García González }

En el ejemplo se ha buscado deliberadamente que la dirección de correo electrónico (juan.garcia@alumnos.euitt.upm.es) esté constituida por campos en los que aparecen organizaciones diferentes a aquellas que determinan el DN. Se ha supuesto que en lo relativo al correo ese usuario depende de una estafeta, denominada *euitt*, que no aparece en el nombre X.500. Ello ejemplifica el que regularmente el nombre de directorio (caso de que el usuario lo tenga asignado) no pueda deducirse de la dirección de correo electrónico, ni viceversa, lo que justifica la inclusión de otros nombres alternativos en la versión 3 de certificado X.509.

7.4. INFRAESTRUCTURA DE CERTIFICACIÓN PEM

La Infraestructura PEM es la primera propuesta sólida que se hizo para solucionar el problema de la organización de múltiples CAs en un dominio extenso, plural y complejo. Se trata de un modelo jerárquico en forma de árbol que ha servido como punto de partida y comparación para todas las PKIs que con el tiempo han venido apareciendo. Por ello, ha sido y es referencia obligada a la hora de abordar la tarea de diseñar una infraestructura de este tipo. Su formulación estable data de 1993⁶ y fue publicada por la *Internet Society* a través del grupo que diseñó la aplicación de correo PEM (*Privacy Enhancement for Internet Electronic Mail*).

La característica más significativa de la Infraestructura PEM es que a partir de un determinado nivel jerárquico existe un isomorfismo total entre el árbol de CAs y el

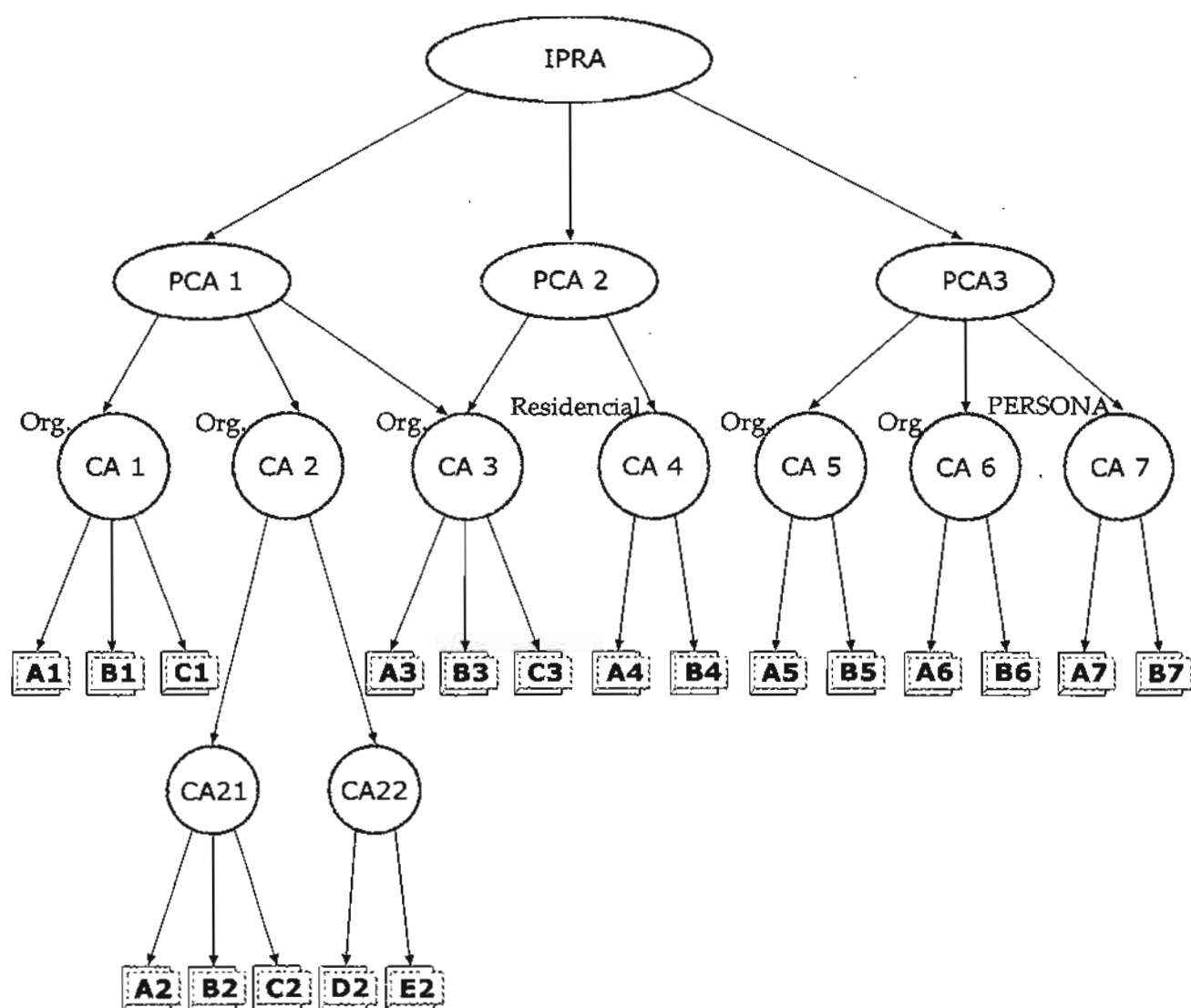


Figura 7.12. Arquitectura del modelo PEM.

Árbol de Información del Directorio (DIT). También la unicidad de nombres dentro de la estructura es garantizada mediante el empleo de nombres distintivos (DNs) X.500. En ello radican sus principales ventajas y sus principales inconvenientes.

DISTINTOS TIPOS DE CAS PRESENTES EN EL MODELO PEM

En el modelo PEM se contemplan varios tipos de autoridades de certificación, cada uno con unas funcionalidades distintas. Éstos son:

- *Autoridad de Registro de Políticas de Internet (IPRA)*

Existe una sola IPRA que ocupa (Figura 7.12) el más alto nivel de la jerarquía de certificación. Su clave pública es conocida en todo el dominio y se encarga de registrar y certificar a las distintas PCAs, dando publicidad al documento descriptivo de las políticas por ellas adoptadas y comprobando la concordancia de esas políticas con la política general establecida por la IPRA.

- *Autoridades de Certificación de Políticas (PCAs)*

En el segundo escalón jerárquico se sitúan las PCAs, que como su nombre sugiere son las encargadas de definir una determinada política que debería ser

conocida y cumplida por todas las CAs y entidades que de ella dependen. En la especificación de esa política deberían contemplarse una serie de aspectos como son: Identidad de la PCA (DN e identificación de sus responsables), ámbito en el que se prevé su servicio, procedimientos de registro, tamaño de las claves, mecanismos de protección y custodia de las claves privadas, periodos límite de validez de los certificados, periodicidad de emisión de sus propias CRLs, lugares de consulta de las CRLs emitidas por las CAs de ella dependientes, riesgos y responsabilidades financieras que se pueden derivar de las transacciones llevadas a cabo conforme a la política en ella definida, etc.

- *Autoridades de Certificación de entidades*

Son las entidades que emiten certificados a los usuarios y entidades finales, aunque pueden desglosarse jerárquicamente, a su vez, en tantas otras CAs como la complejidad organizativa o geográfica del subdominio que abarcan lo aconseje. Deberían cumplir lo regulado en la política definida por la PCA de la que dependen y controlar los DNs de las entidades a las que emitan certificados para garantizar la unicidad de nombres, evitar inconsistencias y comprobar que todos los nombres cumplen la **regla de subordinación nominal conforme a la estructura del DIT** del Directorio X.500.

Aunque las infraestructuras jerárquicas están especialmente adaptadas a las necesidades de las personas y sistemas agrupados en organizaciones, para no dejar fuera de la infraestructura a nadie, se han previsto (más en la teoría que en la práctica) tres tipos distintos de CAs:

- *CAs Organizacionales*

Emiten certificados a entidades (u otras CAs jerárquicamente dependientes de ella) encuadradas en cualquier tipo de organización, sea de tipo gubernamental, académico o empresarial. Según lo comentado más arriba, estas CAs constituirían la inmensa mayoría de las presentes en la infraestructura.

- *CAs Residenciales*

Están destinadas a emitir certificados a favor de individuos no encuadrados en organizaciones concretas presentes en la infraestructura. Están pensadas para certificar a entidades gubernamentales civiles aisladas. Debido a ello, en la mayoría de los casos no será posible mantener la subordinación nominal, por lo que todas las CAs residenciales existentes deberán cooperar entre sí para garantizar la unicidad de nombres y la consistencia del modelo.

- *CAs PERSONA*

Están previstas para proporcionar certificados a personas que quieran mantener su anonimato en las comunicaciones seguras que lleven a cabo utilizando la infraestructura PEM. Por ello, el DN que aparece en el certificado no se corresponde con su verdadera identidad ni con su ubicación legal en el DIT.

En la Figura 7.12 hemos supuesto que la CA4 sea del tipo *Residencial* y la CA7 del tipo *PERSONA*. Todas las restantes vamos a suponer que son CAs «normales», es decir, de tipo *Organizacional*.

La variedad de CAs contempladas en el modelo y la diferenciación en lo que se refiere a las funcionalidades asignadas a cada una de ellas hace que, basándose en este modelo, pueda diseñarse una infraestructura de certificación perfectamente escalable que puede extenderse a entornos muy amplios. Además, el hecho de que la política de certificación pueda ser diferente en los distintos subdominios introduce un factor añadido de flexibilidad que favorece esa expansión.

SUBORDINACIÓN DE NOMBRES Y POLÍTICAS DE CERTIFICACIÓN

Exceptuando esos dos casos especiales que representan las CAs Residenciales y las CAs PERSONA, la primera característica importante de las CAs presentes en la Infraestructura PEM consiste en que:

- a) La IPRA no le exige a las PCAs ninguna restricción respecto a su DN. Esto mismo ocurre con las PCAs respecto a las CAs a ellas subordinadas.
- b) A partir de esa primera CA certificada, la regla de subordinación nominal restringe radicalmente la identidad de las entidades que pueden ser certificadas por ella: solamente puede emitir certificados a aquellas entidades cuyos DNs sean subordinados del DN de esa CA dentro del árbol de nombres inherente al DIT del Directorio (y esto es igualmente válido tanto si es una entidad final como si se trata de otra CA subordinada).

La segunda característica significativa de la Infraestructura PEM está relacionada con el tratamiento que aquí se hace de la política de certificación. En principio, el objetivo a cubrir tanto por una CA aislada como por un conjunto de ellas trabajando coordinadamente es conseguir que una entidad de comunicación que opere en ese entorno tenga una confianza razonable en la validez de las claves públicas de las restantes entidades. No obstante, en el modelo PEM se introduce un nuevo concepto: esa validez de las claves públicas tiene que estar soportada, en todos los casos, por una política de certificación concreta deducible del esquema de organización de las CAs. (Esta innovación representa un argumento más a la hora de valorar la enorme contribución que para el avance de las técnicas de seguridad en redes ha supuesto la propuesta PEM en su conjunto.)

En relación con la política de certificación bajo la que operan las distintas entidades comunicantes certificadas, se pueden presentar varios casos:

1. En el caso más simple, una determinada PCA (supongamos la PCA1 de la Figura 7.12) sólo registra en la IPRA una política de certificación, por lo que obtiene de ésta un único certificado. Por tanto, todas las CAs que dependen de ella trabajan bajo una sola política. Una entidad certificada (supongamos que A2) subordinada de una de esas CAs operará siempre bajo la misma política: *aquella que se deduce de la PCA presente en su camino de certificación*.
2. Una misma PCA (supongamos la PCA3) puede registrar en la IPRA más de una política, obteniendo de ésta más de un certificado. La IPRA exige un par de claves distintas para cada certificado que le emita a PCA3 (PEM elige esta opción para hacer más sencilla la evaluación de la política bajo la que se opera, aunque también hubiera podido usarse el mismo par de claves para todos los certificados que recibiera PCA3). Por tanto, las CAs subordinadas

de PCA3 pueden recibir de ésta más de un certificado, cada uno emitido bajo una política distinta.

Consecuentemente, una entidad certificada (supongamos que A5) subordinada de una de esas CAs organizacionales podría operar bajo *varias políticas distintas asociadas todas ellas a un mismo camino de certificación (cada una bajo un certificado de PCA concreto)*.

3. La arquitectura permite un pequeño «atajo»: una CA organizacional puede ser certificada por más de una PCA (caso de CA3 en el ejemplo de la figura). Por ello, una entidad certificada dependiente de CA3 (supongamos que A3) podría operar bajo *más de una política, asociada cada una de ellas a un camino de certificación distinto*.

Obviamente, pueden darse situaciones en las que concurren los casos segundo y tercero de la relación anterior.

En su concepción inicial, la Infraestructura PEM se basa en la versión 1 del Certificado X.509, lo que limita bastante la información que se puede incluir para determinar la política de certificación bajo la que opera un certificado de entidad final. (Por este y otros motivos es por lo que se propusieron las extensiones de la versión 3 del Certificado, una de las cuales es *CertificatePolicies*.) Una solución a este problema es incluir en el campo opcional *Issuer-Certificate* (también presente en la versión 1) todos los certificados que conforman el camino de certificación hasta la IPRA, pero esto puede resultar una carga pesada cuando la estructura está muy ramificada y existen muchas CAs en el camino de certificación.

Para garantizar la unicidad de nombres (que es una exigencia fundamental en este modelo) es necesario que la IPRA mantenga una base de datos en la que consten los nombres de las PCAs registradas y la clave pública asignada a cada política. Este mismo rigor en cuanto a la unicidad de nombres es necesario mantenerlo en los restantes escalones jerárquicos, aunque gracias a la regla de subordinación de nombres que opera a partir de las CAs Organizacionales, se limitan los riesgos de inconsistencia de la arquitectura.

VENTAJAS E INCONVENIENTES DEL MODELO PEM

La principal ventaja de la Infraestructura PEM proviene de la sencillez y claridad de su organización debido al isomorfismo con el DIT que impone en la mayor parte del entorno de seguridad. De esta misma circunstancia se derivan algunos de sus principales inconvenientes.

De una parte, todas las entidades presentes en el entorno tienen que poseer un DN X.500. Pero, al mismo tiempo, PEM no impone la existencia de un Directorio distribuido como lugar obligado para depositar los certificados y CRLs, con lo cual muchas de las entidades finales certificadas no estarán adscritas a ese servicio y, sin embargo, se verán forzadas a poseer obligatoriamente un nombre de Directorio. Esta aceptación a medias del Directorio (en un contexto de descuelgue generalizado de la *Internet Society* respecto de las aplicaciones desarrolladas en el ámbito de ISO e ITU-T)⁷ tuvo la virtud de afianzar definitivamente la concepción y estructura del Certificado X.509, pero no aprovechó la oportunidad para promover la implantación de un Directorio único distribuido internacionalmente y mantenido de forma coordinada, lo que hubiera supuesto, a buen seguro, un mayor éxito de implantación para esta plataforma de correo y para muchas otras de las soluciones que se están procurando para satisfacer los requisitos de seguridad en las redes.

Una gran ventaja de este modelo es posibilitar la existencia de una red de CAs perfectamente adaptada para su expansión en múltiples países y organizaciones internacionales, facilitando, no obstante, que las distintas ramas que componen ese gigantesco árbol puedan tener toda la autonomía que requieran, tanto en la definición de la política como en la estructura jerárquica particular de cada uno de los subdominios que se definan. Para una aplicación del mismo tipo que el correo electrónico, que abarca todo el planeta y en el que todas las entidades comunicantes pueden tener unos requisitos de seguridad semejantes, la solución propuesta en PEM es muy adecuada.

Como contrapartida, una infraestructura jerárquica pura como esta (con pocos atajos posibles) representa un grave inconveniente a la hora de diseñar la estructura de certificación para un grupo o consorcio constituido por un conjunto de sucursales distribuidas entre varios países. Pensemos, por ejemplo, en un proyecto de investigación constituido por varios grupos de trabajo cada uno presente en una Universidad distinta de un país distinto. En efecto, cada una de estas sucursales podría tener su parcela local de certificación perfectamente controlada y adaptada a sus necesidades. Además, debido a la proximidad virtual existente entre las CAs y el conjunto de miembros que constituyen ese grupo local, el nivel de confianza en el proceso de certificación puede ser muy alto (independientemente de lo alejada de la raíz que se encuentre esa «parcela» de certificación). Sin embargo, llegado el momento en el que dos entidades finales (pertenecientes a grupos distintos) necesiten establecer una comunicación segura, para establecer un camino de confianza se verán obligadas a incluir multitud de CAs, teniendo que llegar posiblemente hasta la misma IPRA, ascendiendo y bajando a través del árbol definido por la estructura del modelo PEM. Esta *distancia* lógica conlleva una elevada carga en el número de certificados necesarios para el establecimiento de comunicaciones seguras y, sobre todo, una inadmisiblemente disminución de la confianza en las claves públicas puestas en juego (tanto más inadmisiblemente cuanto mayor sea el valor de las transacciones que se establezcan en ese dominio de seguridad).

Esta dificultad que tiene la Infraestructura PEM para establecer atajos procede no sólo del excesivo isomorfismo que presenta con respecto al DIT, sino también de la rigidez de la jerarquía establecida. De hecho, si ese hipotético proyecto de investigación usase un Directorio distribuido en el que cada grupo remoto dispusiese de una DSA para gestionar su trozo de rama del DIT, cualquier DUA podría acceder directamente y sin restricciones a cualquiera de esas DSAs, de forma que con un adecuado control de acceso, los miembros de proyecto dispondrían, virtualmente, de un repositorio particular, privado y «aislado» del resto del Directorio. De forma semejante, sería de desear que la Infraestructura de Certificación le permitiese también a este grupo de usuarios establecer un dominio de seguridad en el que el esquema de certificación fuese también particular, privado y aislado (desde un punto de vista virtual, al menos).

En resumen, la rigidez de la jerarquía de la Infraestructura PEM es su mayor inconveniente, por lo que será necesario buscar soluciones y modificaciones que, respetando y aprovechando las considerables ventajas de este modelo, resuelvan sus deficiencias en cuanto a adaptabilidad a situaciones particulares. De ello se discutirá en el epígrafe 7.6.

7.5. OTRAS INFRAESTRUCTURAS CON CAS ORGANIZADAS JERÁRQUICAMENTE

Las propuestas de infraestructura planteadas en los apartados anteriores son de utilidad para ser aplicadas en situaciones en las que la dependencia organizativa es muy clara y no se presentan escenarios particulares en los que aparezcan dominios de seguridad que incluyan entidades encuadradas en sitios dispersos del árbol jerárquico. En este epígrafe se presentan dos ejemplos claros de este tipo de arquitecturas de propósito general.

En primer lugar se estudia el caso de una infraestructura que sirva para dar solución a las necesidades de comunicación que aparecen dentro de las administraciones públicas y a continuación se describe una infraestructura de carácter privado y comercial desarrollada con un propósito muy específico: el derivado del uso de tarjetas de crédito.

UNA INFRAESTRUCTURA PARA LAS ADMINISTRACIONES PÚBLICAS

Todas las administraciones públicas, tanto de carácter local como estatal, presentan un perfil bastante similar: están constituidas por departamentos que dependen de un centro político unificado, pero que mantienen particularidades específicas que hacen que también deba ser específica la política de seguridad que sea necesario implantar para proteger las comunicaciones telemáticas que se den en su ámbito.

En la Figura 7.13 se representa el esquema propuesto en la Infraestructura MISSI⁸ del Gobierno Federal de Estados Unidos, muy similar al modelo PEM, que puede servir de modelo organizativo para cualquier otro tipo de administración pública. Si se trata de una infraestructura de ámbito estatal, el primer nivel jerárquico que aparece después de la CA raíz corresponderá a los ministerios, y cabe suponer que sean necesarios varios escalones jerárquicos más para poder particularizar, dentro de cada uno de ellos, las necesidades de las distintas secretarías y dependencias. En el caso de administraciones locales, es de esperar que la pluralidad de requisitos se reduzca y, con ello, la complejidad que debe presentar la estructura. En cualquier caso, todo dependerá del tamaño y complejidad del estado o comunidad de que se trate.

En este modelo (véase Figura 7.13) se contemplan distintos tipos de CAs, cada uno con una funcionalidad bien definida. Son, vistas de arriba abajo, las siguientes:

- *PAA. Autoridad de aprobación de políticas (Policy Approving Authority).* Ocupa la raíz del árbol jerárquico. Certifica las PCAs definidas en el siguiente escalón y, como su nombre indica, sirve para autorizar las políticas definidas en ellas.
- *PCA. CA de creación de política (Policy Creation Authority).* Cada una de las cuales encabeza un subdominio de seguridad propio, con una política específica.
- *CA de segundo nivel.* Sirve para segmentar el ámbito abarcado por la PCA conforme a diversas secciones y departamentos administrativos. Puede, a su vez, desglosarse en tantos niveles jerárquicos como sea necesario.

En este modelo se contempla también la existencia de un conjunto de autoridades de registro, llamadas *ORA (Organizational Registration Authority)* que sirven de apoyo a las CAs a la hora de autenticar y dar de alta a los distintos usuarios finales.

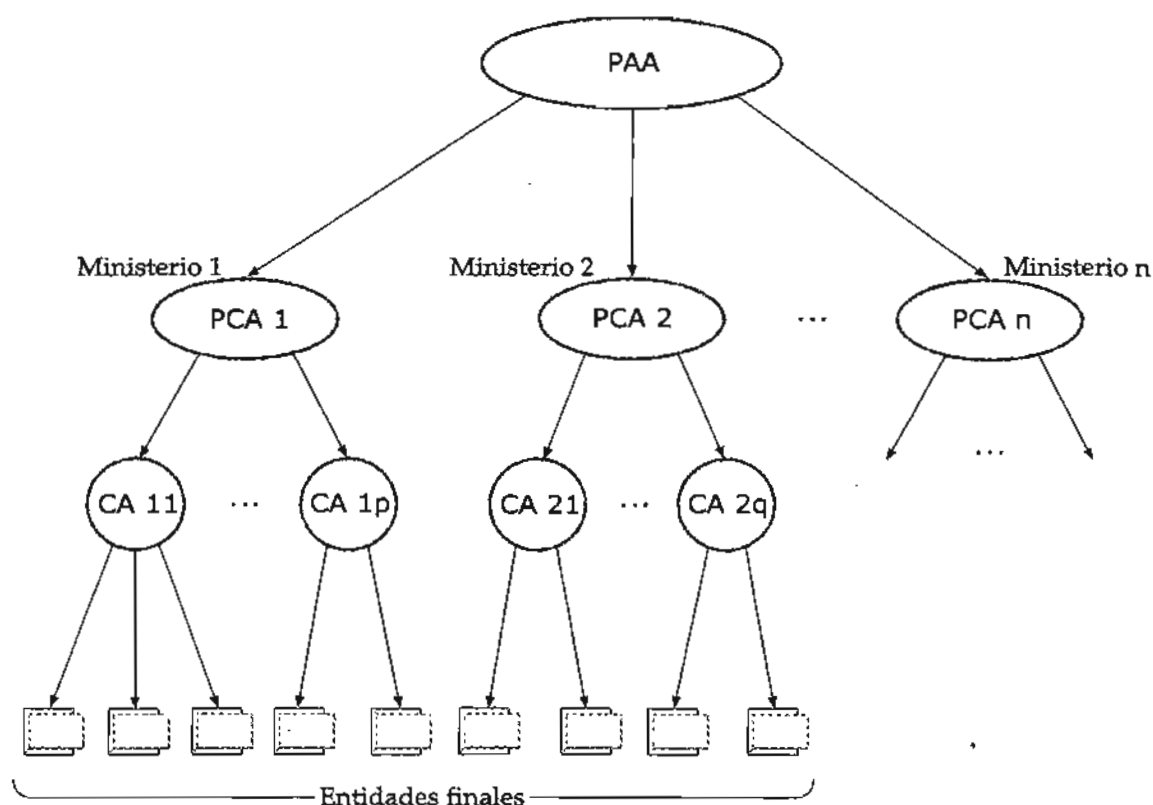


Figura 7.13. Organización de CAs en la Infraestructura MISSI.

SET: UNA INFRAESTRUCTURA DE PROPÓSITO ESPECÍFICO

Cuando se define un entorno de seguridad muy específico en el que los requisitos de certificación son muy particulares, y el valor de las transacciones que se llevan a cabo bajo su paraguas de protección es suficientemente elevado, puede resultar conveniente diseñar una PKI *ad hoc*. Este es el caso de la Infraestructura SET (*Secure Electronic Transaction*) [ViMa96], una iniciativa desarrollada inicialmente por VISA a la que más tarde se unió MasterCard, otra compañía también dedicada al negocio de las tarjetas de crédito. A modo de ejemplo, hemos recogido aquí un resumen de las principales características de este modelo que es, sin duda, la propuesta de infraestructura de clave pública más completa que se ha realizado hasta la fecha en el ámbito de los dominios de seguridad particulares y específicos.

El objetivo de esta infraestructura de seguridad es salvaguardar los pagos bancarios con tarjetas de crédito que se realicen a través de Internet y otras redes telemáticas. Las entidades de comunicación presentes en este escenario son:

- Titular de la tarjeta (*Cardholder*). Es el usuario final al que se le ha expedido una tarjeta de crédito. Puede tratarse de una tarjeta convencional, aunque está previsto la incorporación de tarjetas inteligentes que tengan capacidad de realizar firmas y operaciones de cifrado (tanto simétrico como asimétrico) sobre los datos con que opera.
- Institución emisora (*Issuer*) de la tarjeta (Visa, MasterCard u otra).
- Comerciante (*Merchant*). Es quien vende un producto o un servicio aceptando la tarjeta como medio de pago.
- Banco o entidad financiera (*Acquirer*), que se encarga de pagar al comerciante y de cobrar al comprador el importe de las transacciones realizadas.

- e) Pasarelas de pago (*Payment gateway*). Son sistemas soportados por las entidades financieras que sirven para que los vendedores de mercancías o servicios obtengan la autorización necesaria para llevar adelante una transacción.
- f) Autoridades de Certificación para garantizar la validez de las claves públicas de todas las entidades participantes en el escenario de comunicación.

En concreto, los servicios de seguridad que se pretende proveer son los de autenticación de las distintas entidades que intervienen en el proceso y los de confidencialidad e integridad de los datos transmitidos. Para ello utiliza técnicas de cifrado con algoritmos simétricos y asimétricos (DES y RSA), firma digital y certificación X.509.

En la Figura 7.14 se representa el esquema de organización de las distintas CAs presentes en la Infraestructura SET. En este modelo se contemplan distintos tipos de CAs, cada uno con una funcionalidad bien diferenciada. Éstos son (de abajo arriba):

- **CCA. CA de titular de tarjeta (Cardholder CA).** Estas CAs se encargan de la generación y distribución de certificados a las personas que poseen una tarjeta de crédito. En la especificación de SET se regulan los procedimientos mediante los cuales un usuario puede solicitar y recibir un certificado con suficientes garantías de autenticidad de ambas partes. Estas CAs no generan ni mantienen CRLs, pero pueden encargarse de la distribución de las CRLs generadas por otras CAs.
- **MCA. CA de comerciante (Merchant CA).** Son las encargadas de la generación de certificados para los comerciantes o vendedores. Para ello deben contar con

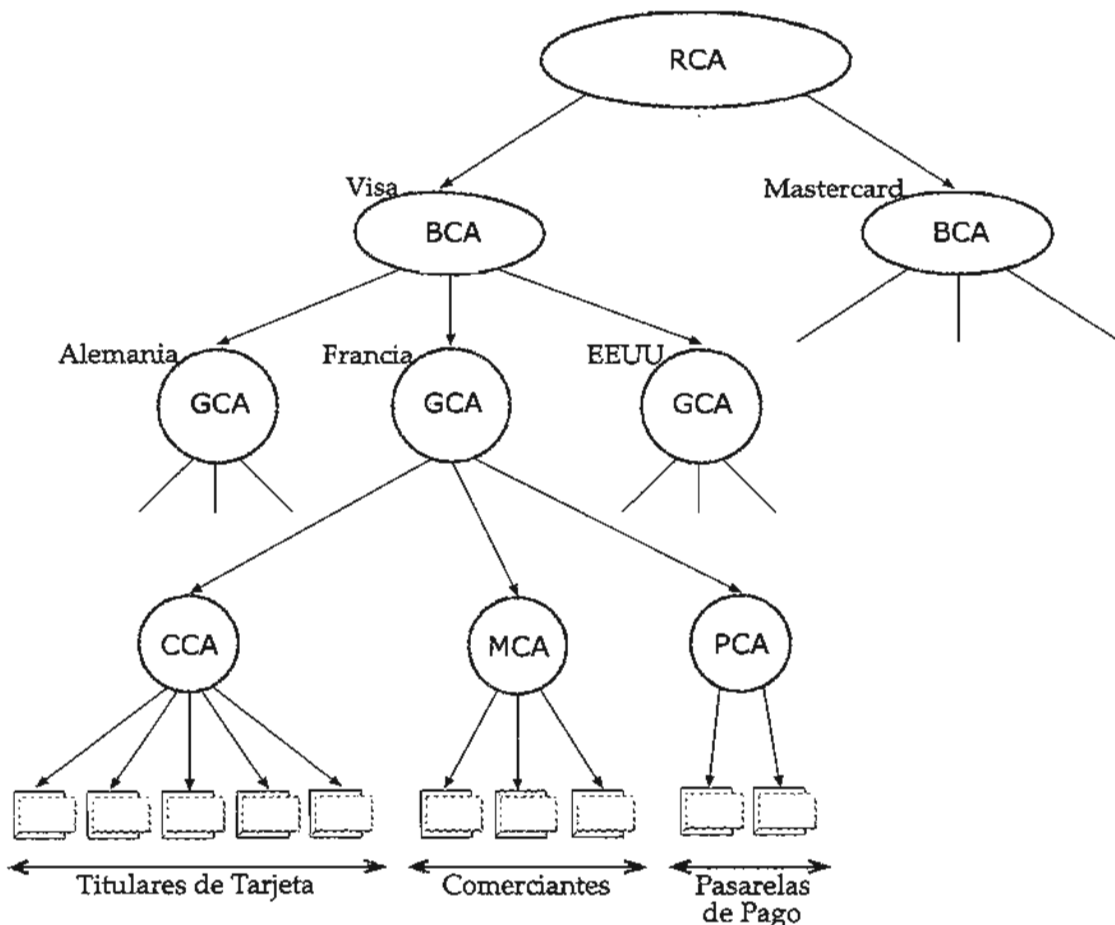


Figura 7.14. Organización de CAs en la Infraestructura SET.

la debida aprobación por parte de la entidad financiera que ha de soportar las transacciones. Estas CAs tampoco generan ni mantienen CRLs.

- *PCA. CA de pasarela de pago (Acquirer Payment Gateway CA)*. Son las encargadas de la emisión de certificados a las pasarelas que sirven de intermediarias entre los vendedores y los bancos o entidades financieras para la autorización y registro de las transacciones realizadas. Estas CAs sí son responsables de la generación, distribución y mantenimiento de las CRLs de certificados emitidos a favor de pasarelas de pago.
- *GCA. CA geopolítica (Geo-Political CA)*. Es un tipo de CA que puede aparecer opcionalmente para adaptar la política global de la marca de tarjeta (Visa, MasterCard, etc.) a las particularidades concretas que puedan darse en una determinada zona geográfica o en ámbitos donde la política financiera allí vigente imponga determinadas restricciones o condicionantes en cuanto a las responsabilidades que se adquieren usando o aceptando una tarjeta de crédito como medio de pago. (Su inclusión representa una innovación interesante por parte de SET, ya que establece un mecanismo para recoger el hecho cierto de que las políticas que afectan a las comunicaciones seguras no son coincidentes en todos los países.)
- *BCA. CA de marca de tarjeta (Brand CA)*. Cada marca de tarjeta está facultada para definir una política concreta y diferenciada. Estas CAs sirven para definir esa política y para garantizar la validez de todo el árbol de certificación de ellas dependiente. Gestionan también las CRLs de sus certificados.
- *RCA. CA raíz (Root CA)*. Ocupa el más alto nivel jerárquico de la estructura y sólo se encarga de la emisión y registro de las (pocas) BCAs que representan a las distintas marcas de tarjeta presentes en la arquitectura.

En SET se utilizan CRLs de versión 2 y certificados X.509 v3 en los que se han definido nuevas extensiones propias para identificar la marca y el contexto en el que será utilizado el certificado. Para simplificar la infraestructura de certificación no se ha incluido ningún tipo de repositorio para el almacenamiento y recuperación de certificados y CRLs. Esto conlleva que todos los certificados presentes en el camino de certificación que son necesarios para la validación de las claves públicas tienen que ser incluidos en la transferencia de datos, lo que representa un aumento importante del tráfico generado (aunque trate de aliviarse esta situación mediante la generación de *huellas* de los certificados, *thumbprints*, de tamaño más reducido, que sustituyen a éstos en las estructuras de datos que se transmiten).

El esquema propuesto es muy complejo e intervienen en la cadena de certificación múltiples y variadas autoridades, por lo que su implantación global está siendo más lenta de lo que en un principio se creía. Además, por decisión de diseño, esta infraestructura *sólo* sirve para la realización de transacciones con tarjetas, situación claramente reflejada en la política imperante en esta arquitectura, de forma que está expresamente desautorizado usar las claves y los certificados en otras comunicaciones seguras que pudiesen interesar a las entidades participantes en el dominio. Por ello, aunque la Infraestructura SET esté operativa, cuando se deseen implantar entornos telemáticos en los que se garantice la seguridad para otras aplicaciones será necesario disponer de otra infraestructura de certificación complementaria.

7.6. HACIA MODELOS FLEXIBLES Y ADAPTADOS A LAS NECESIDADES DE CADA GRUPO DE USUARIOS

Aunque la Infraestructura PEM se pensó en un principio para cubrir las necesidades de certificación del correo electrónico seguro, la propuesta final aspiraba a servir para dar soporte a todas las aplicaciones telemáticas que se desarrollasen en el entorno de Internet. Es, por tanto, una arquitectura que pretende ser abierta y genérica. Las limitaciones que presenta han sido ya discutidas en apartados anteriores.

Por otra parte, los modelos estudiados en el epígrafe anterior están decididamente orientados a dominios de seguridad en los que el colectivo de entidades comunicantes que necesitan ser certificadas está claramente definido y acotado. No son, en principio, soluciones que puedan servir para cualquier grupo de usuarios con organización y problemática diferenciada.

Por todo ello, resulta necesario buscar soluciones que permitan establecer infraestructuras de certificación de configuración flexible y adaptada a las necesidades de cada grupo de usuarios. O lo que es lo mismo: una vez definido un dominio de seguridad específico, es necesario encontrar una infraestructura que se adapte a las peculiaridades de las entidades comunicantes que lo componen, al «tamaño» y topología del entorno de comunicación en el que se desenvuelve y a la política de seguridad en él definida.

La solución encontrada puede establecerse de dos formas distintas:

- a) Apoyándose en otras infraestructuras previamente implementadas (a las que las entidades del dominio tengan acceso).
- b) Si ello no es posible, definiendo una nueva configuración.

En la búsqueda de estas soluciones es donde se enmarca el análisis de las insuficiencias de las versiones iniciales del Certificado X.509 y, consecuentemente, el estudio de nuevos campos que condujo a la definición de la versión 3. Otro tanto cabe decir de la nueva concepción de las CRLs que dio como resultado la definición de la versión 2. Ya se comentó el papel destacado que en esa tarea de perfeccionamiento y definición del Certificado X.509 v3 ha tenido, además de ISO e ITU-T, el grupo de trabajo Internet PKIX (*Public Key Infrastructure*).

En los párrafos que siguen se pretende presentar una panorámica de los principales elementos que sirven para la configuración de este tipo de modelos de certificación flexibles y adaptados a que venimos haciendo referencia. No se tratará, por tanto, de proponer un modelo completo y acabado cuanto de destacar los distintos mecanismos que permitan, en cada caso, su diseño y configuración. Para ello se utilizarán conceptos presentes en la Arquitectura PKIX y en otras propuestas realizadas con similares objetivos [LoCa97], siempre bajo el protagonismo relevante de la información que se puede aportar mediante el campo *extensions* de la versión 3 del Certificado X.509.

ELEMENTOS DE DISEÑO

Para definir adecuadamente una infraestructura que se adapte a las necesidades reales del conjunto de entidades que constituyen un determinado dominio de seguridad es necesario contar con una serie de condicionantes entre los que cabe destacar:

- *El número de CAs debe estar en consonancia con la complejidad del dominio*

Es evidente que cuando el número de entidades presentes es muy elevado y además el dominio es complejo desde un punto de vista geopolítico y administrativo, será necesario el concurso de muchas CAs de diverso tipo. Sin embargo, cuando no se dan estas circunstancias, el grupo de usuarios no tendrá por qué soportar la carga que representa establecer los procedimientos de certificación sobre una infraestructura compleja.

Recuérdese el ejemplo que se expuso al analizar los inconvenientes del Modelo PEM, consistente en imaginar un proyecto de investigación constituido por varios grupos de trabajo, cada uno presente en una Universidad distinta de un país distinto, cada uno de los cuales podría tener su parcela local de certificación adaptada a sus necesidades. No obstante, llegado el momento en el que dos entidades finales (pertenecientes a grupos distintos) necesiten establecer un camino de confianza se verán obligadas a incluir multitud de CAs, teniendo que llegar posiblemente hasta la misma IPRA, ascendiendo y bajando a través del árbol definido en ese modelo.

En consecuencia, dado un dominio, el número de CAs que aparezcan en la «foto» de la infraestructura de certificación bajo la que opera debe ser rigurosamente proporcional a la complejidad que allí se presenta. Por tanto, las arquitecturas resultantes podrán ser muy sencillas, con una única CA certificando a todos los usuarios, o muy complejas, con muchas CAs de muchos tipos organizadas jerárquicamente en un árbol de múltiples ramas.

- *La política y las subpolíticas deben estar asociadas a determinadas CAs*

Como ya se recoge en PEM (las denominadas PCAs) y en las restantes propuestas de infraestructuras jerarquizadas formuladas posteriormente, en la parte alta de la jerarquía deberán existir CAs en las que se defina la política imperante en todo el trozo de árbol de certificación desplegado a partir de ellas. Caso de ser necesario, cada una de esas políticas se podrá dividir en subpolíticas mediante el procedimiento de implantar otras CAs de definición de política en el nivel jerárquico inmediatamente inferior a aquel. En la Figura 7.15, que más adelante se comenta, se recoge una situación de este tipo.

Además de esas CAs de definición de política, podrán existir otras CAs, jerárquicamente dependientes de ellas, cuya única finalidad sea la de generar certificados manteniendo la política definida en la rama del árbol en que se encuentren.

- *El punto común de confianza está marcado por una política común*

Cuando dos entidades finales pretenden establecer una comunicación segura, la cadena de certificados necesarios para validar una clave pública está constituida por todos los certificados que van desde el certificado de la entidad final hasta un punto común en el árbol de certificación. Por *punto común* deberá entenderse no sólo que exista, dentro de la topología de la infraestructura, una CA común entre ellos, sino que los certificados emitidos por la CA para ambas ramas del *camino de certificación* estén generados bajo la misma política (o equivalente). Existen extensiones del certificado que permiten identificar las políticas bajo las que se emite un certificado y otras para la determinación de políticas compatibles. También es viable la definición de nuevas extensiones para facilitar el conocimiento del camino de certificación.

- *El árbol de certificación debe ofrecer flexibilidad para el establecimiento de caminos de certificación*

Dentro de lo posible deberán establecerse relaciones que faciliten la existencia de caminos de certificación mediante los cuales las entidades finales puedan construir una relación de confianza en sus respectivas claves públicas. Esto se puede conseguir:

- *Estableciendo reconocimiento de políticas*

Se puede implantar la correspondencia de políticas mediante certificados cruzados (o en un solo sentido) entre CAs responsables de la definición de políticas para conseguir un determinado grado de correspondencia entre las políticas definidas por cada una de ellas.

- *Posibilitando que una CA «organizacional» sea certificada bajo más de una política*

Una solución sencilla para conseguir esto sin merma de la consistencia del sistema es que una de las CAs que emiten certificados directamente a las entidades finales pueda estar certificada, a su vez, por distintas CAs de definición de política. Cuando esa CA emita certificados para los usuarios finales podrá elegir bajo qué política de certificación los genera. En la Figura 7.16 que se comenta en el siguiente apartado se describe este caso.

- *Posibilitando que un usuario pueda operar bajo más de una política*

Esto se puede conseguir mediante el mecanismo descrito en el párrafo anterior o permitiendo que un usuario final sea certificado por más de una CA pertenecientes a árboles generados bajo distintas políticas (Figura 7.16).

- *El crecimiento del árbol de certificación debe estar sometido a restricciones*

Mediante el uso de determinadas extensiones de la v3 del Certificado X.509 se pueden establecer limitaciones tanto al crecimiento del árbol como a los nombres de entidades finales a las que una CA tiene permitido emitir certificados. Con ello se preserva la disminución de confianza que se deriva de caminos demasiado largos y se facilita la adecuada configuración del dominio de seguridad.

Veamos a continuación de forma más pormenorizada algunos de los aspectos recogidos en la descripción precedente, analizando con cierto detalle la estructura de alguna de las extensiones citadas y el uso que de ellas puede hacerse.

DETERMINACIÓN DE LAS POLÍTICAS PERMITIDAS

Como antes apuntábamos, para el diseño de infraestructuras consistentes es muy conveniente que las políticas y subpolíticas de certificación estén asociadas a unas CAs especializadas que ocupen los niveles más altos de la jerarquía. En la Figura 7.15 se recoge la arquitectura de la infraestructura de un dominio en la que existe en la raíz del árbol de certificación una *CA de definición de política*, que aprovechando la nomenclatura de PEM, y a los solos efectos de facilitar su representación gráfica, denominaremos aquí con el acrónimo *PCA*. (En cada modelo que se diseñe se utilizará la nomenclatura que mejor cuadre con sus particularidades.)

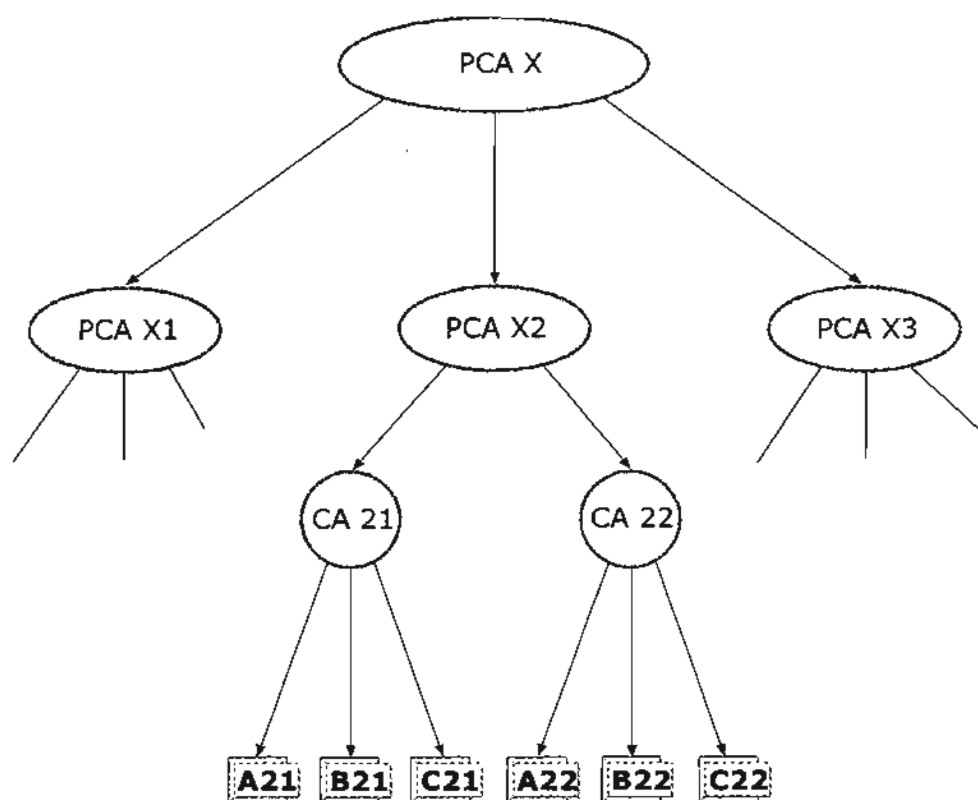


Figura 7.15. Infraestructura con CAs de definición de políticas y subpolíticas.

En el ejemplo de la figura, la política general del dominio se ha dividido en tres subpolíticas mediante la inserción de otras tres PCAs en el segundo nivel jerárquico. En este caso, para garantizar la consistencia del sistema, cada subpolítica deberá consistir en un conjunto de restricciones sobre la política definida en el nivel inmediatamente superior (este es el caso de las GCAs en relación con las BCAs en el modelo SET). En caso de estructuras muy complejas este mismo proceso de subdivisión se puede reproducir varias veces.

Si la complejidad del dominio no fuese muy grande, esas mismas PCAs podrían emitir directamente certificados a las entidades finales. En el ejemplo se ha supuesto, no obstante, que el número de entidades finales que le correspondería certificar a cada una de las PCAs de subpolítica sería muy grande, por lo que se ha considerado conveniente incorporar otras CAs, jerárquicamente dependientes de ellas, cuya única finalidad es la de descargar de trabajo a las PCAs, generando certificados que mantienen la política definida en la rama del árbol en que se encuentren. Esta situación es muy similar a lo que ocurre con las *CAs Organizacionales* de PEM, por lo que, para facilitar las descripciones que siguen, adoptaremos aquí esta misma nomenclatura.

La política de certificación, *certificate policy*, se deberá especificar en un documento identificado mediante un OBJECT IDENTIFIER en el que se recojan las reglas que indiquen la aplicabilidad y las restricciones de los certificados generados bajo esa política. Deberá existir también una Declaración de Prácticas de Certificación o CPS (*Certification Practice Statement*) que será publicada por el proveedor del servicio, en la que se recojan otros detalles acerca de las condiciones en que los certificados son generados y pueden ser utilizados. Aunque a este nivel de detalle dé la impresión de que se trata de lo mismo, son dos cosas diferentes (aunque con elementos comunes). En el Capítulo 10, especialmente dedicado a la política de seguridad, se aclaran esas diferencias.

En el entorno telemático cuya infraestructura de certificación (Figura 7.15) se ha tomado como ejemplo, la política por la que se rige la PCA X deberá ser registrada de forma que pueda ser conocida por todos los usuarios (pertenecientes o no a este entorno) que vayan a hacer uso de los certificados emitidos por CAs que operen bajo esa política. Igualmente, las restricciones particulares que PCA X1, PCA X2 y PCA X3 introduzcan en esa política también deberán estar reflejadas en documentos públicos de fácil acceso para esos usuarios. La forma más adecuada de identificar estos documentos es mediante el uso de un OBJECT IDENTIFIER, tal y como este concepto está recogido en ASN.1.

El Certificado X.509 v3 dispone de una *extensión* que permite especificar la política o políticas bajo las que se puede usar un certificado. Se trata de la extensión *certificatePolicies*, parte de cuya especificación ASN.1 es la siguiente:

```
certificatePolicies EXTENSION ::= {
    SYNTAX          CertificatePoliciesSyntax
    IDENTIFIED BY   id-ce-certificatePolicies }

CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
    policyIdentifier      CertPolicyId,
    policyQualifiers      SEQUENCE SIZE (1..MAX) OF
                          PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER
```

Como puede observarse, permite identificar más de una política bajo la cual el certificado puede utilizarse, aunque más adelante daremos argumentos para recomendar que no se use esta facilidad y sólo sea una la política identificada. Cada una de las políticas reconocidas queda determinada de forma no ambigua mediante un identificador de objeto, a través del cual el hipotético usuario del certificado podrá acceder a las reglas que marcan esa política. Para simplificar, en el segmento de especificación de la extensión que hemos reproducido se ha omitido la definición del campo *PolicyQualifierInfo*, que más adelante comentaremos.

Además, en esta extensión, el campo que marca la criticidad tiene un significado muy importante. En efecto, si la extensión es no-crítica ello significa que todas las políticas especificadas en el campo *PolicyInformation* son aceptadas como válidas para el uso del certificado, pero deja abierta la puerta para que el certificado pueda usarse bajo otras políticas distintas. Por contra, si la extensión es calificada como crítica, sólo está permitido utilizar el certificado bajo una de las políticas identificadas en la extensión.

En el apartado anterior, uno de los elementos de diseño que destacamos como condicionantes para la adecuada definición de una infraestructura adaptada a las necesidades del entorno fue que la arquitectura permitiese que tanto una CA organizacional como una entidad final pudiesen estar certificadas bajo varias políticas. En la Figura 7.16 se representa una infraestructura donde se produce ese esquema de certificación. En este ejemplo se representan tres árboles de certificación independientes cada uno de ellos operando bajo una política propia.

En estas condiciones, la entidad final A21 posee dos certificados, uno de ellos emitido por la CA 13 y otro por la CA 21, cada una de ellas encuadradas en una

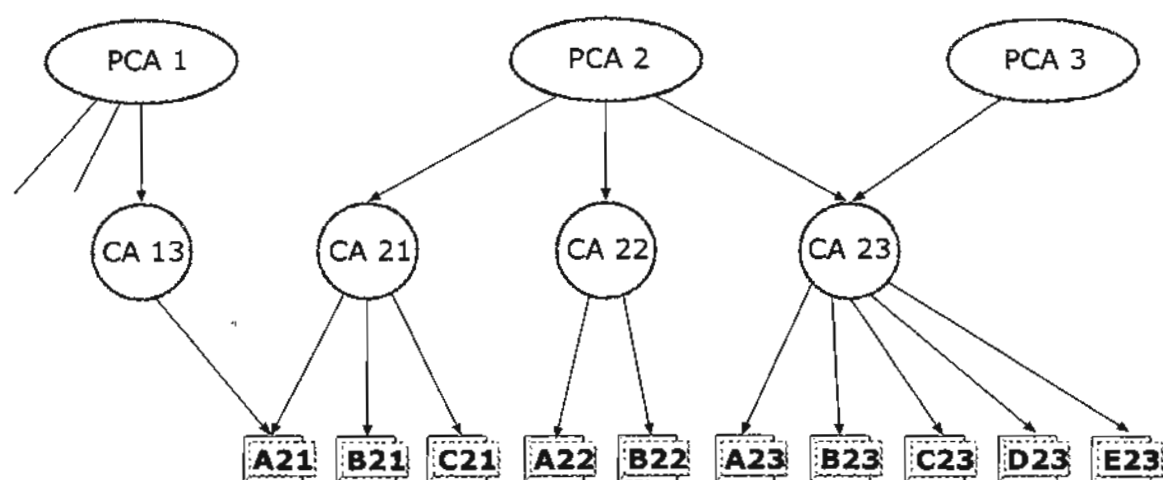


Figura 7.16. Usuarios y CAs operando bajo varias políticas.

política distinta. De igual modo, conforme a la ordenación reflejada en la figura, la CA 23 también recibe dos certificados, procedentes de PCA2 y PCA3 respectivamente, lo que posibilita que todas las entidades finales certificadas por esa CA puedan trabajar bajo cualquiera de las dos políticas definidas por esas PCAs.

El ejemplo propuesto sería todavía más plural en cuanto a políticas se refiere si considerásemos que cualquiera de las PCAs pudiese operar bajo más de una política. Si la PCA2 de la Figura 7.16 representase a una determinada universidad (en la que las CAs organizacionales podrían representar a distintos centros y departamentos) podrían haberse definido dos políticas distintas: una, por ejemplo, para protección del acceso a información interna reservada y para el intercambio de mensajes de correo seguros, y una segunda política (más crítica) para las cuestiones administrativas relacionadas con la gestión de calificaciones y la expedición de certificados académicos. La PCA3 podría representar a un centro de investigación que quisiese que el departamento representado por CA 23, en su conjunto, pudiese emitir certificados para operar también bajo algunas de las políticas allí definidas. Por último, la PCA1 podría ser la que definiese la política de una empresa concreta, una de cuyas CAs organizacionales, la CA 13, hubiese emitido un certificado a favor de A21, un determinado miembro de la universidad con el que mantuviesen comunicaciones seguras frecuentes.

Es importante señalar que si se considera conveniente que una PCA concreta tenga registrada más de una política, puede usar para cada una de ellas un distinto par de claves (como exige PEM), aunque es igualmente posible usar el mismo par de claves para todas ellas con la sola particularidad de generar certificados con número de serie distintos y con indicación explícita de la política bajo la que han sido generados. Por motivos prácticos suele ser recomendable usar la primera opción cuando una misma CA sostiene varias políticas con diferentes exigencias de seguridad y de responsabilidad en el uso de los certificados que genera: para las que conllevan mayor compromiso se exigen mayores salvaguardas en la custodia de la clave privada con la que han sido firmados y mayores exigencias para el sistema con que se firman.

Sea como fuere, con planteamientos similares a los antes expuestos es posible diseñar infraestructuras en las que diversas entidades puedan operar bajo múltiples políticas. Una forma de dejar claro bajo qué política ha sido generado un certificado es incluyendo en él la extensión *certificatePolicies*. Si se acepta la recomendación de que por cada política sea generado un certificado distinto, la determinación de esa política estará recogida en la cadena de enteros que constituyen el identificador de

objeto que aparezca como valor del único *policyIdentifier* presente en el campo *PolicyInformation*.

Si esto es así, y el usuario A23 del ejemplo antes descrito está certificado para trabajar con dos políticas distintas (una de ellas dependiente de PCA2 y la otra de PCA3), tendría dos certificados distintos, cada uno con su correspondiente *policyIdentifier* (ambos emitidos por la CA 23). Uno de estos dos certificados formaría parte del camino de certificación necesario para encontrar un punto común de confianza con otra entidad, por ejemplo A21, que operase bajo la misma política definida en PCA2, y el otro certificado formaría parte del camino de certificación que se estableciese para comunicarse con otra entidad certificada bajo la política definida en PCA3. Es decir, el que dos caminos distintos estén constituidos por certificados diferenciados en función de la política bajo la que operan refuerza la ligazón entre camino de certificación y política y simplifica la «navegación» por los árboles de certificación.

En el apartado *Múltiples CAs organizadas jerárquicamente* del epígrafe 7.2 se apuntaron algunos procedimientos útiles para la localización del camino de certificación. La siempre delicada y costosa tarea de encontrar un camino de certificación que conduzca a un punto común de confianza en estructuras complejas y con muchos niveles jerárquicos se verá facilitada en gran medida si la búsqueda se realiza de forma que todos los certificados que constituyan el camino tengan un solo *policyIdentifier* marcado con el identificador de objeto de la política bajo la que se opera.

(Aun manteniendo la conveniencia de seguir la recomendación de generar un certificado distinto por cada política, cabe decir que una solución contundente para resolver el problema de la localización del camino de certificación consiste en definir una nueva extensión al Certificado X.509 especialmente dedicada a contener una secuencia de identificadores de CAs y de certificados que permitan obtener fácilmente el punto común de confianza⁹.)

En la especificación que más arriba se recoge de la extensión *certificatePolicies* se refleja que cada *policyIdentifier* va acompañado, opcionalmente, por un campo *policyQualifiers*, constituido a su vez por una secuencia de *PolicyQualifierInfo*. Estos calificadores de la política pueden ser usados por los diseñadores de una determinada infraestructura para incluir información que ayude a las aplicaciones seguras que utilizan el certificado a tomar decisiones acerca de lo adecuado o no de aceptarlo como válido en el dominio en que operen.

Ya vimos cómo, de forma totalmente elemental, la aplicación que gobierna el PGP está programada para tomar determinadas decisiones ante la recepción de una clave pública certificada. De igual forma, en dominios mucho más complejos en los que operen distintas aplicaciones telemáticas, los sistemas deberían estar programados para saber interpretar automáticamente determinada información contenida en el certificado, principalmente en lo relativo a la política o políticas válidas, de cara a la aceptación o no de ese certificado. Lo que aporta el campo *policyIdentifier* no es más que una ristra de enteros que identifican a la política, sin más matizaciones. Mediante estos *policyQualifiers* se puede añadir, por ejemplo, una URL que sirva como puntero de un sitio donde esté el documento definitorio de la política. Si además se avanza en la dirección (absolutamente necesaria) de especificar formalmente las reglas que definen una política, esa reglas podrán ser entendidas automáticamente por el programa que gobierna la aplicación telemática de que se trate.

A modo de cierre de estos comentarios sobre la determinación de la política, es menester aclarar que todos estos refinamientos acerca de la inclusión en el certificado de referencias a la política tienen sentido en infraestructuras que presenten un cierto nivel de complejidad en las que esa información sirva para el establecimiento de las relaciones

de confianza. En estructuras sencillas podemos prescindir de todas ellas sin que se produzcan mermas en la consistencia del sistema. Sea, por ejemplo, un dominio de seguridad circunscrito a una empresa u organización con un número no muy elevado de usuarios en la que están definidas dos políticas: una más crítica reservada para transacciones en las que los bienes puestos en juego (financieros o de otro tipo) sean elevados, y otra más general para ser utilizada en aplicaciones que exijan un menor nivel de seguridad.

Sin abordar por ahora la problemática derivada de la comunicación con otros entornos ajenos a la empresa, para resolver los problemas de comunicación aquí planteados podría disponer de dos PCAs distintas, una para cada política, con medidas distintas para la protección de su clave privada y para la generación de firmas. Cada PCA distribuiría sus certificados (sin indicación alguna sobre la identidad de la política) entre los empleados que previamente hayan sido autorizados para trabajar en aplicaciones que sigan una u otra política. En este caso, las reglas que rigen cada política son perfectamente conocidas de antemano y las aplicaciones estarán previamente programadas para que la mera utilización de uno u otro certificado (si es que lo posee) determine el tipo de tareas que pueden llevar a cabo y los límites que rigen las distintas transacciones que puedan realizar.

RESTRICCIONES A LA POLÍTICA Y CORRESPONDENCIA ENTRE POLÍTICAS

Como más arriba apuntábamos cuando seleccionamos los principales elementos de diseño que deben presidir la adecuada configuración de una infraestructura, el árbol de certificación debe ofrecer flexibilidad para definir caminos que permitan establecer correspondencias entre políticas. Veamos un caso en el que esto resulta conveniente y los mecanismos de que disponemos para implantarlo.

Continuemos con el último ejemplo planteado en el apartado anterior. En la Figura 7.17, la PCA X1 representa a la autoridad de certificación que tiene definida la política más general y menos exigente, mientras que la PCA X2 es la responsable de emitir certificados bajo la política más crítica, reservada, entre otras cosas, para realizar transacciones financieras cuyo importe supere cierto nivel. Podemos suponer que para la gestión de esas transacciones, los empleados que trabajan bajo la política definida en PCA X2 necesitan comunicarse con otras entidades comunicantes situadas en el dominio de seguridad de un banco o entidad financiera, cuya política vamos a suponer que está definida en la PCA Y.

Para que la entidad E pueda confiar en la clave pública de A', y viceversa, es necesario que exista un certificado cruzado entre PCA X2 y PCA Y. No obstante, si

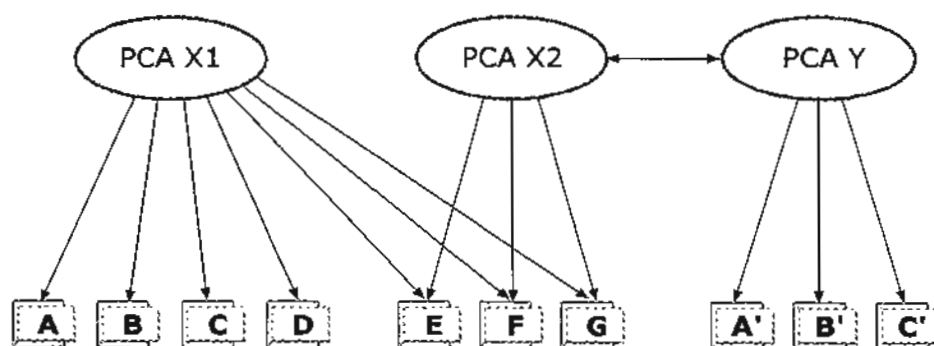


Figura 7.17. Correspondencia entre políticas.

como hemos venido postulando, para establecer una relación de confianza es necesario no sólo el reconocimiento de un punto común de certificación sino una concordancia en las políticas bajo las que los certificados han sido generados, una buena solución es añadir en ese certificado cruzado una información que indique que las políticas imperantes en ambas CAs son equivalentes, por lo que deben ser admitidas como válidas en ambos extremos. La forma más adecuada de conseguir esto es incluyendo en ese certificado la extensión *policyMappings*, cuya especificación ASN.1 (ya presentada a modo de ejemplo en el epígrafe 6.6) es la siguiente:

```

policyMappings EXTENSION ::= {
    SYNTAX                                PolicyMappingsSyntax
    IDENTIFIED BY                        id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
    issuerDomainPolicy                  CertPolicyId,
    subjectDomainPolicy                CertPolicyId }

CertPolicyId ::= OBJECT IDENTIFIER

```

Como puede observarse, la extensión sirve para establecer correspondencia entre varios pares de políticas, aunque en el caso del ejemplo sólo se rellena con un par. El tipo *CertPolicyId* es el mismo que aparece en la extensión *certificatePolicy* para denotar el campo *policyIdentifer* al que nos hemos venido refiriendo reiteradamente en el apartado anterior. No es obligatorio que esta extensión vaya solamente en certificados cruzados. De hecho, la correspondencia de políticas puede establecerse en certificados de un solo sentido.

Lo primero que hay que señalar es que conforme con las recomendaciones de diseño que venimos apuntando en estos apartados, este tipo de extensiones deben ser usadas solamente en certificados que relacionen a CAs responsables de la definición de políticas (las que hemos venido nombrando como PCAs). Usarlas en otras relaciones puede acarrear situaciones de inconsistencia.

En realidad, el establecimiento de correspondencia entre políticas sirve para flexibilizar la estructura del árbol de certificación ofreciendo «atajos» que simplifiquen y acorten los caminos de certificación. De no existir esta alternativa, en el ejemplo de la figura habría sido necesario que existiese otra CA jerárquicamente «por encima» de las PCAs que presiden cada uno de los dominios, con el consiguiente alargamiento de los caminos de certificación.

El uso de estos atajos en arquitecturas jerárquicas complejas con múltiples niveles de CAs puede ser peligroso de cara a la consistencia general del sistema, por lo que deben ser usados con mucha precaución. Resultan, sin embargo, muy útiles para diseñar arquitecturas con CAs de igual jerarquía, no excesivamente complejas. De hecho, el ejemplo mostrado en la Figura 7.17 representa una organización de este tipo.

La correspondencia que ofrece esta extensión abarca a la totalidad de las políticas, no permitiendo restricciones de ningún tipo. El establecimiento de restricciones está también en la base de la posible existencia de PCAs que definan subpolíticas tal y como se recoge en el ejemplo representado en la Figura 7.15.

En estos casos, para restringir la política sólo disponemos de las reglas que constituyen su definición, ya que la extensión *policyConstraints*, en contra de lo que de su propio nombre cabría esperar, no nos sirve de ayuda para este cometido, ya que solamente sirve para prohibir correspondencias (*mappings*) y para exigir que aparezcan explícitamente indicadores de políticas en los siguientes certificados de la cadena.

Las reglas que definan la subpolítica deberán ser un subconjunto de aquellas que se han utilizado para estipular las condiciones que rigen la política de la que procede. A medida que se vaya avanzando en la especificación formal de las reglas que constituyen la política, esas restricciones será también posible expresarlas de forma simplificada, explícita, no ambigua y automáticamente inteligible por las entidades que han de manejar esos certificados.

Una extensión que sí puede ser de utilidad para restringir en alguna medida la política a partir de un determinado escalón de la jerarquía es la que permite restringir el uso de la clave certificada. Se trata de la extensión *Uso de la clave* (*keyUsage*) y su definición ASN.1 es:

```
keyUsage EXTENSION ::= {
    SYNTAX
    IDENTIFIED BY
    KeyUsage
    id-ce-keyUsage }

KeyUsage ::= BIT STRING {
    digitalSignature      (0),
    nonRepudiation       (1),
    keyEncipherment      (2),
    dataEncipherment     (3),
    keyAgreement         (4),
    keyCertSign          (5),
    cRLSign              (6)
}
```

Si la extensión se marca como crítica, el certificado generado (y, consecuentemente, los que se generen a partir de ese escalón) sólo podrán ser usados en las tareas cuyo correspondiente bit esté marcado con valor uno. Queda, por tanto, expresamente prohibido utilizar esos certificados en todas aquellas tareas marcadas con cero, lo que conlleva una cierta restricción sobre el uso que estuviese permitido en el escalón jerárquico precedente.

RESTRICCIONES A LA POLÍTICA MEDIANTE RESTRICCIONES EN LOS NOMBRES

Que «el crecimiento del árbol de certificación debe estar sometido a restricciones» es la última de las consideraciones que seleccionamos como convenientes cuando señalamos los elementos de diseño que son de utilidad para definir adecuadamente una infraestructura que se adapte a las necesidades reales del grupo de usuarios que constituyen el dominio. Con ello, decíamos, se preserva la disminución de confianza que se deriva de caminos demasiado largos y se facilita la adecuada configuración del dominio de seguridad.

Una extensión del Certificado X.509, bastante sencilla en su formulación, que sirve para delimitar el tamaño del árbol de certificación es la que impone *Restricciones básicas* (*basicConstraints*). Su especificación es:

```
basicConstraints EXTENSION ::= {
    SYNTAX
    IDENTIFIED BY
    BasicConstraintsSyntax
    id-ce-basicConstraints }
```

```

BasicConstraintsSyntax ::= SEQUENCE {
    ca BOOLEAN DEFAULT FALSE,
    pathLenConstraint INTEGER (0..MAX) OPTIONAL
}
    
```

En realidad, la acotación del árbol supone el establecimiento de limitaciones en la política, por lo que si se siguen las recomendaciones de diseño que estamos proponiendo, esta extensión sólo debería aparecer en los certificados generados por PCAs. Es decir, el campo *ca* debe aparecer siempre con valor *TRUE*.

Por esa misma ligazón con la política, esta información bien pudiese estar contenida en la definición de la política que se encuentra recogida en el documento correspondiente. No obstante, puede resultar interesante utilizarlo en determinadas PCAs de definición de subpolíticas para limitar de forma específica el tamaño de ciertos subárboles de la estructura.

Mucho mayor interés y potencia tiene otra extensión del certificado. Se trata de la extensión *Restricciones sobre el nombre* (*nameConstraints*), cuya formulación es la siguiente:

```

nameConstraints EXTENSION ::= {
    SYNTAX NameConstraintsSyntax
    IDENTIFIED BY id-ce-nameConstraints }

NameConstraintsSyntax ::= SEQUENCE {
    permittedSubtrees [0] GeneralSubtrees OPTIONAL,
    excludedSubtrees [1] GeneralSubtrees OPTIONAL }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
    base GeneralName,
    minimum [0] BaseDistance DEFAULT 0,
    maximum [1] BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)
    
```

Como puede deducirse de la definición de la extensión, es posible especificar un número tan elevado como se desee de subárboles permitidos y excluidos.

Para entender mejor las restricciones que pueden imponerse con su uso volvamos al ejemplo representado en la Figura 7.16, en el que hemos supuesto que la CA 23 emite certificados en el ámbito de un determinado departamento universitario y que, al estar también certificada por la PCA 3, que hemos supuesto que representa a un hipotético centro de investigación ajeno a esa universidad, puede emitir certificados para operar bajo alguna de las políticas definidas en ese centro. Es decir, tal y como está configurada la infraestructura, esta CA 23 puede generar certificados bajo las dos políticas definidas en PCA2 (una más rigurosa y otra más laxa) además de una tercera correspondiente a la PCA3 del centro de investigación externo.

Es razonable suponer que la CA 23 genera certificados para todos los miembros de un departamento universitario, tanto profesores (A23 y B23), como alumnos (C23 y D23) como personal de administración y servicios (E23). Además, todas estas entidades comunicantes estarán organizadas en una rama del árbol de Directorio, dentro del ámbito del departamento, en distintas unidades organizativas (OUs) conforme a la estructura interna que allí se considere adecuada.

Lo más probable sería que el centro externo estuviese interesado en que solamente los miembros de un determinado grupo de investigación de ese departamento (en el que puede haber tanto profesores como alumnos como personal auxiliar) estuviese autorizado para operar bajo su política, excluyendo de forma explícita a todos los restantes miembros del departamento con los que no tiene establecida ninguna colaboración.

Partiendo del conocimiento de la estructura del Directorio y utilizando las posibilidades que le ofrece la extensión *nameConstraints* para especificar qué subárboles están permitidos y cuáles están excluidos, en el certificado que la PCA3 extienda a la CA23, la PCA3 puede incluir estas restricciones para garantizar que sólo determinadas personas de ese departamento estén capacitadas para utilizar certificados expedidos bajo esa política. Además, para definir las restricciones es posible usar no solamente nombres X.500 sino, aprovechando las posibilidades que ofrece el *GeneralName*, establecer limitaciones en base a la dirección de correo de las entidades comunicantes. Si a eso se añade que pueden establecerse restricciones sobre el uso de la clave mediante la extensión *keyUsage* antes descrita, el centro de investigación externo podrá regular con bastante precisión qué tipo de comunicaciones seguras está interesado en mantener con miembros de ese departamento.

Una solución similar puede buscarse a la hora de definir los certificados que la PCA2 emita en favor de la CA 23, con objeto de restringir quiénes están capacitados para usar certificados emitidos bajo cada una de las dos políticas definidas en su ámbito.

Por último, cabe decir que en el caso de certificados cruzados las restricciones pueden ser asimétricas. Así, en el ejemplo que se describe en la Figura 7.17, la PCA X2 (de la empresa) puede reconocer todos los certificados que cuelguen de la PCA Y del banco, pero éste puede solamente reconocer a determinados miembros de la empresa y excluir explícitamente a todos los demás.

7.7. LA VALIDEZ DEL CERTIFICADO. CRLs Y SERVICIOS OCSP

Todo lo que hemos venido describiendo refuerza la idea de que el certificado es la pieza clave y el principal protagonista de las distintas configuraciones que puede adoptar una infraestructura de certificación. Por tanto, para confiar en una clave pública tendremos que ser capaces de localizar el certificado que la avala y toda la cadena de certificados que constituyen el camino de certificación hasta alcanzar un punto de confianza. Por si esto no fuera ya de por sí una tarea significativa, aparece, además, la duda o la sospecha acerca de si los certificados a los que hemos conseguido acceder son certificados válidos o, por contra, han dejado de tener vigencia. Complicada tarea.

Este desaguisado pasa por permitir que un certificado pueda ser revocado dentro del periodo de validez en que fue emitido. Pero la revocación es la única solución posible ante el temor o la evidencia de que la validez de la clave certificada ha sido comprometida. El hecho de que también se pueda producir una revocación por causas administrativas, por ejemplo que una determinada persona deje de pertenecer a alguna organización, no presenta mayor complicación y en la definición de la política se puede decidir cuáles son los momentos más adecuados para dejarlo reflejado.

Por todo ello, el mejor remedio para todo esto es definir una política de seguridad que reduzca al mínimo posible la probabilidad de que la seguridad de las claves privadas sea vulnerada. El uso de tarjetas inteligentes u otros dispositivos resistentes ante manipulación (*tamper-resistant*) unido a unas reglas de comportamiento rigurosas

pueden ayudar a ello. En cualquier caso, llegado el momento de usar una clave certificada para cifrar confidencialmente un mensaje o para verificar una firma digital, es necesario tener una confianza adecuada en la vigencia del certificado que se maneja.

En el anterior capítulo se estudió con cierto detenimiento la definición y estructura de las listas de certificados revocados, CRLs, así como la de las extensiones específicas con ellas relacionadas. En los párrafos siguientes abordaremos los problemas que presenta el acceso a ese tipo de información y las soluciones que se ofrecen.

REVOCACIÓN EN UN DOMINIO CON UNA SOLA CA

Pensemos en un caso sencillo en el que la misma CA que emitió el certificado en cuestión sea la que emite una CRL con todos «sus» certificados revocados. Conforme a la política definida en ese entorno, la CA actualizará periódicamente la CRL. Cuando, por ejemplo, la entidad A quiera enviar un mensaje cifrado a B, además de contar con el certificado $CA\langle\langle B \rangle\rangle$, tendrá que consultar la CRL para comprobar que ese certificado no se encuentra incluido en esa lista. Pero puede darse el caso (Figura 7.18) de que B haya solicitado a la CA la revocación de su certificado un tiempo después de que la CA emitiera la última CRL pero antes del instante en el que A está procediendo al cifrado de la información.

En esa situación la existencia de la CRL no evita que, llegado el caso, B no se haga responsable de la posible revelación de ese mensaje. Evidentemente, cuanto más frecuentemente actualice la CA esa CRL, añadiendo los certificados que han sido revocados desde la última emisión, tanto menor es el peligro de que lo que acabamos de describir suceda. Lo que pasa es que todo tiene un límite y un coste, y es por eso por lo que decimos que la confianza en la vigencia del certificado debe ser *la adecuada*, por entender que esa confianza debe ser proporcional al riesgo que representaría usar un certificado que hubiese sido revocado en ese periodo de sombra. Es decir, dependiendo de los niveles de seguridad que se hayan establecido, será necesario establecer distintos niveles de comprobación acerca de la continuidad en la vigencia del certificado.

La única forma de resolver el problema es recurriendo a la *revocación inmediata* del certificado en cuanto se reciba en la CA (o en la autoridad auxiliar que se decida) la petición de revocación de un certificado. Como hay que establecer las cautelas correspondientes al caso, tampoco es dado decir que esta inmediatez pueda plasmarse a la velocidad de la luz. El problema a partir de ese momento es cómo hacer saber a las entidades usuarias de los certificados de esa CA que el certificado en cuestión ha sido revocado. Para simplificar, podríamos considerar agrupadas en dos las soluciones posibles a este problema (ambas, a nuestro juicio, de problemática implantación).

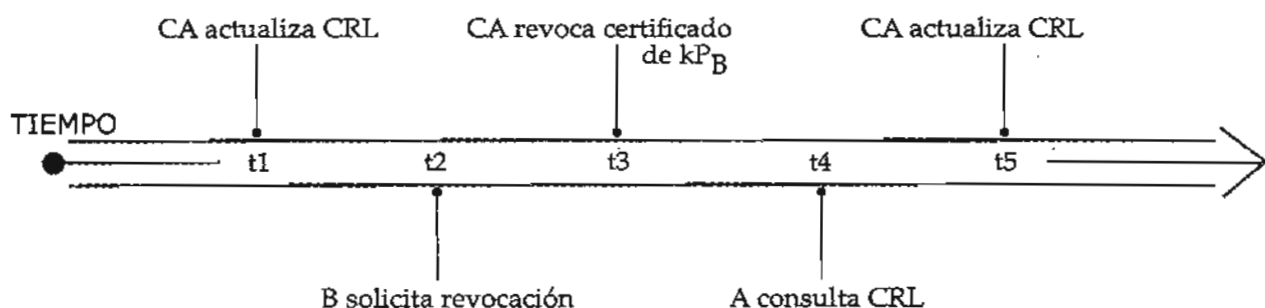


Figura 7.18. Periodo de sombra en la actualización de una CRL.

La primera de ellas consiste en difundir desde la CA ese certificado revocado a todas las entidades que hipotéticamente puedan usarlo. Para que ello fuese eficaz ese envío debería contar con los necesarios servicios de seguridad para garantizar que llegue adecuadamente a sus destinos y no sean posibles falsas notificaciones. Aparte del tráfico que representa (prohibitivo en entornos de muchísimos usuarios) y de las medidas necesarias para garantizar, entre otros, los servicios de no repudio, rompe con la idea primigenia de contar con una pieza de información segura en sí misma (esto es válido tanto para el certificado como para la CRL) que puede transferirse sin necesidad de servicios complementarios de seguridad. No obstante, existen arquitecturas de seguridad pensadas para ámbitos de defensa militar que utilizan soluciones de este tipo.

La otra posible solución es securizar no sólo las piezas de información sino los servidores donde están contenidas, de forma que se forzase a que, antes de usar un certificado, fuese necesario hacer una consulta segura para saber si ese certificado sigue siendo válido. Esto puede llevarse a cabo implementando un Directorio seguro (opción bien definida en las correspondientes normas) o dotando de los adecuados servicios de seguridad el acceso a un servidor web que contuviese la información requerida. Para confiar en la veracidad de tan delicada consulta sería necesario no sólo confiar en que la CA va a generar adecuadamente los certificados, sino confiar también en los servidores y en la seguridad de las transacciones que realizan entre sí. Y este incremento de confianzas debe gestionarse con sumo cuidado (ya se ha dicho aquí que una buena norma de seguridad es fiarse lo menos posible del menor número posible de terceras partes).

Como contrapartida de todas estas dificultades, hay que decir que en la inmensa mayoría de los casos que pueden presentarse para garantizar lo que hemos denominado Seguridad Cívica, es decir, la provisión de servicios de seguridad para proteger las comunicaciones de los ciudadanos corrientes, un esquema de acceso a un determinado servidor para la recuperación de la última CRL emitida es más que suficiente para garantizar una adecuada «frescura» de los certificados y, por tanto, unos muy elevados niveles de confianza y libertad. Basta para ello con extremar las medidas de protección de las claves y con decidir cuál debe ser el periodo de generación de sucesivas CRL actualizadas. No obstante, la última de las soluciones que se han comentado para abordar la *revocación inmediata* puede usarse, de forma complementaria, en estos entornos cívicos para resolver los casos, que deben ser proporcionalmente pocos, en los que se desee realizar transacciones con un nivel máximo de seguridad.

DISMINUCIÓN DEL VOLUMEN DE INFORMACIÓN: DELTA-CRLs

En entornos reducidos, el problema de la revocación de certificados es un problema fácilmente abordable. No obstante, cuando el número de entidades a las que se ha expedido un certificado es muy elevado, aumenta la probabilidad de que se produzcan circunstancias que invaliden un certificado, debido a lo cual las listas de certificados revocados pueden alcanzar volúmenes difíciles de manejar. Una forma de paliar este problema es la generación de delta-CRLs.

Una *delta-CRL* es una CRL incremental, es decir, una lista donde aparecen solamente los cambios o variaciones respecto a una anterior lista sobre la que ésta se construye, recogiendo, por tanto, sólo los eventos ocurridos en el periodo de tiempo que media entre la emisión de ambos ejemplares de la misma serie. Ello obliga, obviamente, a las entidades usuarias de certificados a tener almacenada de forma segura la CRL anterior sobre la que ésta se apoya.

Para indicar que una CRL es únicamente una delta-CRL se usa una extensión especial:

Indicador de delta-CRL (deltaCRLIndicator).

La definición de esta extensión es sumamente sencilla. Además del correspondiente identificador de objeto, solamente aporta un número (del tipo *CRLNumber*) que hace referencia a la CRL en base a la cual aquí se anotan los cambios producidos. Su especificación es la siguiente:

deltaCRLIndicator EXTENSION ::= {
 SYNTAX **BaseCRLNumber**
 IDENTIFIED BY **id-ce-deltaCRLIndicator }**

BaseCRLNumber ::= CRLNumber

Uno de estos posibles cambios que se pueden producir es que un certificado desaparezca de la lista de certificados revocados. Según se describió en el último epígrafe del capítulo anterior, en la extensión *reasonCode* aparece un motivo, *removeFromCRL*, que puede ser empleado en una delta-CRL. En este caso, queda por aclarar si su desaparición de la lista es debida a que ha expirado su periodo de validez o a que se trata de un certificado suspendido que ha recuperado su condición de activo.

Una mala manera de reducir el tamaño de las listas es reducir el periodo de vigencia de los certificados, de tal forma que los certificados se caigan solos de la lista cuando su vigencia caduque. La decisión acerca del periodo de vigencia, que en principio debe ser tan grande como sea posible, debe ser tomada conforme a una política de seguridad rigurosa en la que deben influir una serie de parámetros que tiendan a minimizar las tareas, ninguno de los cuales debe ser precisamente este de adelgazamiento forzoso de las CRLs.

Otra forma de disminuir el tamaño de las listas que una determinada entidad necesita manejar es crear CRLs separadas y especializadas de menor tamaño. Esta especialización se puede conseguir bien dividiendo al colectivo de entidades del dominio en grupos más reducidos (cada uno con su particular CRL), bien separando las listas en función del motivo de la revocación, bien en base a otro criterio de interés en ese dominio. Eso se consigue con la funcionalidad de los *puntos de distribución de CRLs* que se discute en el siguiente apartado.

PUNTOS DE DISTRIBUCIÓN DE CRLs Y CRLs INDIRECTAS

Como ya se ha dicho, el Directorio es un sitio adecuado para servir de repositorio tanto para certificados como para CRLs. En los planteamientos iniciales, allá por 1988, se preveía que cada CA tuviese almacenada, en forma de un atributo X.500, su propia CRL en la que se recogiesen sólo los certificados revocados que esa CA había expedido a entidades finales. Además, caso de que hubiese emitido certificados a otras CAs, tendría una CRL especial para esos certificados.

Todo se preveía que girase en torno a atributos y a entradas del Directorio. Aparte de que este esquema era de por sí bastante rígido, el hecho de que el Directorio no cumpliera, ni de lejos, con las expectativas de implantación que se habían previsto,

hizo que con motivo de la publicación de la v3 del certificado y la v2 de la CRL se ampliase el margen de posibilidades y se flexibilizaran los planteamientos. En lo tocante a la autoría y ubicación de las CRLs, esta flexibilización está representada por varias extensiones:

1. Puntos de distribución de CRL (*cRLDistributionPoints*)

No es una extensión específica de CRLs, sino una extensión pensada para ser incluida en un certificado. Identifica el nombre y ubicación del punto o puntos de distribución en los que se puede encontrar información sobre una posible revocación del certificado en el que esta extensión aparece. Para captar la versatilidad que ofrece, analicemos la definición de esta importante extensión. Ésta es:

```

cRLDistributionPoints EXTENSION ::= {
    SYNTAX CRLDistPointsSyntax
    IDENTIFIED BY id-ce-cRLDistributionPoints }

CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint

DistributionPoint ::= SEQUENCE {
    distributionPoint [0] DistributionPointName OPTIONAL,
    reasons [1] ReasonFlags OPTIONAL,
    cRLIssuer [2] GeneralNames OPTIONAL }

DistributionPointName ::= CHOICE {
    fullName [0] GeneralNames,
    nameRelativeToCRLIssuer [1] RelativeDistinguishedName
}

ReasonFlags ::= BIT STRING {
    unused (0),
    keyCompromise (1),
    cACompromise (2),
    affiliationChanged (3),
    superseded (4),
    cessationOfOperation (5),
    certificateHold (6),
    privilegeWithdrawn (7),
    aACompromise (8) }

```

En primer lugar, cabe decir que en un dominio donde el colectivo de entidades finales sea muy numeroso, pueden éstas dividirse en grupos atendiendo a un criterio geográfico, de jerarquía o de funcionalidad. Aprovechando que esta extensión va añadida a cada certificado, se puede especificar que cada uno de estos grupos tenga su particular punto (o puntos) en el que distribuir las revocaciones de los certificados.

En la definición observamos que son varios los puntos de distribución que pueden definirse, de forma que **cada uno de ellos puede especializarse** en función de las razones o motivos que posteriormente se especifican. También podemos observar que existe un campo opcional para indicar el nombre de la autoridad (CA o no) que emitirá y firmará la correspondiente CRL, y que ésta puede ser **distinta de la CA que generó** el certificado. Cuando este valor no está presente, significa que es la misma.

Permite, además, que el nombre (lo que conlleva su dirección) del punto de distribución no coincida exactamente con el de la autoridad que emite la CRL. Además, al permitir el tipo *GeneralNames*, ya descrito cuando comentamos las extensiones del certificado, es posible asignarle varios nombres (dirección de correo, URL, etc.) para facilitar a los usuarios el acceso a la información que difunden.

Para especificar la posible especialización de cada punto de distribución (y posiblemente también de la autoridad que emite y firma cada CRL) existe el tipo *ReasonFlags*. Por ser del tipo BIT STRING, pueden marcarse una o más razones. Lo más importante y significativo de esta opción es que permite crear CRLs de distinta gravedad y frecuencia. En efecto, podríamos crear una CRL especializada solamente en certificados revocados a causa de estar comprometida la clave de las CAs que afectan a ese certificado. Igualmente, dado un colectivo numeroso de usuarios, podría crearse una CRL dedicada solamente a las revocaciones debidas a la pérdida de seguridad en sus claves privadas. Cabe esperar que el esmero de los usuarios en la custodia de sus claves conlleve que el número de certificados revocados por esta causa sea escaso, lo que permitiría aumentar la frecuencia de generación de esta lista (disminuyendo el riesgo que supone usar claves inseguras) sin aumentar excesivamente el volumen de información intercambiado. Por contra, los certificados revocados por otros motivos menos importantes y más frecuentes podrían alojarse en otras listas más voluminosas que apareciesen en periodos de tiempo más dilatados.

Esta extensión puede ser crítica o no crítica, ya que aunque la información que comporta es de suma importancia para resolver los problemas que se presentan en arquitecturas complejas y numerosas, si la entidad que usa el certificado no es capaz de interpretarla, pueden establecerse otros mecanismos que resuelvan la situación.

2. Nombre del punto de distribución de la CRL (*IssuingDistributionPoint*)

Esta es una extensión definida para ser incluida en una CRL concreta y sirve para informar sobre el nombre del punto de distribución en el que puede encontrarse esta CRL. Su sintaxis está definida como una SEQUENCE constituida por los siguientes campos:

distributionPoint	[0] DistributionPointName OPTIONAL,
onlyContainsUserCerts	[1] BOOLEAN DEFAULT FALSE,
onlyContainsCACerts	[2] BOOLEAN DEFAULT FALSE,
onlySomeReasons	[3] ReasonFlags OPTIONAL,
indirectCRL	[4] BOOLEAN DEFAULT FALSE }

Como puede observarse, para los posibles nombres del punto de distribución utiliza el tipo definido en la extensión anteriormente comentada. Además, proporciona información adicional sobre si sólo se utiliza para revocación de certificados de usuarios finales, o sólo para certificados de CAs, o si se limita a otras razones (utilizando para ello el tipo *ReasonFlags* definido en esa otra extensión).

En el último campo se hace referencia a un nuevo concepto: el de **CRL indirecta**. Si su valor es TRUE significa que esa CRL puede contener notificaciones de revocación para CAs diferentes de la CA emisora. Esta idea está ya presente en la extensión *cRlDistributionPoints* cuando se permite especificar el nombre de la autoridad emisora de la CRL y el del punto de distribución, que pueden ser distintos entre sí y, a su vez, distintos de la CA que ha generado el certificado en el que esa extensión se incluye. También tiene relación con otra extensión, no crítica, que sirve para dejar anotada esa dualidad. Es la extensión siguiente:

3. CA emisora del certificado (*certificateIssuer*)

Es una extensión para ser incluida en el campo *crlEntryExtensions* perteneciente a la secuencia que informa acerca de cada uno de los certificados revocados contenidos en la lista. Su estructura es muy simple y solamente contiene un identificador de la CA emisora del certificado revocado incluido en una CRL indirecta, es decir, una CRL que tiene activo el indicador *IndirectCRL*. Para aliviar la carga de información contenida en una CRL calificada como *indirecta* existen unas sencillas reglas que asignan su valor, por defecto, a la CA emisora de la CRL o a la CA que emitió el certificado que aparece previamente consignado en la lista.

SERVICIOS OSCP

Para conocer adecuadamente el estado de un certificado existe una alternativa a los métodos antes descritos orientados hacia consultas de distintas CRLs. Se trata de protocolos que permiten a la entidad comunicante que haga uso del servicio en cuestión obtener, mediante una llamada directa a un servidor, el estado de vigencia o revocación de los certificados consultados.

Uno de estos protocolos es el OSCP (*Online Certificate Status Protocol*) definido en la RFC 2560 del mismo título [Myer99]. Este protocolo permite, en general, obtener información sobre el estado de un certificado en un menor tiempo que el resultante de consultas a CRLs, pero requiere de mecanismos adicionales que garanticen tanto la protección de la información transmitida como las relaciones de confianza entre los clientes y los servidores.

Siendo cierto que estos servicios representan una mejora en relación con los «clásicos» de consulta de CRLs, hay que tener en cuenta que tampoco resulta la panacea para resolver totalmente el problema del no repudio. En primer lugar porque puede que el servidor no sea capaz de proporcionarnos la información requerida, dejándonos con la misma duda, y en segundo lugar porque, como podremos deducir de las definiciones del protocolo que a continuación se describen, el hecho de que la petición sea *on-line* no significa obligatoriamente que la información recibida acerca del estado del certificado esté exactamente ajustada al instante en el que se realiza la consulta. A veces, el aroma de las palabras nos hace incurrir en interpretaciones que no se ajustan a la realidad.

Como era de esperar, la consulta se divide en preguntas que se realizan al servidor y respuestas que éste proporciona. La definición formal de la petición recogida en la norma citada es la siguiente:

```
OCSPRequest ::= SEQUENCE {
    tbsRequest TBSRequest,
    optionalSignature [0] EXPLICIT Signature OPTIONAL }

TBSRequest ::= SEQUENCE {
    version [0] EXPLICIT Version DEFAULT v1,
    requestorName [1] EXPLICIT GeneralName OPTIONAL,
    requestList SEQUENCE OF Request,
    requestExtensions [2] EXPLICIT Extensions OPTIONAL }
```

En primer lugar, puede observarse que está previsto que la petición vaya firmada (*tbs* significa, según hemos visto en otros ejemplos, *to be signed*), aunque deja como

optativo el realizar esta operación. La petición es de por sí una secuencia de peticiones en cada una de las cuales es necesario reflejar la versión del protocolo, la identificación del peticionario y la lista de preguntas, *requestList*, que es a su vez una secuencia de peticiones (*SEQUENCE OF Request*). Es decir, el protocolo ofrece mucha versatilidad para que se aproveche una misma conexión *on-line* para realizar múltiples consultas. El formato de cada una de estas consultas está definido como:

```
Request ::= SEQUENCE {
    reqCert CertID,
    singleRequestExtensions [0] EXPLICIT Extensions OPTIONAL
}
```

Es decir, incluye un campo para la identificación del certificado y otro para añadir un conjunto de extensiones especiales definidas para matizar las condiciones en las que se realiza la consulta. Para la identificación del certificado utiliza un formato equivalente al empleado en la definición de otros protocolos, con indicación expresa de la identidad de la CA emisora y del número de serie que le asignó al generarlo.

Como resultado de la consulta realizada, el servidor enviará a la entidad peticionaria del servicio una **respuesta** firmada digitalmente indicando el estado de los certificados analizados. La clave pública pareja de la clave privada con que se firmen las respuestas debe ser perfectamente conocida por la entidad peticionaria, la cual debe confiar en la probidad del servidor y de sus gestores. Dependiendo de la configuración que se elija esta clave de firmado puede provenir de:

- La CA que generó el certificado de que se trate. Como quiera que una CA no debe prestar absolutamente a nadie su clave privada, esto significa que en este caso el servicio OCSP es un servicio anejo a la propia CA.
- Un servidor OCSP en el cual confía plenamente el cliente del servicio.
- Un servidor OCSP «autorizado» por la CA que generó el certificado en cuestión. Para ello, el servidor OCSP dispone de un certificado especial emitido y publicado por esa CA en el que se reconoce su habilitación para generar respuestas relativas a certificados generados por dicha CA.

Está previsto que los servidores OCSP puedan generar distintos tipos de respuestas. De entre ellas se define un *tipo básico* de respuesta que debe ser conocido e interpretado por todos los servidores y clientes. Con la intención de abreviar y de recoger aquí las características más significativas de estos servicios, en los párrafos que siguen nos limitaremos a reflejar resumidamente parte de la información que aparece en ese tipo básico de respuesta.

En la especificación formal de la respuesta básica se recoge, como ya se ha dicho, que los datos que constituyen la respuesta (*ResponseData*) vayan firmados por el servidor. La definición de estos datos es:

```
ResponseData ::= SEQUENCE {
    version                [0] EXPLICIT Version DEFAULT v1,
    responderID            ResponderID,
    producedAt             GeneralizedTime,
    responses              SEQUENCE OF SingleResponse,
    responseExtensions    [1] EXPLICIT Extensions OPTIONAL
}
```


Como puede observarse, se recoge, además de la versión y de un campo de extensiones, la identificación del servidor que responde, la hora en la que la respuesta es generada y una secuencia de respuestas particularizadas (*SingleResponse*) para cada certificado analizado. La estructura de esas respuestas particularizadas es la siguiente:

```
SingleResponse ::= SEQUENCE {
    certID CertID,
    certStatus CertStatus,
    thisUpdate GeneralizedTime,
    nextUpdate [0] EXPLICIT GeneralizedTime OPTIONAL,
    singleExtensions [1] EXPLICIT Extensions OPTIONAL
}
```

Además de un campo para extensiones, cuyas posibilidades obviamos explicar en aras a la brevedad de estos comentarios, se incluye un campo para identificar el certificado de que se trata, otro para indicar el estado del certificado y otros dos para reflejar las condiciones temporales en que el análisis ha sido realizado. El primero de ellos (*thisUpdate*) indica el momento en el que el servidor OCSP ha recibido información relativa a ese certificado. El segundo (*nextUpdate*) indica el momento en el que el servidor espera recibir nueva información relativa a ese certificado.

Realmente, todo este viaje tiene como principal objetivo conocer el estado del certificado sobre cuya validez existen dudas. Todo lo demás no son sino parámetros necesarios pero complementarios. El valor proporcionado por el servidor OCSP sobre el estado del certificado es uno de los tres siguientes: **Bueno** (*good*), **Revocado** (*revoked*) o **Desconocido** (*unknown*).

Como puede observarse de la definición de la respuesta (*SingleResponse*) no es obligatorio que el servidor se comprometa a informar a la entidad peticionaria acerca del estado del certificado **en el momento exacto** en el que la petición ha sido realizada. Antes al contrario, permite la existencia de un tiempo de latencia, el comprendido entre *thisUpdate* y *nextUpdate*, durante el cual el estado del certificado no está siendo actualizado. Bien es verdad que estos periodos de latencia pueden ser mucho más reducidos que los que se implanten en los distintos tipos de CRLs. Pero sólo eso.

Sí existe, en cambio, un factor que diferencia positivamente el servicio que puede obtenerse de un OCPS en comparación con la consulta directa de CRLs. En el primero de ellos, la entidad comunicante que necesita comprobar la vigencia de un certificado (engarzado en la cadena de certificados que constituyen un camino de certificación) queda relevada de la tarea de localizar en los correspondientes repositorios la información que le garantice la no revocación de ninguno de ellos.

En casos sencillos, esta diferencia puede no ser muy relevante, pero cuando se trata de infraestructuras de certificación organizativamente complejas, las tareas de búsqueda pueden representar una enorme complejidad para las entidades finales. En este sentido, los servicios OCSP actúan por **delegación** de la entidad que necesita hacer esas comprobaciones. Este planteamiento va en la misma línea que los emergentes *servicios delegados* DPV y DPD que se comentan someramente en el epígrafe 8.10 del siguiente capítulo.

7.8. INFRAESTRUCTURA DE GESTIÓN DE PRIVILEGIOS, PMI

El fin primero y último del certificado de la clave pública es, como su propio nombre indica, garantizar la validez de la clave pública de una entidad comunicante. Lo que ocurre es que como esa clave pública es pareja de otra clave privada (que esa entidad mantiene en secreto y es de su exclusiva propiedad) el certificado sirve para ligar esa clave pública con *la identidad* de un usuario final. Por tanto, este tipo de certificado sirve para la provisión de servicios de *autenticación* que permiten a una determinada entidad probar quién es. Debido a ello, a estos certificados podríamos también llamarlos *certificados de identidad*.

Luego viene, como hemos visto en la definición del certificado X.509, lo de añadir parámetros que ayuden a usarlo con propiedad e informen de detalles relacionados con su ubicación en la infraestructura de certificación. El campo de extensiones permite multiplicar las posibilidades de inclusión de ese tipo de detalles complementarios.

Pero todo tiene un límite. Una forma sencilla de establecer un servicio de *control de acceso* es utilizar el certificado de la clave pública para garantizar la identidad del cliente y posteriormente averiguar los *privilegios* que él tiene en el acceso a un recurso. Esta sería una forma de obtener *autorizaciones* basada solamente en la identidad. Pero esto no es suficiente en muchos casos; a veces se requieren mecanismos de control de acceso basados en reglas y en roles que se pueden plasmar en parámetros y atributos. Es fácil pensar que bien podrían ir estos atributos insertados en el certificado. El incluir esos datos sería rebasar el límite de lo que constituye la concepción misma de un certificado de la clave pública.

Pero hay algo más. El certificado de la clave pública sirve para identificar a la entidad que lo posee durante periodos de tiempo que pueden ser bastante largos. En cambio, los privilegios asignados a esa misma entidad cambiarán con mayor frecuencia, dependiendo de circunstancias de uso de otros servicios y servidores para los que se pretenda tener regulado el control de acceso. De estar insertados en el certificado de la clave pública, cambiar esos privilegios supondría tener que revocar ese certificado. Además, una CA es una tercera parte de confianza en lo que se refiere a garantizar la validez y el adecuado uso de la clave pero no es razonable pensar que dentro de un dominio de seguridad las autorizaciones para el uso de sistemas y recursos tengan que ser también depositadas en ella.

Por todo ello, y por algún requisito más que analizaremos más adelante, para la inclusión de información sobre privilegios se ha considerado conveniente definir una nueva estructura de datos firmada: el Certificado de Atributos.

CERTIFICADOS DE ATRIBUTOS Y AUTORIDADES DE ATRIBUTOS

La definición de este nuevo certificado está incluida en la Recomendación X.509 de ITU-T y es aceptada por otros organismos de normalización¹⁰. La estructura del Certificado de Atributos es similar a la ya estudiada del certificado de la clave pública con la diferencia de que éste no contiene ninguna clave y sí un conjunto de atributos que especifican circunstancias tales como: pertenencia a determinados grupos, roles desempeñados, acreditaciones de seguridad y otros parámetros asociados con la entidad propietaria de certificado.

La Autoridad de Atributos es una TTP especial que genera y firma el Certificado de Atributos asignando los privilegios que correspondan a cada entidad.

En la Figura 7.19 se representa la estructura del Certificado de Atributos. Como puede observarse, su formato es muy parecido al del certificado de la clave pública que estudiamos en el Capítulo 6, cuya representación genérica se recoge en la Figura 6.13.

El significado de estos campos es muy similar a los equivalentes del certificado de la clave pública, aunque debido a la fecha en la que fueron definidos ofrecen más versatilidad que aquellos y no tienen la «hipoteca» de ser compatibles con versiones muy antiguas. Los distintos campos son los siguientes:

- Versión*. Indica la versión del certificado conforme a la cual están definidos formalmente sus distintos campos. La versión que tiene vigencia es la v2, representada por el valor 1.
- Nombre del titular (holder)*. Es el nombre de la entidad comunicante a la que se le ha expedido el certificado. No tiene por qué ser obligatoriamente un *nombre X.500* como en el certificado de la clave pública. Por contra, existe bastante versatilidad en la norma para cumplimentar este parámetro. Una opción puede ser usar el tipo *GeneralNames* (que ya antes hemos analizado) y otra posibilidad es apuntar de forma explícita a uno o a varios certificados de identidad (certificados de la clave pública) emitidos a nombre del titular.

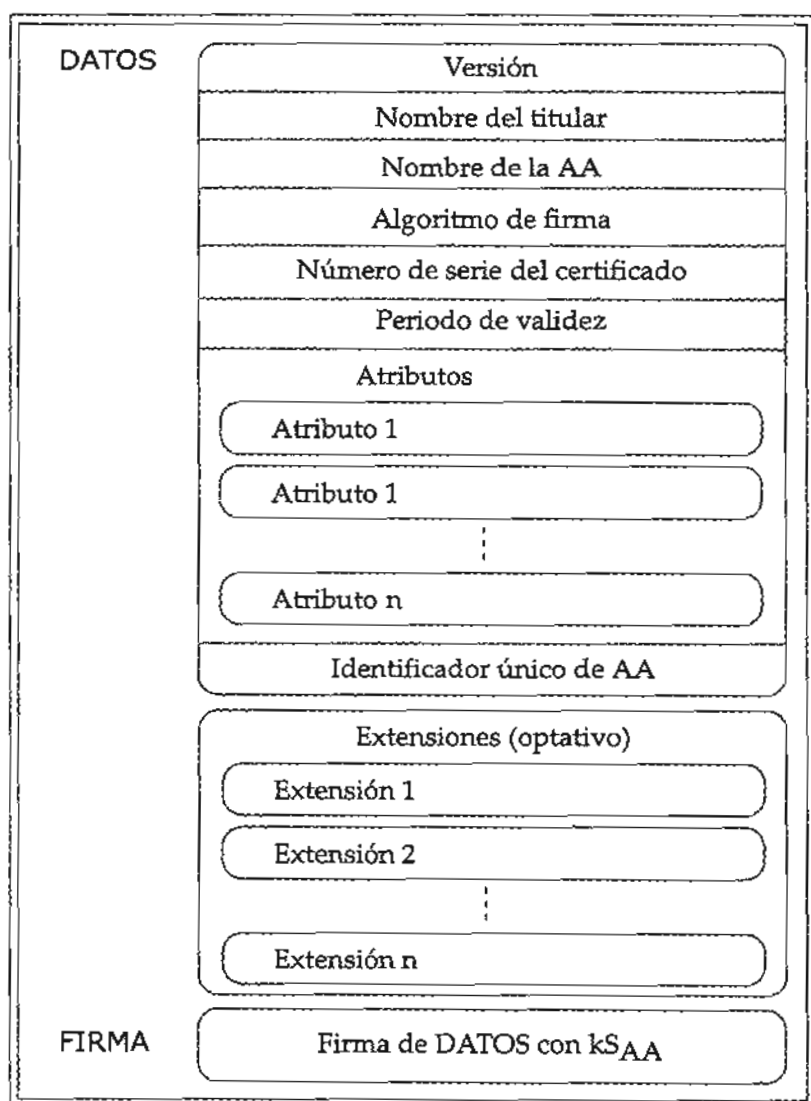


Figura 7.19. Estructura del Certificado de Atributos.

- c) *Nombre de la AA emisora (issuer)*. Permite indicar el nombre telemático de la AA que genera este certificado. No tiene por qué ser obligatoriamente un *nombre X.500*, sino que es del tipo *GeneralNames* acompañado (en la versión 2) de otros parámetros complementarios.
- d) *Algoritmo de firma del certificado (signature)*. Identifica el algoritmo utilizado por la AA para firmar los datos. Lo habitual es que sea RSA o DSA.
- e) *Número de Serie*. Cada AA irá numerando correlativamente todos los certificados que emita, por lo que el número de serie, junto con la identidad de la AA, sirve de identificador único para este certificado.
- f) *Periodo de validez*. Indica el comienzo y el final del periodo durante el cual el certificado es válido.
- g) *Atributos*. Se trata de una secuencia de atributos cada uno de los cuales tiene el mismo formato que hemos visto para los atributos X.500. Es decir, un identificador único de atributo y un valor o conjunto de valores asignados.
- h) *Identificador único de AA emisora*. Es un campo opcional que sirve para, en determinadas circunstancias, quitar ambigüedad respecto a la identidad de la entidad emisora.
- i) *Extensiones*. El formato de este campo es exactamente coincidente con el que hemos visto antes para el certificado de la clave pública y para las CRLs. Se trata, por tanto, de un conjunto ilimitado de estructuras que permiten añadir información complementaria. En principio, cualquiera de las extensiones que hemos visto hasta ahora podrían incluirse aquí, aunque están pensadas varias extensiones específicas para ser usadas para cubrir los objetivos que se procuran con los certificados de atributos.

En esta descripción de los campos del Certificado de Atributos se ha prescindido de detalles que serán analizados cuando se discuta la especificación formal de este certificado (en el Capítulo 9 dedicado al control de acceso). En ese mismo lugar se estudiarán también las distintas extensiones propuestas y la utilidad que de ellas puede derivarse.

ORGANIZACIÓN Y COMPONENTES DE UNA PMI

Además de las Autoridades de Atributos (AAs) y de los Certificados de Atributos (ACs), en una Infraestructura de Gestión de Privilegios (PMI) cabe distinguir los siguientes componentes:

- *Usuarios del Certificado de Atributos (AC users)*. Son aquellas entidades que se fían de la información contenida en estos certificados. Obviamente, pueden hacer uso de estos certificados aunque no tengan expedido a su nombre ningún certificado de atributos. Se corresponden con lo que se ha denominado también *relying parties* en el caso de los certificados de la clave pública.
- *Entidades verificadoras de certificados de atributos (ACs verifiers)*. Son las que, basándose en la firma de los certificados, comprueban o dan por válidos los datos allí contenidos.
- *ARLs: listas de revocación de certificados de atributos (AC revocation lists)*. Tienen una significación y papel equivalente al de las CRLs antes estudiadas.
- *Repositorios*. Sirven para almacenar certificados de atributos y ARLs. Para su implantación pueden usarse los mismos sistemas empleados para el almacenamiento de certificados de la clave pública y CRLs.

Según se indicaba en la introducción de este epígrafe, la aparición de los certificados de atributos tiene que ver con las autorizaciones y los privilegios asignados a una entidad que pretende acceder a un determinado recurso. De acuerdo con la norma que define estos certificados, la *autorización* es la transferencia de *privilegios* desde una entidad que posee estos privilegios a otra entidad que los recibe. La expedición de un certificado de atributos a una entidad final por parte de una AA es, por tanto, un mecanismo de autorización.

De igual forma, si una AA expide un certificado de atributos a otra AA, lo que está haciendo es transferirle alguno de los privilegios que posee y delegando en ella para que realice, a su vez, autorizaciones.

De la misma manera que en una determinada Infraestructura de Certificación pueden existir múltiples CAs organizadas conforme a cierta jerarquía, también es teóricamente posible la existencia de una Infraestructura de Gestión de Privilegios con múltiples AAs organizadas jerárquicamente y la necesidad de gestionar *cadena*s de certificados de atributos y sus correspondientes *camino*s de *confianza*. En estos casos, la AA raíz que gobierna la asignación de atributos recibe el nombre de Fuente de Autoridad o SOA (*Source of Authority*). No obstante, por el momento cabe recomendar que no se aborden arquitecturas de este tipo debido a la complejidad que ello conlleva. Lo que sí puede ser recomendable es que en una determinada Infraestructura se implanten diversas AAs especializadas cada una de ellas en privilegios distintos.

RELACIÓN ENTRE LA PMI Y LA PKI

En realidad, la PMI está concebida como una infraestructura autónoma pero fuertemente relacionada con la PKI. Esto es así porque mediante la PMI se puede obtener la información necesaria para conocer los privilegios que una determinada entidad comunicante tiene a la hora de acceder a un recurso, pero no se pueden obtener garantías sobre su identidad. Para ello es necesario recurrir a un servicio de autenticación, lo que implica el acceso a certificados de la clave pública.

Como se ha comentado más arriba, el campo de certificado relativo a la identidad del titular (*holder*) puede apuntar de forma explícita a uno o a varios certificados de identidad (certificados de la clave pública) emitidos a nombre del titular. En la Figura 7.20 se representa esta ligazón en el caso de que la entidad titular del certificado de atributos tenga la misma identidad en varios certificados de clave pública.

En cuanto a terminología también surge un cierto solapamiento entre la PMI y la PKI. Con la aparición de este nuevo certificado, además de la confusión que a veces se produce entre ambos tipos de certificados, se nos presenta una cierta ambigüedad en la nomenclatura que venimos utilizando. En efecto, el *certificado de la clave pública* (al que hemos venido haciendo reiterada referencia en este y otros capítulos) lo hemos denotado frecuentemente como *certificado X.509* debido a que la estructura especificada en esta norma es hoy en día la universalmente aceptada. Ahora nos encontramos con que el *certificado de atributos* también está definido en la Recomendación X.509, por lo que la denominación *certificado X.509* podría resultar ambigua y válida para ambos tipos de certificados. Para resolver esa confusión, en las propuestas emanadas dentro del entorno de las RFCs se ha optado por usar para el primero de ellos el acrónimo PKC (*public key certificate*) y para el segundo de ellos AC (*attribute certificate*).

Aun admitiendo lo acertado de esta decisión, en las descripciones que siguen, como sinónimo de *certificado de la clave pública* seguiremos utilizando el término

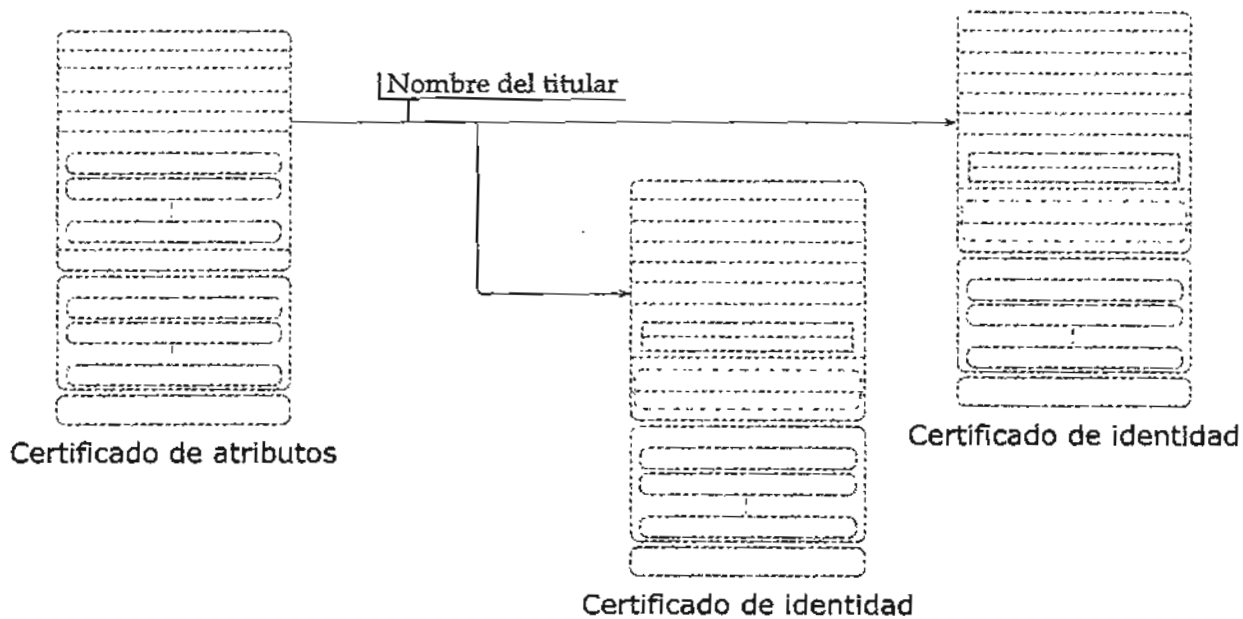


Figura 7.20. Relación entre certificados de atributos y certificados de identidad.

certificado X.509 o, simple y llanamente, el de *certificado*. Cuando queramos hacer referencia al *certificado de atributos* (AC) lo haremos así, de forma expresa y con todas sus letras. La razón para ello es que este es un texto con pretensiones didácticas y la repercusión que hoy en día tiene el certificado de la clave pública en el mundo de la seguridad en redes telemáticas es inmensamente superior a la de los certificados de atributos, cuya aplicación está limitada a casos restringidos y cuyo desarrollo es aún emergente (no sabemos qué pasará en un futuro). No nos parece útil hacer reiteradamente un distinguo para algo cuya aparición en escena va a ser tan poco frecuente. Debido a ello, insistimos, para nosotros, decir *certificado* será decir, tomando la parte por el todo, *certificado de la clave pública*.

7.9. AUTORIDADES DE SELLADO DE TIEMPO: TSAS

En el intercambio de información a través de redes telemáticas es necesario en muchas ocasiones reflejar de forma fiable la fecha y hora en que un documento ha sido creado o enviado o recibido. En muchos casos, como ocurre en las comunicaciones convencionales, bastará con la fecha que el autor o el receptor de un mensaje afirmen que tal evento se ha producido. En otros, en cambio, es necesario contar con *evidencias fiables* que reflejen esa situación. A estas evidencias las denominaremos *sellos de tiempo* (*time-stamp tokens, TST*) y serán generadas por unas TTPs especializadas en estas tareas: las *Autoridades de Sellado de Tiempo* (*Time-Stamping Authorities, TSAs*).

La necesidad de sellados de tiempo realizados por terceras partes de confianza es mucho mayor en las comunicaciones telemáticas que en las comunicaciones convencionales que usan el papel como soporte. En estas últimas, la práctica diaria hace que para la mayoría de los intercambios de información (por ejemplo, una carta firmada) la validez de las fechas se sustente en la relación de confianza que pueda existir entre los participantes en la comunicación. Sólo en contados casos se recurrirá al correo certificado, y solamente en casos especiales se acudirá a un notario para que dé fe del contenido y del momento en que se genera una notificación.

Por contra, en el caso de las comunicaciones telemáticas seguras, todos los protocolos que utilicen cifrados con algoritmos de clave pública necesitan presentar garantías acerca de la validez del par de claves con el que se opera, lo que equivale a utilizar certificados vigentes y no revocados **en el momento** en el que la operación se realiza. Un documento firmado digitalmente será válido si **en el momento de la firma** es válido el certificado de la clave pública pareja de la clave privada con que se firmó. Y podrá seguir siendo válido durante mucho tiempo después aunque ese certificado haya sido revocado. En este caso, el quid de la cuestión es tener garantías acerca del momento en el que la firma fue realizada.

Esta necesidad de reflejar el momento en el que se realizan ciertas operaciones telemáticas ha motivado que diversos organismos de normalización hayan desarrollado propuestas para especificar los servicios de sellado de tiempo¹¹. Los protocolos desarrollados se basan en dos formas distintas de generar las evidencias:

- a) Mediante **sellos de tiempo independientes**. La entidad que requiere un marcado de tiempo en un mensaje le envía el *hash* de ese mensaje a la TSA y ésta le devuelve una ligazón entre esa unidad de datos y un sello de tiempo. Para cada solicitud la TSA genera un sello de tiempo distinto e independiente. Este comportamiento es el que se recoge en la norma ISO/IEC 18014 *Time-stamping services. Part 2: Mechanisms producing independent tokens* y en la RFC 3161 *Time-Stamp Protocol (TSP)*.
- b) Mediante la generación de **sellos de tiempo ligados** (*linked tokens*). La TSA se apoya en la seguridad de la función *hash* para enlazar un sello de tiempo estampado en un documento con otros sellos de tiempo previamente generados por ella. Mediante el uso de *hash* acumulados garantiza que se conozca que un documento ha «aparecido» antes que otro. La fortaleza del sistema reside en la integridad y fiabilidad del repositorio en el que se publiquen los sellos ligados y no requiere obligatoriamente la existencia de una PKI. Entre las propuestas formuladas por los organismos de normalización, solamente los trabajos (en curso) llevados a cabo en ISO para la definición de la tercera parte de la ISO 18014 contemplan esta alternativa.

El modelo recogido en la opción a) es, con mucho, el más aceptado y es en el que se apoyan las propuestas de los restantes organismos de normalización. Por ello, en los párrafos que siguen, será en el que nos basemos para definir la estructura y funcionamiento de las TSAs.

COMPONENTES DEL SERVICIO DE SELLADO DE TIEMPO

En la Figura 7.21 se representan las entidades presentes en un proceso de sellado de tiempo y el intercambio de información que se produce entre ellas. Antes de describir los componentes de esta comunicación, pasemos a definir con mayor precisión algunos conceptos:

- *Fecha y hora (point in time)*. Representa un *instante de tiempo* puntual que será reflejado mediante: año, mes, día del mes, hora y segundos (podrían, incluso, reflejarse fracciones de segundo). *Fechado* será la operación de incluir fecha y hora en una estructura de datos.
- *Sello de tiempo (time-stamp token)*. Es una estructura de datos, verificable criptográficamente, que contiene una ligazón entre una unidad de datos y un instante de tiempo (fecha y hora).

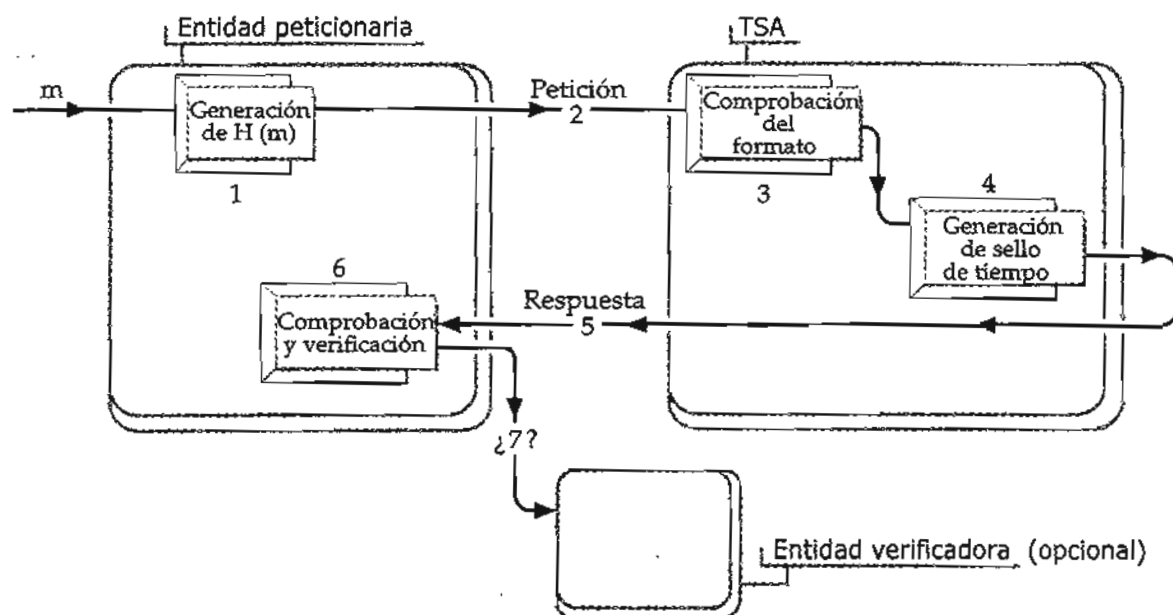


Figura 7.21. Intercambio de información para la obtención de un sello de tiempo.

- *Servicio de sellado de tiempo (time-stamping service)*. Es un servicio que produce evidencias (*sellos de tiempo*) que garantizan que determinados datos existían antes de una fecha y hora concretas.

Las entidades implicadas en el proceso de sellado de tiempo son:

- Entidad peticionaria. Es la entidad que posee los datos que desea sean sellados.
- Autoridad de sellado de tiempo, TSA. Es la que provee el servicio de sellado.
- Entidad verificadora. Verifica la validez de los sellos de tiempo, pudiendo hacer uso de la información complementaria que obtenga al efecto.
- Entidad que confía en el sello de tiempo (*relying party*). Es la entidad que recibe un mensaje con sello de tiempo y confía en la validez de ese sello. Esta garantía puede obtenerla de por sí o mediante la ayuda de una entidad verificadora.

En la Figura 7.21 se describen los pasos que componen el proceso de sellado, desde la solicitud hasta la obtención del sello de tiempo. No se incluye en ella, por tanto, la entidad que confía en el sello de tiempo. Son los siguientes:

1. La entidad peticionaria, que posee los datos que van a ser sellados, genera el valor resumen (*hash*) del mensaje *m*.
2. Dicha entidad envía la petición a la TSA correspondiente. En esta petición se incluyen, obligatoriamente, el valor hash y la identificación del algoritmo de resumen utilizado. Opcionalmente, puede incluir otros parámetros complementarios.
3. La TSA comprueba la corrección del formato de la petición.
4. La TSA genera el sello de tiempo en función de datos propios y de la información contenida en la petición. El sello de tiempo es una estructura de datos que contiene:
 - Parámetros sobre fecha y hora obtenidos de una fuente fiable.
 - Los datos enviados por la entidad peticionaria.

- Los datos generados por la TSA para ligar, mediante una transformación criptográfica robusta, los datos de la petición y la información de fechado.
5. La TSA envía la respuesta a la entidad peticionaria.
 6. La entidad solicitante comprueba de forma inmediata si la respuesta es completa y correcta, es decir, si cumple con los requisitos especificados en la petición, de tal forma que puedan ser también verificables por parte de una entidad (*relying party*) que reciba ese mensaje con el sello de tiempo.
 7. Opcionalmente, la entidad peticionaria puede encargar el trabajo de validación a una entidad verificadora.

Como puede comprobarse, la entidad peticionaria se fía de la TSA sólo en lo que se refiere al fechado, pero no le envía en claro el contenido del mensaje a sellar, sino solamente el valor hash. Posteriormente, se supone que enviará a una tercera entidad el mensaje en cuestión con su correspondiente sello de tiempo.

PROTOCOLO DE SELLADO DE TIEMPO: FORMATO DE LOS MENSAJES

Además de otras especificaciones y reglas que lo definen, lo más significativo del protocolo de sellado de tiempo es el formato de los dos mensajes que se intercambian la entidad peticionaria y la TSA: la petición de sellado de tiempo y la respuesta (que se corresponden con los pasos 2 y 5 de la Figura 7.21). Analizando los campos que los constituyen aparecen los detalles más importantes del comportamiento del sistema.

A continuación vamos a comentar la especificación formal de estos dos mensajes. Su formulación es prácticamente coincidente en la ISO 18014-2 y en la RFC 3161.

Petición de sellado de tiempo (*TimeStampReq*)

Su descripción formal es la siguiente:

```

TimeStampReq ::= SEQUENCE {
    version                INTEGER { v1(1) },
    messageImprint         MessageImprint,
                           -- un identificador OID de un algoritmo hash y el valor hash
                           -- del dato que será sellado
    reqPolicy              TSAPolicyId                OPTIONAL,
    nonce                  INTEGER                    OPTIONAL,
    certReq                BOOLEAN                   DEFAULT FALSE,
    extensions             [0] IMPLICIT Extensions    OPTIONAL
}

```

Como en otras ocasiones, hemos optado por traducir los comentarios de la especificación al español (eso no afecta para nada a la formalidad de la especificación). El significado de los seis campos es el siguiente:

Al igual que en otras especificaciones, el campo *version* sirve para indicar la versión del protocolo. La codificación del tipo *MessageImprint* es la siguiente:

```

MessageImprint ::= SEQUENCE {
    hashAlgorithm         AlgorithmIdentifier,
    hashedMessage         OCTET STRING
}

```

El tipo opcional *TSAPolicyId* se define como un identificador de objeto donde se describe la política de sellado bajo la que opera la TSA.

El campo *nonce* es un entero aleatorio que genera la entidad peticionaria y sirve a modo de identificador de la solicitud, por lo que ese mismo valor deberá ser incluido en la respuesta que envíe la TSA. Como es un número grande (del orden de los 64 bits) es poco probable que se repita en una nueva solicitud de sellado.

Si el campo *certReq* toma el valor TRUE, ello significa que en la respuesta la TSA debe incluir su propio certificado.

Como en otras especificaciones que ya hemos visto, el campo *extensions* deja abierta la puerta para que se incluya información adicional que ayude a decidir las condiciones en las que se realiza la solicitud de sellado.

Formato de la respuesta (*TimeStampResp*)

La descripción formal de este mensaje es la siguiente:

```

TimeStampResp ::= SEQUENCE {
    status PKIStatusInfo,
    timeStampToken TimeStampToken OPTIONAL
}
    
```

Es decir, la respuesta está compuesta por dos campos: información sobre el resultado de la petición (*status*) y el sello de tiempo propiamente dicho (*timeStampToken*). Este último es optativo porque en el caso de que la respuesta no sea favorable, la TSA no manda el sello (*token*). Para representar el resultado de la operación se utiliza el tipo *PKIStatusInfo* mediante el cual se asigna un valor entero a *status*. El valor 0 representa el estado «garantizado» y el valor 1 representa «garantizado con modificaciones». Cuando la respuesta contiene uno de estos dos valores es obligatorio mandar el sello de tiempo. Existen otros valores para informar sobre dificultades en la petición como rechazos, demoras y revocaciones.

El tipo *TimeStampToken* es una estructura que se usa para encapsular la información acerca del sello. El sello de tiempo, TTS, contiene la firma digital de la TSA e información sobre el certificado que garantiza su clave pública. La estructura (con los comentarios en español) de la información sobre el sello de tiempo *TTSInfo* es la siguiente:

```

TSTInfo ::= SEQUENCE {
    version INTEGER { v1(1) },
    policy TSAPolicyId,
    messageImprint MessageImprint,
    -- DEBE tener el mismo valor que el campo similar en TimeStampReq
    serialNumber INTEGER,
    genTime GeneralizedTime,
    accuracy Accuracy OPTIONAL,
    ordering BOOLEAN DEFAULT FALSE,
    nonce INTEGER OPTIONAL,
    -- DEBE estar presente si el campo similar estaba presente en
    -- TimeStampReq. En tal caso DEBE tener el mismo valor
    tsa [0] GeneralName OPTIONAL,
    extensions [1] IMPLICIT Extensions OPTIONAL
}
    
```

El significado de los distintos campos es el siguiente:

Los tres primeros campos (*version*, *policy* y *messageImprint*) tienen el mismo significado que el ya comentado en el formato de la petición, con la particularidad que en la estructura de la respuesta la información sobre política debe aparecer obligatoriamente.

El campo *serialNumber* representa un número de serie asignado por la TSA a cada sello de tiempo que genera, con lo cual un determinado sello de tiempo está claramente identificado por el nombre de la TSA que lo generó y por el número de serie que le fue asignado.

El campo *genTime* representa la fecha y hora a la que el sello de tiempo fue creado. Es del tipo *GeneralizedTime*, que se expresa como UTC (*coordinated universal time*) para tener una hora única en todo el planeta, evitando la confusión de las horas locales (su sinónimo es «Zulu Time», que es el usado por la aviación civil y se representa por una «Z»). Su formato contiene: dos dígitos para cada uno de los siguientes datos: centuria (19-99), año (00-99), mes (01-12), día del mes (01-31), hora (00-23), minuto (00-59), segundo (00-59). Optativamente puede llevar otros dos dígitos, separados por un punto, para representar fracciones de segundo. La cadena se cierra con una «Z» indicadora de que es UTC (Zulu Time).

El campo optativo *accuracy* representa la desviación o la exactitud del campo *genTime* en comparación con UTC. Una explicación pormenorizada sobre las diferentes representaciones del tiempo puede encontrarse en las normas citadas y en [PPR03].

Cuando el campo *ordering* está presente y tiene valor TRUE, los distintos sellos de tiempo generados por la misma TSA estarán siempre ordenados sobre la base del contenido del campo *genTime*. En caso contrario, la ordenación depende del campo *accuracy*.

Los campos *nonce* y *extensions* tienen aquí el mismo significado que en la estructura de datos *TimeStampReq* utilizada para la petición de sellado. Como se comenta en la definición, el valor del campo *nonce* debe ser el mismo en ambos mensajes.

El último campo, *tsa*, sirve para identificar de forma no ambigua a la TSA que ha generado el sello de tiempo. El tipo utilizado es el *GeneralizedTime*, ya comentado antes reiteradamente.

Además de los mensajes de petición y respuesta, en las normas y documentos antes citados se recogen otras estructuras de datos y otras particularidades del protocolo de sellado de tiempo que, por cuestión de simplificación, obviamos describir aquí. En los párrafos que preceden se ha procurado resaltar los elementos más significativos de este servicio, describiendo con cierto detalle el formato de los dos mensajes más importantes, con objeto de obtener una idea suficientemente precisa de lo que representa un sello de tiempo. Como ya se ha dicho, cada vez resulta más necesario que en el intercambio de información en redes telemáticas aparezcan estas garantías. Quizás el mecanismo que más urgentemente demanda la presencia de sellos de tiempo es la *firma electrónica*, según se discutirá en el siguiente capítulo.

NOTAS AL CAPÍTULO 7

¹ Aunque en teoría también podrían incluirse máquinas automáticas y servidores, por la naturaleza de las comunidades que se constituyen bajo una configuración de este tipo, suelen ser comunicaciones personales las que presentan mayor interés. Para incluir en el dominio a una máquina automática (una impresora, por ejemplo) las reglas de comportamiento tendrían

que estar definidas de forma no ambigua para permitir a esa entidad automática tomar decisiones acerca de la validez o no de los certificados recibidos.

² Esta CA, la que ocupa el nivel más alto de la jerarquía, suele denominarse *Top CA* y también, por ocupar la raíz del árbol, *Root CA*. No obstante, en el documento del PKIX WG titulado *Internet X.509 Public Key Infrastructure: Roadmap* [ADT02] se pretende hacer un distinguo entre ambos términos, cosa que no se hace en otros documentos del PKIX. Así, se reserva el nombre de *Top CA* para la de más alta jerarquía, mientras que se denomina *Root CA* (*CA Raíz*) a aquella en la que confían directamente un conjunto de entidades finales (a través de un mecanismo extratelemático), sin que esa confianza se derive de una confianza delegada en otra CA, pudiendo ocupar cualquier puesto en «el escalafón» jerárquico. A nuestro parecer, este distinguo introduce más confusión que claridad a la hora de entender este tipo de estructuras.

³ La primera propuesta reconocida que surgió para abordar una infraestructura de este tipo se dio dentro del proyecto europeo PASSWORD (*Piloting Authentication and Security Services with OSI Application for RTD Information*), cuyo objetivo principal era el desarrollo (1993-1994) de un entorno de seguridad para la comunidad investigadora europea. El Departamento Diatel de la UPM, en el cual desarrolla su labor investigadora el autor de este texto, participó en el proyecto como «pilot site». Para salvar las reticencias existentes entre los diversos países se definió una estructura de certificación similar a la recogida en la Figura 7.4 b), aunque al final del proyecto se vio que este tipo de soluciones introducen muchas complicaciones y los problemas que resuelven tienen más que ver con reticencias y rencillas que con situaciones reales, por lo que se decidió que era más adecuado definir una estructura con una sola CA en el máximo nivel jerárquico.

⁴ Desde 1988, año en el que ISO asumió la versión inicial del Servicio de Directorio elaborada por el CCITT (del que ITU-T es su heredero), las sucesivas actualizaciones de este conjunto de normas han sido realizadas de forma consensuada por ambos organismos internacionales. La Recomendación X.500 de ITU-T está constituida por nueve normas, una de las cuales es la X.509 (debido a las sucesivas actualizaciones y supresiones estas recomendaciones no tienen asignados todos los números correlativos y comprenden desde la X.501 a la X.525). En ISO, el Directorio se recoge bajo la denominación genérica ISO/IEC 9594, dividida en nueve normas (desde la ISO/IEC 9594-1 a la ISO/IEC 9594-9) idénticas a las correspondientes recomendaciones de ITU-T (la X.509 se corresponde con la ISO/IEC 9594-8). Para más detalle, consúltese el anexo de Bibliografía.

⁵ En realidad, la X.500 no recoge, en su versión actual, el término «Sistema de Directorio», que sí aparecía en los primeros borradores de la norma. Posteriormente, lo que abarca el conjunto de DSAs (Agentes del Sistema de Directorio) aparece en la norma bajo el ampuloso nombre de «El Directorio», aunque se sigue manteniendo el concepto de «Protocolo del Sistema de Directorio» (DSP). Aquí se ha optado por utilizar una representación diferente de la recogida en la norma para acercar mejor al lector la imagen de lo que es un sistema distribuido de Directorio y la semejanza que presenta en relación con otros sistemas telemáticos distribuidos [Carr89], como es el caso del Sistema de Tratamiento de Mensajes (MHS) desde los que, por otra parte, puede ser accedido.

⁶ Aunque en 1989 ya apareció una primera versión de sus especificaciones (RFC 1113, 1114 y 1115) fue en 1993 cuando se publicó una versión mejorada de las definiciones y protocolos allí propuestos. Se trata de las normas: RFC 1421 *Privacy Enhancement for Internet Electronic Mail. Part I: Message Encipherment and Authentication Procedures*, RFC 1422, que contiene la *Part II: Certificate-Based Key Management*, RFC 1423, que contiene la *Part III: Algorithms, Modes and Identifiers*, y la RFC 1424, en la que se recoge la *Part IV: Key Certificates and Related Services*.

⁷ A principio de los noventa empezaron los vientos (vientos que no sólo no han dejado de soplar en la actualidad, sino que soplan aún con mayor fuerza) tendentes a desplazar los foros de decisión desde una centralidad universal, participada por todos los países, hacia opciones unipolares. Una consecuencia de esa actitud centrípeta (en lo que a Telemática se refiere) ha sido el desprestigio programado de todos los protocolos desarrollados en ámbitos de titularidad internacional en los que todos los países tengan las mismas oportunidades para opinar (al

menos en lo regulado por sus estatutos). Y esto afecta tanto a ITU como a ISO, cuya legitimidad proviene de la misma ONU. (Habida cuenta de que la ONU es una pálida sombra de lo que fue, es fácil deducir el estado de salud de los organismos que de ella se derivan.) Pero eso no significa que la ONU y estos organismos no sean de utilidad. Antes al contrario, ahora más que nunca (cuando cada vez son más las parcelas de la vida diaria que se ven afectadas por la expansión de las redes telemáticas) son necesarios esos organismos internacionales de normalización ante los que la ciudadanía, directamente o a través de sus representantes, pueda exigir soluciones y responsabilidades.

⁸ La Infraestructura MISSI, *Multilevel Information Systems Security Initiative*, fue elaborada por la NSA, *National Security Agency*, para regular las comunicaciones en las distintas dependencias del Gobierno Federal de los Estados Unidos. Como en otras ocasiones, la iniciativa partió del Departamento de Defensa en cuyo ámbito se requiere el trasiego de mensajes con distintos niveles de secreto, lo que conlleva la definición de políticas de seguridad muy estratificadas.

⁹ Una de las grandes ventajas de la definición de la versión 3 del certificado consiste en que, además de las extensiones estándar, en cada dominio de seguridad concreto se pueden definir nuevas extensiones que ayuden a llevar a cabo la política de seguridad que allí se haya establecido. En [LoCa97] se define una nueva extensión al Certificado X.509 que contiene la secuencia de identificadores de CAs y de certificados que permitan obtener con fiabilidad, en cada caso, el camino de certificación y el punto de confianza en la comunicación segura entre dos entidades. En [Lope98] se justifica mediante múltiples ejemplos la utilidad de una extensión de este tipo en determinados entornos de comunicación.

¹⁰ La primera publicación del Certificado de Atributos fue llevada a cabo por ANSI, que definió la versión 1 de esta estructura. Posteriormente fue creada la versión 2 mejorando el campo de identificación de la entidad propietaria. En 1997 ITU-T incluye esta estructura de datos dentro de la Recomendación X.509. En la revisión de la X.509 del año 2000 se perfecciona su definición resolviendo insuficiencias de las especificaciones anteriores. Tanto ITU-T como ANSI y el IETF han trabajado en el desarrollo de extensiones normalizadas y atributos para incluirlos en la definición de este certificado. Dentro del entorno del IETF el documento del PKI-WG *Internet X.509 Public Key Infrastructure: Roadmap* [ADT02] y la RFC 3281 (*An Internet Attribute Certificate Profile for Authorization*) aclaran su utilización y significado, y detallan su estructura.

¹¹ En el entorno de ISO se ha desarrollado la norma ISO/IEC 18014 *Time-stamping services* dividida en dos partes (*Part 1: Framework; Part 2: Mechanisms producing independent tokens*) más una tercera (*Part 3: Mechanisms producing linked tokens*) sobre cuya formulación no existe mucho consenso. El ETSI (*European Telecommunications Standard Institute*) dentro de los trabajos conducentes a especificar la *firma electrónica* (que se estudiará en el próximo capítulo) ha desarrollado una norma sobre perfiles, TS 101 861 *Time Stamping Profile*, y otra, TS 101 023, sobre requisitos de las TSAs. Dentro del IETF se ha desarrollado una norma sobre los protocolos a utilizar, RFC 3161 *Internet X.509 Public Key Infrastructure. Time-Stamp Protocol (TSP)* y se ha generado un documento *Policy Requirements for Time-Stamping Authorities* [PPR03] (técnicamente equivalente a la TS 101 023 de ETSI) que discute los conceptos contenidos en todos los anteriores y analiza los requisitos para una política de seguridad sobre estos servicios, razón por la cual lo veremos en el Capítulo 10.

CAPÍTULO

8

ESCENARIOS DE FIRMA Y AUTENTICACIÓN EN REDES TELEMÁTICAS

Tanto las máquinas que prestan los servicios como las personas que los usan tienen en las redes telemáticas una presencia puramente virtual. Solamente son reconocidos por los mensajes que emiten, en los que dicen quiénes son y qué información suministran o solicitan. Esta ausencia de presencia física hace que exista un grave riesgo de suplantación de personalidad y que, consecuentemente, sea necesario proveer servicios de autenticación adecuados a cada circunstancia.

La firma digital es uno de los elementos clave que pueden ser utilizados para estos intentos de asegurar la identidad de las entidades comunicantes. Llevamos siglos y siglos aceptando la validez de la firma caligráfica para el establecimiento de acuerdos de todo tipo, tanto de orden político como económico y social. Por ello, la firma digital es un mecanismo que pretende (ventajosamente, como veremos) jugar en el ciberespacio un papel equivalente al que hasta la fecha ha jugado la firma manuscrita en las comunicaciones convencionales a través de papel.

Ya en el Capítulo 5 se presentó el formato de la firma digital. Además, para la definición del certificado X.509, discutida con bastante detalle en el Capítulo 6, hemos venido haciendo cumplido uso de este mecanismo. En el presente capítulo se abordará con suficiente detalle este tema dentro del contexto de los protocolos de autenticación. Podríamos haber optado por incluir en el Capítulo 5 los algoritmos básicos de firma, al lado de los restantes algoritmos criptográficos asimétricos, y dejar para éste sólo lo relativo a los escenarios de comunicación en entornos PKI (utilidad de las distintas TTPs, uso de los certificados, reglamentación legal, etc.). No obstante, se ha optado por agrupar todo lo relacionado con este tema en el presente capítulo para obtener una visión de conjunto, porque tan importante o más que el mecanismo de firma propiamente dicho es el escenario telemático en el que se manifiesta.

8.1. GENERALIDADES

Como ya se definió, el servicio de autenticación (*authentication*, en inglés) sirve para garantizar que una entidad comunicante *es quien dice ser*. En la literatura puede

encontrarse también bajo la denominación de «servicio de autenticación», ya que ambos términos son perfectamente válidos en español, aunque «autenticación» es de origen más antiguo y más auténtico (y más breve).

Dada la importancia que en el ámbito de las redes telemáticas tiene garantizar la identidad de las entidades que participan en la comunicación, este servicio de seguridad es de los más necesarios. Además, otros servicios (y de forma especial el de control de acceso) dependen de la autenticación, en gran número de circunstancias, para llevar a cabo la protección que tienen encomendada.

Dependiendo del contexto en el que se provea este servicio, cabe hablar de dos situaciones distintas:

1. **Autenticación de entidad** (*Entity authentication*). Una determinada entidad involucrada en una conexión telemática tiene que demostrar su identidad ante otra entidad o sistema. En este caso cabría distinguir a su vez entre *autenticación unidireccional*, en cuyo caso sólo uno de los participantes en la comunicación está obligado a demostrar su identidad y *autenticación mutua*, en la que la operación de aseguramiento de la identidad de las entidades se realiza en ambos sentidos de la comunicación.
2. **Autenticación del origen de los datos** (*Data origin authentication*). Se produce cuando, en una transferencia de datos, como puede ser el caso de un mensaje de correo electrónico, sólo se requiere demostrar que los datos en cuestión han sido originados por una determinada entidad.

AUTENTICACIÓN DE ENTIDADES

En general, podríamos decir que la autenticación de una persona o entidad comunicante puede llevarse a cabo apoyándose en diversas características, entre las que podríamos destacar:

- a) Por algo que sólo conocen la persona o entidad que trata de autenticarse y la entidad que verifica la autenticación. Podría tratarse de alguna palabra de paso, de alguna clave criptográfica simétrica o de algunos datos personales.
- b) Por algo que sólo quien trata de autenticarse posee. Puede tratarse de algún dispositivo físico o de un determinado secreto, por ejemplo una determinada clave privada almacenada en una tarjeta inteligente. En este último supuesto, cuando usa su firma digital, la entidad que se autentica sólo demuestra la posesión del secreto, no siendo necesario compartirlo con la entidad que verifica la autenticación. Algo similar ocurre con las técnicas de *conocimiento cero* (*zero-knowledge*), mediante las cuales alguien demuestra, respondiendo a ciertas interrogaciones, que está en posesión de un secreto sin necesidad de mostrar parte alguna de él.
- c) Por algo que la persona sabe hacer. Este sería el caso, por ejemplo, de la firma manuscrita o alguna otra habilidad exclusiva de la persona que trata de autenticarse. Existen periféricos que, mediante un lápiz conectado al ordenador, permiten obtener una imagen electrónica de la firma caligráfica que el usuario realice ante la entidad que verifica la autenticación (se puede reproducir no sólo la forma y la presión sobre el papel, sino la velocidad del trazado).
- d) Por cómo es. La persona que trata de autenticarse exhibe determinadas características biométricas que le son inmutablemente propias. La huella digital o

la forma del iris del ojo son dos ejemplos destacados de este tipo de mecanismos.

Por otra parte, en las redes telemáticas existen dos situaciones en las que resulta necesaria la autenticación de entidades:

- Una de ellas es la que se presenta habitualmente: cuando una entidad comunicante necesita acreditar su identidad ante otra entidad ubicada en un punto remoto de la red telemática. En este caso, sólo las características marcadas como a) y b) en la relación anterior resultan de interés, ya que las otras dos, por su propia naturaleza, se prestan poco a este tipo de reconocimientos.
- La otra situación que se presenta es cuando una persona trata de acceder a uno de los terminales de la red. Éstos pueden ser todo tipo de máquinas o de agentes, incluyendo desde ordenadores protegidos a tarjetas inteligentes (como ya se comentó en el Capítulo 2). En estos casos, suelen ser de utilidad mecanismos basados en las cuatro características antes reseñadas, si bien es verdad que los que utilizan habilidades del tipo de las recogidas en c) van teniendo cada vez menos predicamento.

AUTENTICACIÓN SIMPLE, AUTENTICACIÓN MEDIANTE DESAFÍO Y AUTENTICACIÓN FUERTE

En la recomendación ITU-T X.509, denominada «Marco de Autenticación» (*Authentication Framework*), se definen, además del certificado y las CRLs, dos esquemas básicos de autenticación: autenticación simple (también llamada *débil*) y autenticación fuerte. La primera se corresponde con los mecanismos en los que se usa una contraseña o palabra de paso (*password*) más o menos protegida, mientras que la segunda se refiere a escenarios en los que se usan los certificados de las claves públicas asignadas a las entidades que se desea autenticar.

El esquema más sencillo de **autenticación simple** mediante contraseña protegida consiste, Figura 8.1, en que la entidad A, que trata de probar su identidad, envía a la entidad B (que es la que verifica la autenticación) su nombre identificativo y un testigo de autenticación, T. Este testigo se construye a partir de su nombre y de la contraseña o password, *pass-A*, todo ello protegido mediante la aplicación de una función unidireccional f_1 . De forma que:

$$T = f_1(A, \text{pass-A})$$

Como ya adelantamos en el Capítulo 5, una *función unidireccional* es una función que se define entre dos conjuntos X e Y , $f: X \rightarrow Y$, de tal manera que el cálculo de $y = f(x)$ es «fácil» de computar, pero el inverso $x = f^{-1}(y)$ es computacionalmente muy «difícil» de hallar. Más adelante, cuando se aborde el tema de las funciones resumen o funciones *hash*, se insistirá de nuevo en este concepto.

Para que el esquema funcione es necesario que el nombre identificativo sea enviado también en claro y que la entidad B conozca y tenga almacenada la contraseña de A. Para evitar ataques de terceros que puedan llegar a conocer esa contraseña, lo más razonable es que B la tenga almacenada previamente protegida por la función unidireccional que se esté usando, es decir, que lo que almacene sea el valor del testigo de autenticación. En la figura se ha denominado T' a este valor, queriendo significar que hasta que no compruebe que el valor T recibido coincide con el valor

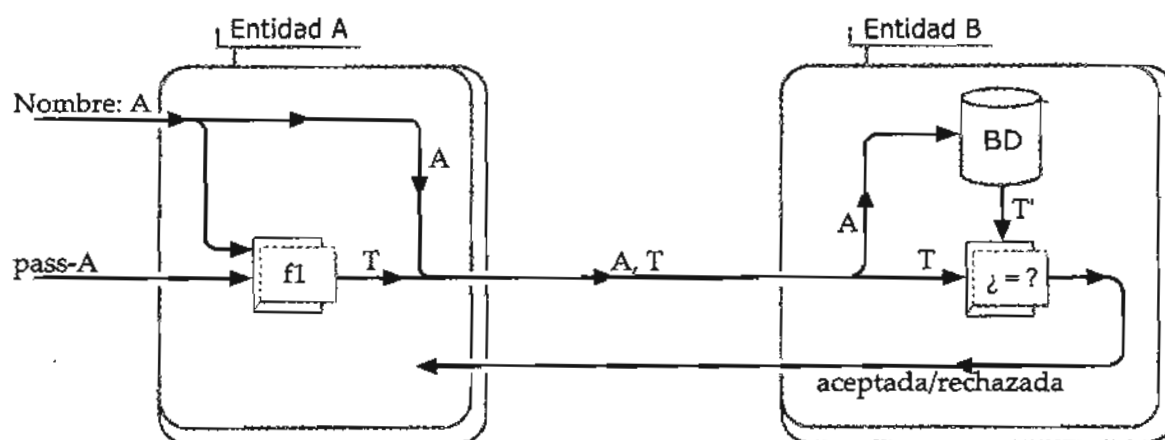


Figura 8.1. Autenticación simple mediante contraseña.

que tiene almacenado, no dará por válido el proceso de autenticación. Para ello se requiere que el algoritmo a utilizar en la función unidireccional esté previamente fijado y concertado entre las dos entidades comunicantes.

La simpleza de este tipo de autenticaciones viene determinada por el hecho de que este esquema es muy vulnerable ante ataques de terceros. Por ser la contraseña una información que ha de elegir el usuario, se producen riesgos debidos a la inadecuada selección de ésta, lo que puede hacer que un atacante avisado pueda imaginarse su valor mediante comprobaciones rutinarias. Asimismo, al ser necesario conservarla secreta durante bastante tiempo, se producen riesgos notables motivados por descuidos en su custodia. Aparte de estos inconvenientes, que pueden corregirse mediante una adecuada política de seguridad que se imponga en el dominio de que se trate, existen otros riesgos derivados de las capturas de datos en la línea que puede realizar un usuario malicioso.

En efecto, un atacante que haya podido capturar el intercambio de información producido en un esquema similar al recogido en la Figura 8.1 puede, tiempo después y contando con que A no haya modificado su password, conectarse con B y mostrarle como identificador el valor $f(A, \text{pass-A})$ que había capturado antes, aunque siga desconociendo el valor de pass-A . Como el testigo sigue siendo válido, B daría por buena la autenticación, con lo que se habría perpetrado un ataque exitoso de suplantación de personalidad.

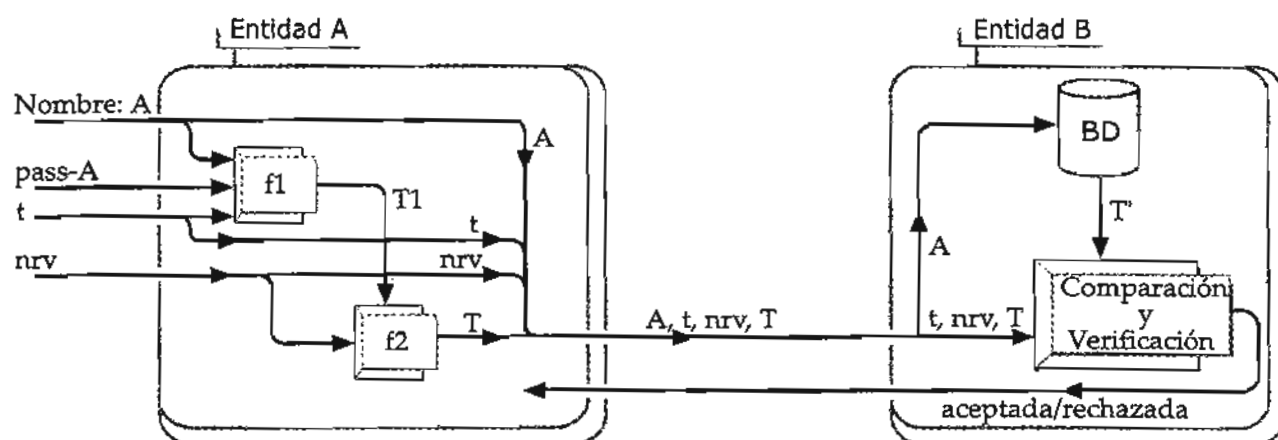


Figura 8.2. Autenticación simple protegida.

Para resolver este problema se propone lo que la X.509 llama «autenticación simple protegida», consistente en poner en juego otros dos parámetros adicionales: un indicador de tiempo y/o un número no repetitivo (conocido también como *nrv*, *non-repeating value*). En la Figura 8.2 se presenta una posible configuración de este tipo de soluciones. El indicador de tiempo, t , previene contra ataques de repetición que pretendan llevarse a cabo cierto tiempo después. De forma análoga, si B almacena los valores de *nrv* que han sido utilizados por A en anteriores conexiones, puede estar en condiciones de rechazar un intento de autenticación fraudulento si comprueba que dicho número ya ha sido utilizado previamente. En el esquema presentado en la Figura 8.2 se ha incluido una segunda función unidireccional, f_2 , que refuerza la protección buscada. Bajo esta misma idea existen diversas combinaciones que pueden incluir más de un indicador de tiempo y más de un *nrv*. En el caso representado, mediante la función f_1 se genera una primera pieza protegida, T_1 , que actúa, junto al *nrv*, como entrada de la segunda función unidireccional que es la que genera el testigo de autenticación que se envía a la entidad remota. De tal forma que:

$$T_1 = f_1(A, \text{pass-A}, t) \\ T = f_2[nrv, f_1(A, \text{pass-A}, t)]$$

La **autenticación por desafío** no deja de ser, en realidad, una variante de la autenticación simple o débil estudiada en los párrafos precedentes. Se basa, al igual que en los casos antes vistos, en el hecho de que tanto la entidad que trata de autenticarse como la entidad que verifica la autenticación están en posesión de una misma información secreta que comparten. En la Figura 8.3 se representa el esquema simplificado del intercambio de información que se produce en este tipo de protocolos. La entidad A envía a B una petición de autenticación y la entidad B le responde con un *reto* o *desafío* que no es sino una pregunta especializada cuya respuesta sólo la entidad A conoce. Bajo este esquema funcionan protocolos de lo más variopinto, cada uno con niveles de protección muy diferentes. Uno de ellos, de gran debilidad porque la mayoría de las veces la información circula en claro, consiste en que la entidad B pregunta, por ejemplo, por el dato que corresponde a un par fila-columna dentro de una matriz de números, de dos dimensiones, que ambas entidades comparten. También bajo este mismo esquema pueden establecerse escenarios más robustos en los que el desafío está constituido por un *nrv* y la respuesta (que depende, naturalmente, de dicho parámetro) está protegida mediante funciones unidireccionales.

Modelos similares de autenticación simple y de desafío pueden establecerse utilizando criptosistemas simétricos para cifrar las contraseñas y las respuestas, de tal forma que las dos entidades que participan en el proceso de autenticación compartan

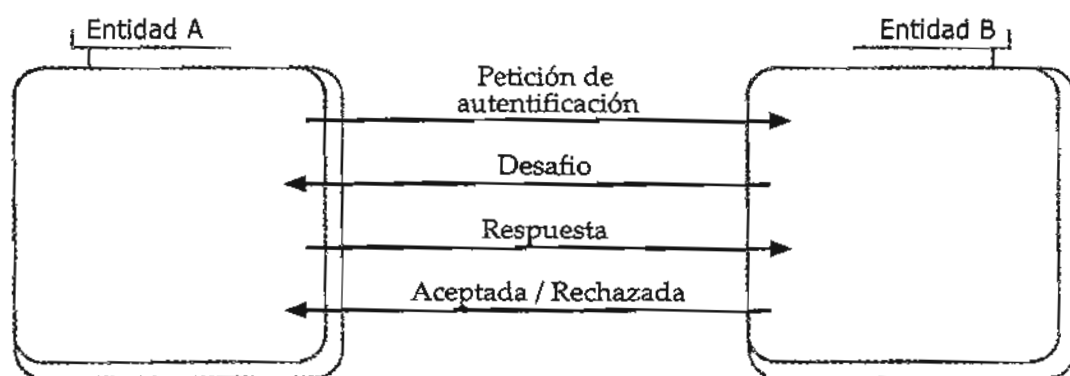


Figura 8.3. Autenticación mediante desafío y respuesta.

una misma clave secreta. En cualquier caso, la autenticación basada en contraseña o password resulta manifiestamente insuficiente para la mayoría de las aplicaciones telemáticas que exijan un nivel de seguridad aceptable, por lo que será necesario recurrir a la denominada **autenticación fuerte**, que se basa, sustancialmente, en que cada una de las entidades participantes en el escenario de autenticación está en posesión de una clave privada, que mantiene en secreto, especialmente destinada a este fin. La forma que tiene una entidad B de comprobar la identidad de una entidad A, copartípe en una comunicación, es corroborar que A está en posesión de esa clave privada. Para ello, B puede hacer uso del correspondiente certificado X.509 de A, una vez haya recibido un testigo cifrado con la clave privada de esa entidad. Es importante destacar que, en este caso, la entidad que se autentica sólo necesita demostrar la posesión del secreto, no siendo necesario compartirlo en modo alguno con la entidad que verifica la autenticación. Se pueden distinguir los siguientes casos de autenticación fuerte:

1. Autenticación unidireccional

Como su nombre sugiere, comporta una sola transferencia de información entre la entidad A, que trata de autenticarse, y la entidad B, que será la que verifique esta autenticación. Pueden presentarse múltiples variantes cuando se trata de fijar cuáles son los elementos que deben componer esa información, aunque en todos ellos debe existir, como mínimo, alguna información cifrada con la clave privada de A y algunos parámetros que protejan contra la reutilización fraudulenta de los testigos de autenticación. Una de esas alternativas puede ser que la información de autenticación enviada sea:

$$A \rightarrow B: \quad A, t_A, nrv_A, B, A_{\text{sig}} [t_A, nrv_A, B], CA\langle\langle A \rangle\rangle$$

En la Figura 8.4 se representa el intercambio de información que se produce entre ambas entidades. A es el nombre que sirve para identificar a la que trata de autenticarse (puede ser su nombre X.500 u otro de los formatos de nombre que hemos visto en la v3 del certificado X.509). El parámetro t_A es una marca de tiempo que puede ser una fecha simple o estar compuesta por una tupla que defina su periodo de vigencia, y nrv_A es un número no repetitivo generado por A. Estos dos parámetros sirven para evitar ataques de reutilización de testigos de autenticación. También se envía el nombre identificativo de B para reflejar que el testigo de autenticación ha sido generado exclusivamente para ser validado por B. En la figura se ha denominado p a la información constituida por la concatenación de estos tres últimos parámetros. Este conjunto de parámetros p se envía firmado por A, es decir, viaja en claro y se le añade la correspondiente firma digital. Por último, A puede enviar también su certificado, aunque este detalle no ha sido incluido en la figura, en la que sí se refleja la posibilidad que ambas entidades tienen de recuperar los certificados que necesiten accediendo a un servidor especializado.

Con esta información, B está en condiciones de comprobar tanto la autenticidad de la identidad de A como que el intento de autenticación es original y no una copia fraudulenta de otro realizado anteriormente. Las dos piezas fundamentales de este proceso son la información firmada por A y el certificado generado por la correspondiente CA, que permite a B obtener el valor de la clave pública de A para verificar la firma (tras comprobar la validez de este certificado y el hecho de que no ha sido revocado).

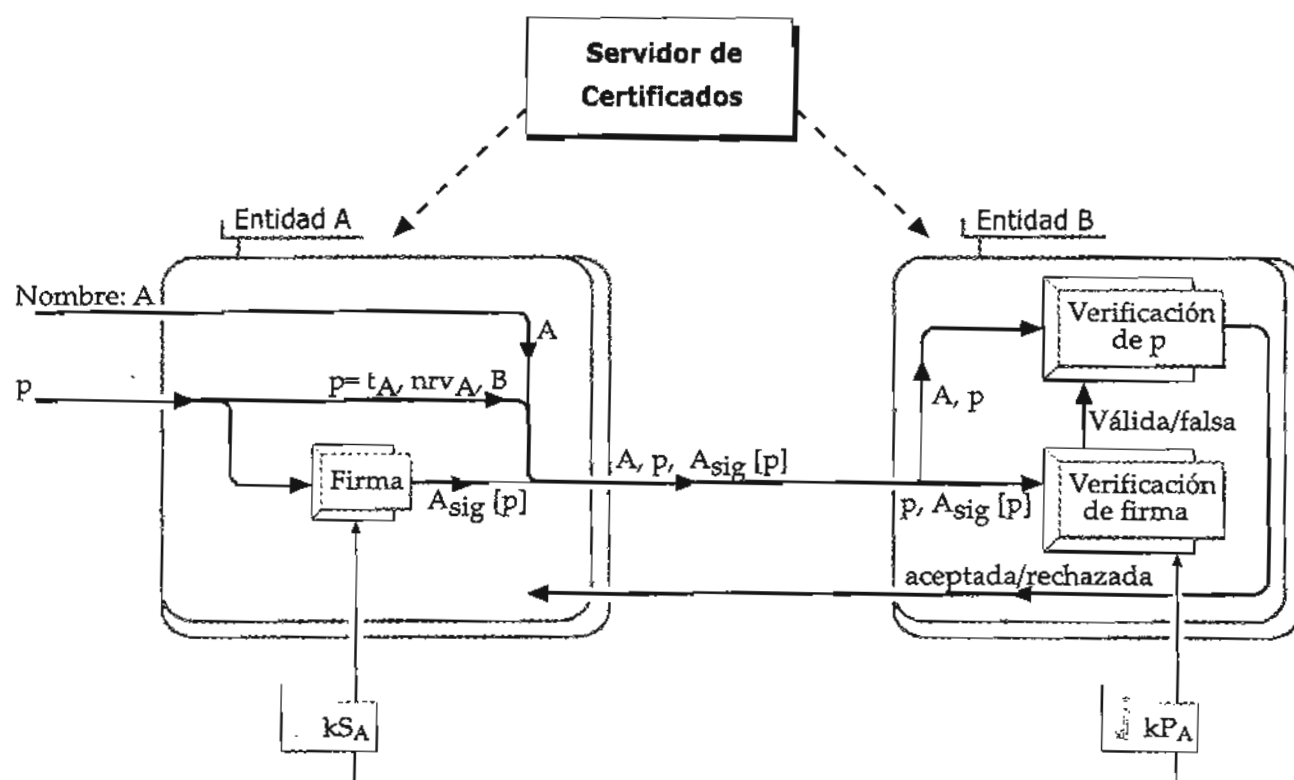


Figura 8.4. Autenticación fuerte unidireccional.

Si B confía en la CA que ha generado el certificado $CA\langle\langle A \rangle\rangle$, el proceso de verificación de la autenticación resulta casi inmediato. Si ambas entidades perteneciesen a un dominio de seguridad complejo en el que existiesen múltiples CAs sería necesario añadir el camino de certificación en el testigo de autenticación (como propone la norma X.509) o establecer en B los procedimientos adecuados conducentes a su averiguación. Algo más complejo sería el caso en el que las dos entidades pertenezcan a dominios de seguridad distintos en los que estén definidas políticas de seguridad diferentes entre sí, ya que en este caso podría ser necesario comprobar, mediante la información aportada por las extensiones del certificado, que ambas políticas permiten el establecimiento de la cooperación que se está tratando de llevar a cabo.

2. Autenticación bidireccional

Una forma de conseguir la *autenticación mutua* es añadir una respuesta de B al proceso de autenticación unidireccional antes descrito. Esta respuesta podría consistir en:

$$B \rightarrow A: \quad t_B, nrv_B, A, nrv_A, B_{sig} [t_B, nrv_B, A, nrv_A], CA\langle\langle B \rangle\rangle$$

Cuando A recibe esta información firmada por B, puede comprobar que es una respuesta al testigo de autenticación enviado en la primera fase, que el indicador de tiempo t_B refleja la necesaria «frescura» de la respuesta y que nrv_B no ha sido utilizado anteriormente por B en otra conexión.

3. Autenticación tridireccional

A veces, para reforzar la autenticación mutua, puede añadirse a los dos intercambios antes descritos una tercera transferencia de datos entre A y B. Aunque en este caso pueden ahorrarse los indicadores de tiempo t_A y t_B , el proceso es más engorroso y no aporta sustanciales mejoras a la autenticación mutua bidireccional, que resulta suficientemente fuerte para la mayoría de las aplicaciones telemáticas donde se requiere este servicio.

Conviene insistir en que los esquemas de autenticación fuerte que acabamos de presentar admiten múltiples variantes en cuanto al número de parámetros presentes en la información que se intercambian ambas entidades, manteniendo, eso sí, las características fundamentales que hemos expuesto.

AUTENTICACIÓN DEL ORIGEN DE LOS DATOS: MAC, HASH Y FIRMA

De la separación que en estas notas introductorias estamos haciendo entre la autenticación de entidades y la autenticación del origen de los datos podría deducirse que ambos procesos son cosas disjuntas y totalmente disociadas. Nada más lejos de la realidad: con frecuencia, en los procesos de autenticación descritos más arriba se incluyen elementos que permiten reconocer el origen de determinados datos que pueden insertarse en el intercambio de información definido.

Lo que ocurre es que en determinadas circunstancias, como es el caso del correo electrónico seguro o la transferencia de ficheros, no se desea el establecimiento de una conexión virtual entre dos entidades, sino solamente asegurarse de que los datos proceden realmente de la entidad que se reclama de ello. En estos casos sólo es necesario añadir un testigo autenticador en el mensaje que se está transfiriendo.

Uno de estos posibles autenticadores es el denominado *código de autenticación de mensaje*, **MAC** (*message authentication code*), que consiste en una pieza de información de tamaño fijo generada mediante un criptosistema de clave secreta. En el siguiente epígrafe se estudiará con cierto detalle tanto su utilización como su estructura y seguridad.

La **firma digital** de un determinado mensaje constituye un testigo autenticador de extraordinaria fortaleza. Ya se ha hecho uso de esta cualidad cuando se ha definido el certificado, firmado por una Autoridad de Certificación, como una pieza de información segura en sí misma. En los escenarios donde está presente una infraestructura de clave pública, PKI, el certificado juega un papel fundamental en los procesos de autenticación. Más adelante, en este mismo capítulo, se estudian los distintos algoritmos existentes y los condicionantes sociales y legales que se imponen en su uso.

Como adelantábamos en el Capítulo 5, un **resumen** o **valor hash** es un elemento de información de tamaño fijo, generado mediante una función unidireccional, que también puede ser utilizado como autenticador de mensajes. No obstante, el verdadero interés de las funciones hash radica en que son parte fundamental en la estructura de los algoritmos de firmado. Por esa razón se estudian con cierto detalle en el epígrafe 8.3 como paso previo antes de abordar en toda su extensión el tema de la firma.

8.2. AUTENTICACIÓN MEDIANTE CRIPTOSISTEMAS DE CLAVE SECRETA

La criptografía de clave secreta puede ser también utilizada para la provisión del servicio de autenticación y, por consiguiente, de los restantes servicios que de él dependen. Lo que ocurre es que, como ya hemos discutido, existe una dificultad insalvable en lo que se refiere a la distribución de claves simétricas. Debido a ello, el uso de protocolos basados exclusivamente en criptosistemas de este tipo tiene un interés reducido y decreciente. No obstante, como veremos en los párrafos que siguen, existen soluciones basadas en criptografía de clave secreta que garantizan la identidad de las entidades participantes en la comunicación, aunque la robustez y seguridad que con ellas se obtiene no llega, ni de lejos, a lo que se puede conseguir con el uso de la firma digital.

CÓDIGOS DE AUTENTICACIÓN DE MENSAJES (MAC)

Un testigo MAC es una pieza de información, de tamaño fijo, generada mediante un proceso de cifrado con clave secreta, que se añade, como un apéndice, al mensaje cuyo origen se quiere autenticar. En la Figura 8.5 se representa el esquema de comportamiento de este mecanismo. La entidad A tiene un mensaje m que desea enviar a B de forma que se garantice la autenticidad del origen de los datos. Para ello genera un testigo, el código MAC, mediante un proceso de cifrado basado en un algoritmo de clave secreta que genera, a la salida, una pieza de información de tamaño fijo, independientemente de lo grande que sea el mensaje m . Este testigo, junto con el mensaje en claro, es enviado a la entidad destinataria. En la figura se ha denominado m' al mensaje en claro que viaja por la red y recibe la entidad B para resaltar que, hasta que la entidad receptora no realice la comprobación, mantiene la duda de si este mensaje m' es realmente el mensaje m que envió A o es, por contra, una falsificación generada por un usuario malicioso (o por el azar).

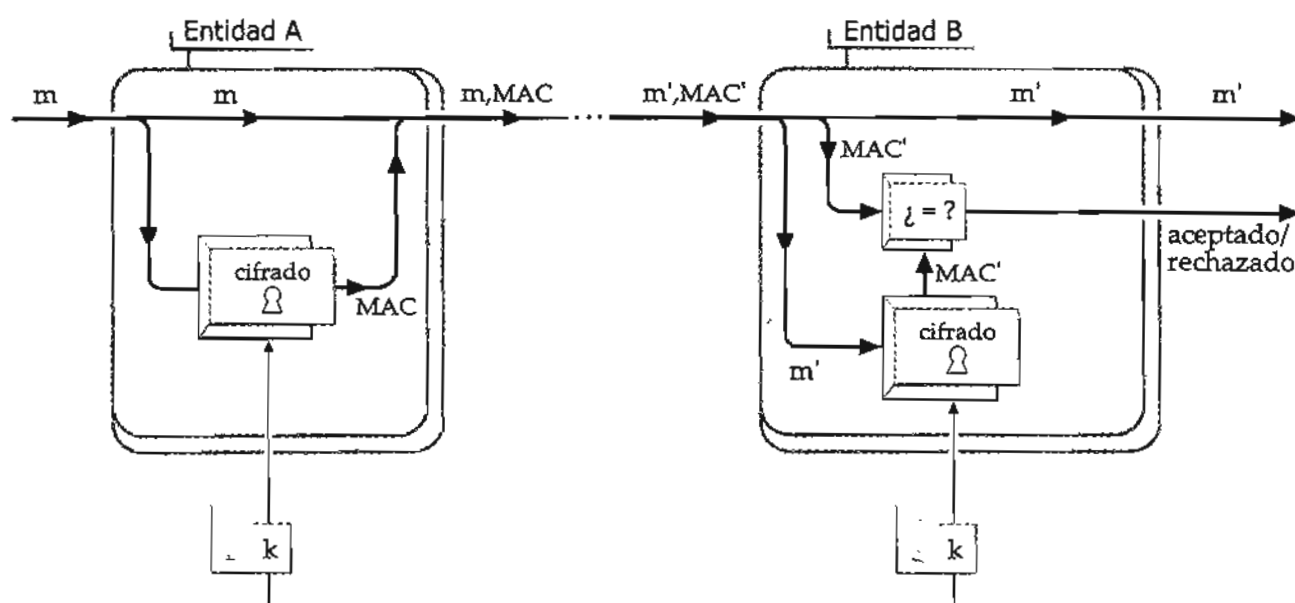


Figura 8.5. Utilización del MAC.

La entidad receptora comparte con A una misma clave secreta (k). Cuando recibe el mensaje, lo cifra, a su vez, con el mismo algoritmo y con la misma clave, generando un nuevo MAC (que en la figura se ha representado como MAC') y establece la comparación con el testigo recibido. Si ambos son iguales, adquiere una razonable certeza de que el mensaje procede de A, ya que esta entidad, y sólo ella, comparte con B esa clave k concreta. De forma añadida, se asegura de que el mensaje que le ha llegado coincide con el mensaje que A envió, ya que si algún atacante hubiese modificado su contenido, al calcular el MAC' en el extremo receptor, éste no habría coincidido con el generado por la entidad A. Es decir, se proveen los servicios de integridad y de autenticación del origen de los datos.

Como puede observarse, hay cierta semejanza formal entre este esquema y los de autenticación débil que presentamos más arriba. La diferencia fundamental entre ambos es que en este caso no existe respuesta hacia el originador del mensaje porque no se trata del establecimiento de una conexión autenticada sino, simplemente, de adquirir garantías respecto al mensaje recibido.

En los protocolos que utilizan MAC también es necesario añadir información acerca de la entidad A de que se trata, debido a que cuando B reciba el mensaje, tendrá que conocer de forma inmediata la identidad de A, para seleccionar, de entre todas las que guarda en su base de datos, la clave secreta k que con ella comparte. Igualmente, para asegurar la integridad de un mensaje que le llega dividido en varias partes será necesario que estos segmentos estén debidamente numerados de forma secuencial. Este es el caso de las PDUs que transportan una determinada SDU (unidad de datos del servicio) en los protocolos del bloque de transporte.

Podría pensarse, con toda razón, que para conseguir los servicios de autenticación y de integridad del origen de los datos no habría sido necesario recurrir a la generación de un testigo MAC mediante un criptosistema de clave secreta, ya que cifrando todo el mensaje con esa misma clave se habría conseguido el mismo efecto, con el añadido de que ganaríamos un servicio más de seguridad: el de confidencialidad de los datos. Esta crítica se acentúa más si tenemos en cuenta que para cifrar el mensaje en recepción, para así poder obtener el testigo, es necesario procesar **todo** el mensaje, al igual que cuando se trata de descifrar un mensaje encriptado. La razón por la que se propuso este mecanismo como alternativa al encriptado completo del mensaje es que, algunas veces, por exceso de carga en el sistema, es necesario procesar con cierta celeridad el mensaje recibido, reservando para determinadas muestras aleatorias la comprobación de la integridad y autenticidad. Otras veces, de lo que se trata es que un mismo mensaje es enviado a un número elevado de usuarios y sólo a un reducidísimo grupo de ellos le corresponde la responsabilidad de realizar las comprobaciones criptográficas. (Como se puede comprobar, blandas razones.)

Otra crítica importante a este mecanismo es que cuando B reciba un determinado mensaje de A, en el caso de que A se niegue a reconocer que envió ese mensaje, la entidad B no está en condiciones de presentar prueba alguna ante terceros, ya que, siendo ella poseedora de la clave con la que fue cifrado el MAC, bien pudiera haber sido ella misma la que hubiese creado ese mensaje y ese testigo. Esta notabilísima carencia hace que los autenticadores MAC sean inútiles a la hora de implementar servicios de no-repudio, que tan importante papel juegan en la implantación de los servicios de seguridad que necesitan los ciudadanos para proteger sus comunicaciones.

Es fácil completar el esquema recogido en la Figura 8.5 para añadir el servicio de confidencialidad: basta con cifrar con otra nueva clave secreta el mensaje resultante de la concatenación de m y del MAC, actuando de forma simétrica en el extremo receptor.

SEGURIDAD DE LOS ALGORITMOS MAC

Lo primero que hay que hacer notar es que los algoritmos de generación de códigos MAC no son algoritmos criptográficos simétricos convencionales, ya que no son reversibles, es decir, no es factible obtener el mensaje a partir del MAC ni siquiera poseyendo la clave secreta.

Ello no quita para que la mayoría de los algoritmos propuestos se basen en modificaciones de esos mismos algoritmos simétricos. Este es el caso del denominado DAC, el más ampliamente utilizado, que consiste simplemente en identificar el MAC con el último de los bloques de cifrado que se obtienen cifrando el mensaje m con un algoritmo DES operando en modo encadenado, CBC. Como se analizó en el apartado 4.2 y se recoge en la Figura 4.3, consiste en que el resultado de cifrar el primer bloque se suma módulo 2 con el segundo bloque del mensaje, de tal forma que el resultado de esta operación XOR constituye la entrada de la operación de cifrado que da lugar al segundo bloque del criptograma, y así sucesivamente, hasta procesar el último de los bloques. En un cifrado normal con DES el criptograma final está constituido por el conjunto de todos los bloques resultantes del cifrado, mientras aquí sólo el último bloque constituye la salida.

Dependiendo de cómo esté configurado, el MAC puede estar formado por los 64 bits del bloque final o solamente por un subconjunto de éste, consistente en los N primeros bits del bloque, donde N debe ser igual o mayor que 16.

Aunque, como se deduce de los párrafos precedentes, los mecanismos MAC no presentan excesivo interés de cara a la búsqueda de soluciones para la protección de las redes telemáticas, conviene detenerse un poco en el estudio de su seguridad. La resistencia ante un ataque por fuerza bruta de un algoritmo MAC hay que contemplarla desde un esquema distinto al que se corresponde con un posible ataque a un texto cifrado con un algoritmo simétrico convencional. Para centrar la discusión supongamos que:

- M es el número de bits del mensaje m que se procesa.
- N es el número de bits del testigo MAC.
- k es el número de bits de la clave secreta.
- n es el número de bits del bloque.

Por ello, existirán 2^M posibles mensajes de M bits y 2^N posibles códigos MAC diferentes. Como cabe suponer que M es mucho mayor que N , y como, además, M no es un número fijo y predeterminado, existirán muchísimos más mensajes que códigos disponibles. Es decir, dos o más mensajes darán lugar a un mismo código MAC, con lo que existe la posibilidad, al menos en teoría, de encontrar otro mensaje distinto de m que dé lugar al mismo código, lo que supondría una violación clarísima de la seguridad del mecanismo.

En el caso de un cifrado convencional, para un ataque por fuerza bruta en el que sólo se disponga del mensaje cifrado, será necesario descifrar este criptograma aplicando cada una de las posibles claves, 2^k , hasta obtener un mensaje que convincentemente pueda parecer que se corresponde con el mensaje original. Es difícil saberlo a ciencia cierta porque puede tratarse de un mensaje truco y sin aparente significación.

En el caso del MAC el atacante conoce el mensaje en claro, por lo que debería cifrar dicho mensaje con todas las claves posibles, 2^k , hasta obtener un resultado que coincida con el MAC que ha capturado. Si, como suele ser habitual, k es mayor

que N , por existir solamente 2^N códigos, no sólo encontrará una coincidencia, sino muchas, y no podrá saber con ese ataque cuál de ellas es la buscada. Necesitará atacar otros nuevos mensajes de los que se haya obtenido un código MAC aplicando la misma clave, para, repitiendo el proceso anterior, ir acotando las coincidencias hasta encontrar la clave buscada. En general, sin entrar en detalles que dependerían mucho de los tamaños relativos de las distintas piezas de información puestas en juego, cabe decir que la seguridad de estos códigos puede ser distinta pero no menor que la de los algoritmos convencionales de cifrado. No obstante, contando con que k sea mayor que N , se estima que el número de bits que debería tener el MAC para obtener una fortaleza razonable debería ser superior a 128, cifra mucho mayor que la puesta en juego con soluciones como DAC.

SERVIDORES DE AUTENTICACIÓN: SISTEMA KERBEROS

Kerberos es un sistema desarrollado en los años ochenta en el Instituto Tecnológico de Massachusetts, MIT, que consiguió proporcionar servicios de seguridad en red utilizando exclusivamente criptografía simétrica. En versiones posteriores se han ampliado la gama de algoritmos empleados y los servicios proporcionados. Lo que aquí nos interesa no es hacer una presentación fidedigna de este sistema sino utilizarlo como ejemplo de cómo puede encadenarse la provisión de servicios de seguridad a partir de la autenticación del usuario en la red a través de un Servidor de Autenticación. En la Figura 8.6 se recoge el esquema global de funcionamiento del sistema para su versión 4.0.

El cliente C conoce una contraseña de acceso al sistema a través de la cual obtiene, mediante una función unidireccional, la clave secreta k que va a permitirle solicitar su entrada en el sistema. En primer lugar, lo que tiene que hacer es establecer una conexión con el Servidor de Autenticación, AS (*Authentication Server*). Este intercambio de información es el marcado con «1» en la Figura 8.6 y se presenta con un poco más de detalle en la Figura 8.7. El cliente envía a AS el siguiente mensaje:

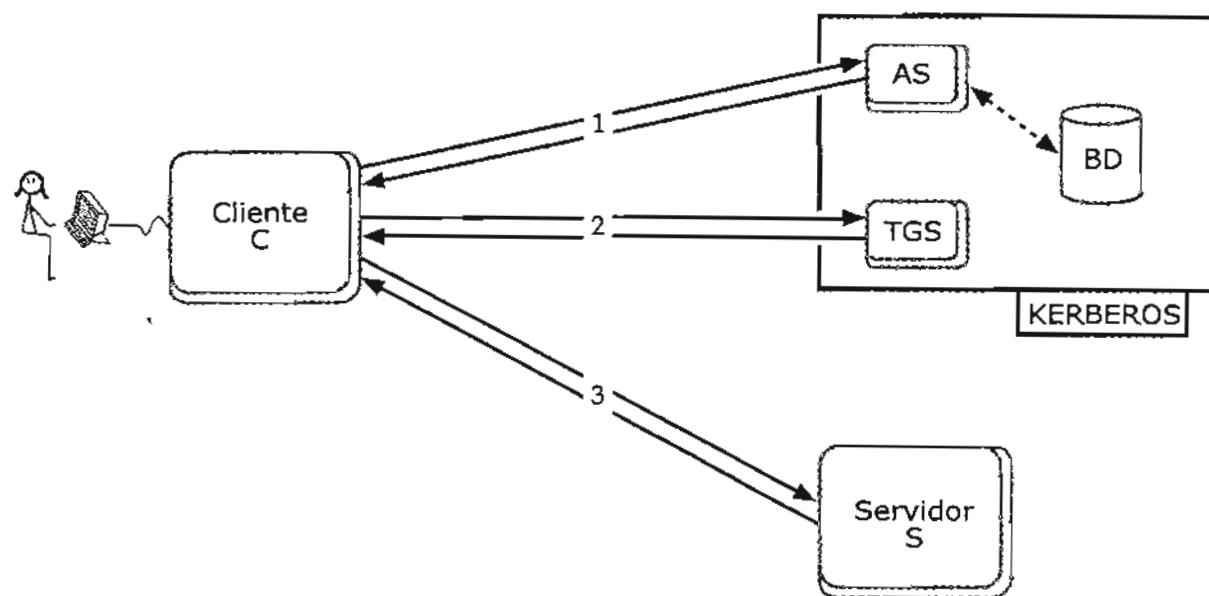
$$C \rightarrow AS: \quad C, tgs, E_k[mt1], mt1$$


Figura 8.6. Arquitectura básica de Kerberos.

Donde C representa su nombre identificativo, $mt1$ es una marca de tiempo usada para evitar reutilizaciones posteriores, que se envía en claro y cifrada con la clave secreta del cliente, y tg es un indicador que significa que lo que está solicitando el cliente es que se le expida una credencial o *ticket* para acceder al servidor de credenciales, denominado TGS (*Ticket Granting Server*). El Servidor de Autenticación y el TGS se encuentran encerrados en un local con fortísimas protecciones físicas y lógicas (este es el verdadero Kerberos, Figura 8.6).

El AS conoce la clave secreta de este y de todos los clientes. Por eso está en condiciones de evaluar el mensaje que ha recibido y, si considera que el cliente es merecedor de ello, le envía la credencial o ticket solicitado para que pueda presentarlo ante el servidor de credenciales (TGS). En este tipo de credenciales es posible establecer categorías en función de la jerarquía del usuario dentro de la red. Le devuelve la siguiente información:

$$AS \rightarrow C: \quad E_k[T_{C-TGS}, k1, tg, mt2]$$

Como puede observarse, esta información está cifrada con la clave secreta k para darle garantías al cliente de que el envío procede del AS. En ella se incluye el ticket solicitado, una clave $k1$, otra marca de tiempo y un identificador que significa que ese ticket es para acceder al TGS. La clave $k1$ es una clave secreta de sesión que permitirá comunicarse al cliente con el TGS y viceversa. La marca de tiempo sirve para que el cliente sepa que se trata de una respuesta a su anterior petición. El ticket está, a su vez, cifrado con una clave secreta (k_{TGS}) ¡conocida por el TGS pero desconocida por el cliente! Es decir, el cliente tiene que presentar ante el servidor de credenciales una información cifrada cuyo contenido desconoce. Dentro de ese ticket se incluye, además de otros datos, el valor de la clave de sesión, $k1$, que le permitirá al servidor responder al cliente.

A continuación, en la interacción entre el cliente y el servidor de credenciales se repite un proceso similar en el que el cliente le indica con qué otro servidor, S en la figura, desea conectarse (una base de datos, servicios de mensajería, etc.). El cliente C se autentica ante el TGS cifrando su identidad con la clave secreta $k1$ que ambos comparten y le entrega el ticket que ha obtenido del AS. Es decir:

$$C \rightarrow TGS: \quad S, T_{C-TGS}, E_{k1}[C, mt3]$$

Como respuesta, recibe una nueva clave secreta de sesión, $k2$, para comunicarse con el servidor S , y un nuevo ticket, T_{C-S} , en el que se indica qué puede hacer (y qué no) cuando pretenda hacer uso de ese servicio concreto. De forma similar al caso anterior, este ticket contiene también la clave de sesión $k2$ y está oscurecido mediante el cifrado con una clave, E_{kS} , que el servidor conoce y el cliente desconoce. Esto es:

$$TGS \rightarrow C: \quad E_{k1}[T_{C-S}, k2, S, mt4]$$

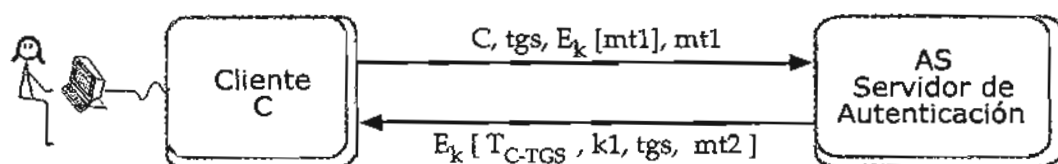


Figura 8.7. Acceso al Servidor de Autenticación.

El tercer paso que se indica en la Figura 8.6 representa, por último, la comunicación que se establece entre el cliente y el servidor telemático propiamente dicho, para lo cual se identificará a través de la clave k_2 que ha recibido en el paso anterior y entregará el ticket T_{C-S} .

Como puede apreciarse, este sistema constituye un refinamiento de los servidores centrales de claves que se estudiaron en apartado 6.1 al discutir el tema de la distribución en red de claves simétricas. El sistema funciona y es de ejecución muy rápida y eficaz, pero se apoya en un requisito de confianza en el sistema central que sólo puede darse en un dominio cerrado en el que los usuarios se vean compelidos a aceptar estas condiciones draconianas. Eso de entregar información cuyo significado se desconoce recuerda a esos cuentos en los que alguien tenía que entregar en mano una carta lacrada en la que se decía que se le cortase la cabeza al portador de la misiva. Algo inadmisibles en esquemas de comportamiento en los que los ciudadanos necesitan y exigen preservar su independencia y su privacidad en las comunicaciones seguras que utilizan.

Con ligeras variantes, existen otros modelos en los que la utilización de similares servidores de autenticación en línea (es decir, se accede a ellos instantes antes de pretender hacer uso de otros servicios de la red) sirve para proporcionar autenticación de entidades en arquitecturas en las que se usa exclusivamente criptografía de clave secreta. Como ya hemos reiterado, serán los protocolos basados en criptografía de clave pública los que realmente den una respuesta satisfactoria a este tipo de necesidades.

8.3. FUNCIONES RESUMEN, O FUNCIONES HASH

Como adelantábamos en el Capítulo 5 y recordábamos en el apartado 8.1, un **resumen** o **valor hash** es una pieza de información de tamaño fijo, generada mediante una función unidireccional, que también puede servir como autenticador de mensajes. No obstante, la importancia de las funciones hash radica en que son parte fundamental de los algoritmos de firma digital. Este tipo de funciones tiene larga tradición en el mundo de la informática y las telecomunicaciones, donde se han usado siempre que se necesitaba transformar un identificador numérico, perteneciente a un conjunto con un número muy grande de elementos, en otro identificador perteneciente a otro conjunto con un número de elementos mucho más reducido, realizándose dicha conversión o aplicación conforme a un algoritmo bien definido, de forma que dado un identificador, siempre que se le aplique la función se obtenga el mismo resultado.

Por tratarse de funciones unidireccionales (*one-way functions*), el cálculo de la función directa es computacionalmente «fácil» de realizar, pero la operación inversa es computacionalmente impracticable. Por ello, estas funciones tienen cierto parecido formal con las que se definen en los algoritmos de clave pública, con la particularidad de que estas últimas sí son invertibles para alguien que conozca una determinada información secreta: una *trampa* o *clave de descifrado*. Por contra, en las funciones hash no existe trampa alguna que posibilite la operación inversa. Además, las funciones hash generan una pieza de información que es siempre del mismo tamaño, independientemente de lo grande que sea el mensaje de entrada que se procese. En este sentido, tienen gran parecido con los procesos que generan un testigo MAC a partir de un determinado mensaje, con la diferencia importante de que estos procesos se realizan conforme a una clave secreta, en tanto que las funciones hash están definidas de forma que no se requiera clave alguna.

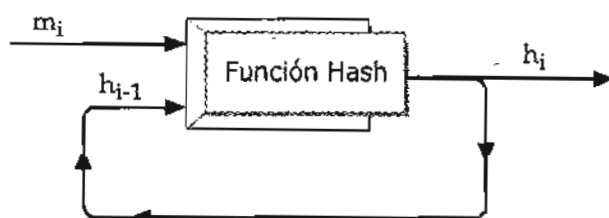


Figura 8.8. Generación del resumen o valor hash de un mensaje m .

Es decir, dado m (de longitud arbitrariamente larga) y H , función resumen o función hash, se obtiene:

$$h = H(m)$$

que es de longitud fija. Además, es impracticable el cálculo de m mediante $H^{-1}(h)$.

Los algoritmos más significativos en los que se basan las funciones hash tienen un comportamiento parecido al de los algoritmos de clave secreta de cifrado en bloque. El mensaje original se divide en p bloques de n bits cada uno:

$$m = (m_1, m_2, \dots, m_p)$$

El primer bloque, m_1 , actúa como entrada de la función hash junto con un vector de inicialización de N bits de longitud (igual al tamaño fijo del resumen), generando a la salida una pieza de información h_1 , también de longitud N . El bloque m_2 junto con h_1 actúan como entrada dando lugar a h_2 , y así sucesivamente, según se muestra en la Figura 8.8, donde el valor de h_0 se corresponde con el vector de inicialización, prefijado en cada algoritmo, a que antes se ha hecho referencia. De forma que:

$$h_i = f(m_i, h_{i-1})$$

donde h_i es el código hash correspondiente al bloque m_i .

El código hash del último bloque es el código hash del mensaje m completo.

$$H(m) = h_p = h$$

LA ROBUSTEZ DE LAS FUNCIONES HASH

Las condiciones que estos algoritmos deben cumplir se pueden resumir en los siguientes requisitos:

1. Dado un mensaje m de cualquier longitud, es computacionalmente factible y relativamente fácil obtener h , de tamaño fijo N , a partir de m .
2. Dado un resumen h , es computacionalmente impracticable deducir m a partir de h .
3. Dado un mensaje m_1 es difícilísimo encontrar otro mensaje m_2 tal que $H(m_1) = H(m_2)$. Esta dificultad es tanto mayor cuanto mayor sea el número de bits del resumen h .
4. Es muy difícil encontrar dos mensajes m_1 y m_2 tales que $H(m_1) = H(m_2)$. Igualmente, esta dificultad es tanto mayor cuanto mayor sea el número de bits del resumen h .

De forma similar a lo que ocurre con los códigos MAC, debido a que el espacio de los posibles mensajes m es mayor que el de los resúmenes h , es posible que se

produzcan *colisiones*, de tal forma que las circunstancias descritas en los puntos 3 y 4 sean posibles. No obstante, cuando el número de bits de h es suficientemente grande, los trabajos necesarios para encontrar un mensaje m_2 que cumpla con lo especificado en el punto 3 son sumamente complejos desde el punto de vista computacional, debido a que la probabilidad de que eso se produzca de forma aleatoria es la inversa de 2^N , que para un valor de N de 128 es una probabilidad sumamente baja. El esfuerzo requerido para conseguir esta colisión por fuerza bruta es proporcional a este factor de 2^N . Además, si de lo que se trata es de falsificar un mensaje del que se conoce su resumen, encontrando otro que presente el mismo valor del resumen, no basta con encontrar otro mensaje cualquiera que satisfaga esa condición, sino otro mensaje que, además de satisfacerla, tenga sentido y un significado favorable para los intereses del falsificador. Y eso añade todavía más dificultad a la tarea.

Los trabajos necesarios para encontrar una situación como la descrita en el punto 4 son de un grado de dificultad algo menor (el valor de la probabilidad de encontrarlos al azar es, en este caso, el inverso de $2^{N/2}$), aunque desde un punto de vista práctico esta tarea resulta también irrealizable. Constituye el denominado «ataque del cumpleaños» que se comentará al hablar de la fortaleza de la firma digital. Para prevenir este ataque, los algoritmos con valor de N de 128 bits (como es el caso de MD5) resultan poco seguros, por lo que resultan más aconsejables algoritmos que generan resúmenes de 160 bits (como es el caso de SHA-1 y RIPEMD-160).

USO DEL VALOR HASH COMO AUTENTICADOR DE MENSAJES

Según ya habíamos anunciado, el resumen de un mensaje puede ser usado como componente de un código de autenticación de mensajes. En la Figura 8.9 se presenta una de estas configuraciones. En ella, el resultado de aplicar la función hash al mensaje, $H(m)$, se cifra con una clave secreta k mediante un algoritmo simétrico. Este criptograma, $E_k(h)$, junto con el mensaje, se envían concatenados al extremo receptor. La entidad receptora debe compartir la clave secreta de cifrado para poder descifrar $E_k(h)$ y obtener el valor hash del mensaje original. El mensaje recibido, que lo denominamos m' ante la sospecha de que haya podido ser modificado durante su transmisión, se procesa mediante la misma función resumen que se usó en la entidad origen de los datos, obteniendo h' . Si los valores de h y de h' coinciden, ello significa que el mensaje proviene de la entidad originadora esperada y que no ha sido manipulado en el camino.

Este esquema de comportamiento es parecido al de utilización de un código MAC representado en la Figura 8.5. En ambos casos se necesita cifrar una determinada información y procesar el mensaje completo, siendo necesario que los protocolos que implementen estos mecanismos añadan información acerca de la identidad de la entidad A, debido a que cuando B reciba el mensaje tendrá que conocer de forma inmediata esa identidad, para seleccionar, de entre todas las que posee en su base de datos, la clave secreta k que comparte con A.

Un segundo esquema para utilizar un resumen como autenticador de mensajes se representa en la Figura 8.10. En este caso no se utiliza ningún criptosistema sino que se hace uso de una información secreta, IS, compartida por ambas entidades, que se concatena, en la entrada de la función hash, con el mensaje cuyo origen se desea autenticar. El resultado de esa transformación es transmitido, junto con el mensaje, a la entidad receptora, la cual procede de forma similar con el mensaje recibido, m' , y con la copia de IS que posee. Como resultado de la comparación

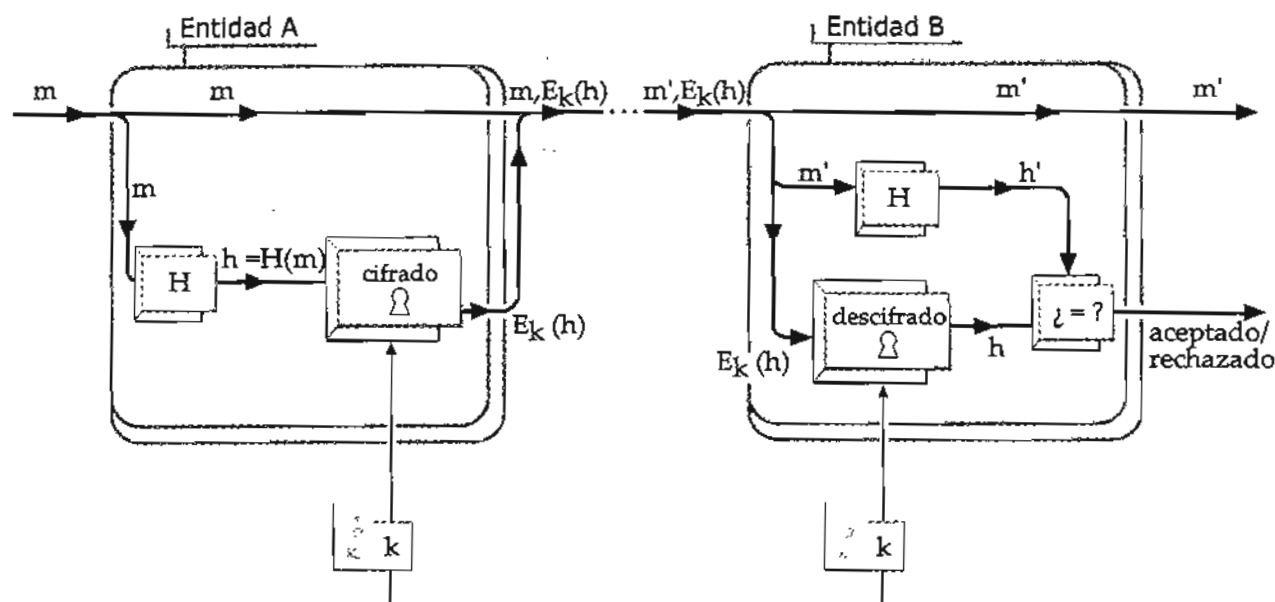


Figura 8.9. Valor hash cifrado usado como autenticador de mensaje.

de ambas muestras del resumen se estará en condiciones de aceptar o rechazar la validez del mensaje.

Tanto el MAC como estos esquemas de funcionamiento que utilizan funciones hash son, desde un punto de vista conceptual, muy similares entre sí (de hecho algunos autores los recogen a todos bajo la denominación genérica MAC), aunque la rapidez de ejecución y la robustez de las funciones hash es muy superior a la de los códigos MAC generados mediante criptosistemas de clave secreta. Como ya se comentó, una carencia importante de estos mecanismos es que no pueden ser utilizados en servicios de no-repudio de origen, ya que cuando B reciba un determinado mensaje de A, en el caso de que A se niegue a reconocer la existencia de este mensaje, la entidad B no dispone de ninguna prueba que presentar ante terceros, ya que ella misma comparte con A la información utilizada en la autenticación. Esta deficiencia se corrige en los mecanismos de firma digital.

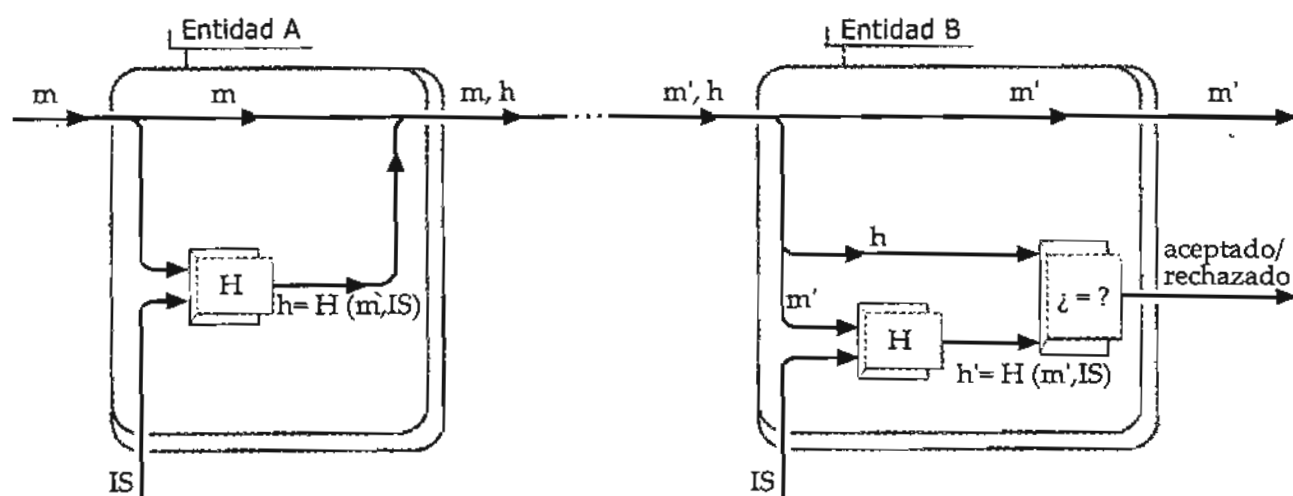


Figura 8.10. Uso combinado del valor hash y de un secreto para autenticación de mensajes.

ALGORITMOS HASH MÁS SIGNIFICATIVOS

Estos algoritmos, también denominados MAA, Message Authentication Algorithms, por ser útiles en distintos mecanismos empleados en la provisión del servicio de autenticación, no se basan en problemas de difícil solución (como las funciones unidireccionales con trampa en que se basan los criptosistemas de clave pública), sino que presentan una estructura interna muy parecida a los algoritmos simétricos o de clave secreta. Como se expresa en la Figura 8.8, el bloque m_i , de n bits, se combina con el código hash de la etapa anterior para dar como resultado el correspondiente h_i de N bits de longitud. Como n es bastante mayor que N , en el proceso de transformación se producen operaciones de compresión que es donde radica la fortaleza del algoritmo, ya que estos algoritmos, como otros, son atacables no sólo por fuerza bruta sino aprovechándose de ciertas características de su diseño particular.

No se van a describir aquí los esquemas de funcionamiento de los algoritmos hash más frecuentemente utilizados, que pueden encontrarse en cualquiera de los tratados de criptografía que hemos venido reseñando, sino que se presentará simplemente un resumen de sus características más importantes. Esos son:

MD-5

Es el más evolucionado de un conjunto de algoritmos hash desarrollados por Ron Rivest, entre los que se encuentran el MD-2 (*message-digest algorithm*), que es relativamente lento e inseguro, y el MD-4, que fue publicado en 1990 y ha servido de modelo para la definición de los algoritmos hash más robustos, que son los que se reseñan en este apartado.

El MD-5 fue publicado en 1991 y recogido como RFC 1321 en 1992. Constituye una mejora del MD-4 antes mencionado, consiguiendo resolver algunas debilidades que se le habían encontrado, por lo que es más complejo y algo más lento que el MD-4. Es uno de los algoritmos más ampliamente utilizados, aunque, debido a la longitud del resumen que genera, está siendo sustituido por el SHA-1 y por RIPEMD-160.

Genera un resumen de 128 bits, por lo que el vector de inicialización que se corresponde con el h_0 de la Figura 8.8 está constituido por cuatro registros de 32 bits cada uno con valores ya definidos en el algoritmo.

Al mensaje original se le añaden 64 bits que informan de su longitud. A continuación, el mensaje resultante, con ese añadido, se divide en bloques de 512 bits, divididos a su vez en 16 subbloques de 16 bits cada uno. Se consigue que la longitud final sea múltiplo de 512 mediante un mecanismo de relleno (*padding*) que consiste en un uno seguido de tantos ceros como sean necesarios. El algoritmo impone que siempre ha de existir relleno, aunque la longitud del mensaje a segmentar sea múltiplo de 512. En este último caso el relleno ocuparía los 512 bits del último bloque.

A continuación el bloque m_i es sometido a cuatro rondas o ciclos de oscurecimiento encadenados entre sí, de forma que la entrada para la primera ronda son m_i y el valor h_{i-1} procedente del procesamiento del bloque anterior. El resultado de esta operación actúa como entrada de la segunda ronda junto con el m_i que se está procesando, y así sucesivamente. Las cuatro rondas tienen una estructura similar, basada en operaciones XOR, AND, OR y NOR, pero difieren entre sí en la función primitiva concreta que se aplica en cada una de ellas. La salida de la cuarta ronda se suma módulo 2 con el h_{i-1} que sirvió de entrada en la primera ronda, dando lugar al h_i resultante del procesamiento del bloque m_i .

Al igual que los restantes algoritmos que aquí se resumen, el valor hash del último bloque es el valor hash del mensaje m completo.

$$H(m) = h_p = h$$

SHA-1, Secure Hash Algorithm

Diseñado por NIST (National Institute of Standard and Technology) y la NSA (National Security Agency) de los Estados Unidos, publicado inicialmente en 1993 y revisado posteriormente en 1995, fue concebido para ser usado en conjunción con el estándar DSS de firma digital (que se describe en el apartado 8.5). El estándar que recoge el SHA-1 es también conocido como SHS (Secure Hash Standard). Ha sido emitido por el FIPS de Estados Unidos (Federal Information Processing Standard) bajo la referencia FIPS PUB 180-1.

Al igual que el MD-5, su diseño se basa en el del MD-4, en el que se introdujeron algunas variaciones. La más significativa de ellas es que genera un resumen de 160 bits. Debido a ello, el vector de inicialización que se corresponde con el h_0 de la Figura 8.8 está constituido por cinco registros de 32 bits cada uno (A, B, C, D, E) con valores enteros definidos en el algoritmo.

De forma análoga a la que hemos descrito para el caso del MD-5, al mensaje original se le añade una pieza de información de 64 bits conteniendo su longitud y, también de la misma forma, se procede al relleno para conseguir un número entero de bloques de 512 bits.

También es muy similar la estructura interna de oscurecimiento a la que se somete al bloque, que está constituida por cuatro rondas y una suma final, aunque las funciones primitivas son distintas en este algoritmo y h_i es, obviamente, de 160 bits.

Es considerado un algoritmo muy seguro tanto por su estructura interna como por el hecho de que al generar un resumen de 160 bits, es muy resistente ante ataques de fuerza bruta. El SHA-1 es más seguro que el MD-5 frente a criptoanálisis, ya que algunas de las debilidades detectadas en éste fueron evitadas cuando se procedió a su diseño. En cuanto a velocidad, el SHA-1 es algo más lento que el MD-5 porque emplea mayor número de operaciones y porque trabaja con registros de 160 bits.

RIPEMD-160

Es un algoritmo más reciente que los otros que hemos presentado [DBP96]. Su diseño tuvo también al MD-4 como punto de partida, analizando y tratando de corregir las vulnerabilidades que se habían detectado, lo que convierte al RIPEMD en el algoritmo más seguro frente a criptoanálisis.

En todo lo relativo a la segmentación del mensaje y el añadido de información sobre la longitud de éste, se comporta igual que el SHA-1. También genera un resumen de 160 bits. A diferencia del MD-5, que admite una longitud indefinida para el mensaje de entrada, estos dos algoritmos limitan a $2^{64} - 1$ el número máximo de bits que puede tener el mensaje.

La función de compresión que aporta robustez al algoritmo es bastante diferente de las correspondientes a los otros algoritmos antes citados, aunque también se basa en operaciones AND, OR, NOR y XOR. En este caso se realizan 160 operaciones (en SHA-1 son 80 y en MD-5 son 64). En cuanto a velocidad, es algo más lento que el

SHA-1, que a su vez es más lento que el MD-5, aunque esa demora quede compensada por la mayor seguridad.

No obstante, cabe decir que cualquiera de estos tres algoritmos proporcionan una seguridad más que suficiente para la mayoría de las operaciones que tienen cabida en las aplicaciones que interesan a los ciudadanos para resolver sus necesidades reales de comunicación.

8.4. FIRMA SIMPLE, FIRMA DIGITAL Y FIRMA ELECTRÓNICA

Ya en 1988, en la norma (repetidamente aquí citada) ISO/IEC 7498-2 *Security Architecture* se recogía el concepto de **mecanismo de firma digital**, cuyas características más relevantes, adoptando una nomenclatura más conforme con las definiciones vigentes en la actualidad, se han resumido en el Capítulo 5 (apartado 5.3 y Figura 5.6, que se reproduce de nuevo como Figura 8.11). Este mecanismo define a su vez dos procesos:

- Firmado de una pieza digital de información por parte del **firmante** o **signatario**.
- Verificación de una pieza de información firmada.

A partir del documento original m , mediante un **Proceso de Firma** se genera la **firma**:

$$f = A_{\text{sig}}(m).$$

El documento **firmado** es el conjunto formado por la concatenación de m y f . Tras la recepción del mensaje firmado por parte de la entidad **verificadora**, mediante un **Proceso de Verificación** se genera una información que indica la validez o falsedad de la firma.

Conforme a estos planteamientos, podemos definir la **firma digital** como *una pieza de información añadida a una unidad de datos, que es el resultado de una*

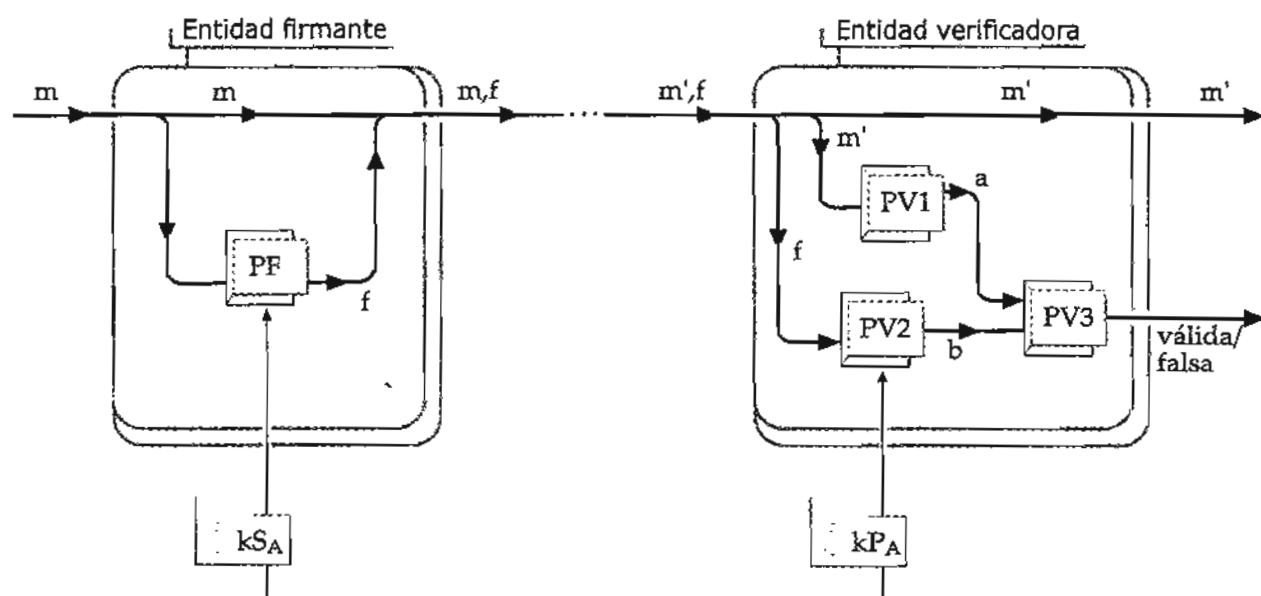


Figura 8.11. Mecanismo de firma digital.

transformación criptográfica de ésta en la que se ha usado una información privada del signatario, que permite a una entidad receptora probar la autenticidad del origen y la integridad de los datos recibidos.

Además, para acercar más la definición a los casos que nos interesan, podemos imponer dos nuevas condiciones:

- a) La firma f debe ser de longitud reducida e independiente del tamaño de m .
- b) La generación de la firma se realiza a partir del resumen (*hash*) del mensaje m mediante un proceso de cifrado en el que se utiliza la clave privada del signatario que la emite.

Antes de añadir estas dos nuevas condiciones, el resultado de cifrar todo el mensaje con la clave privada del signatario podría ser considerado como una firma digital, por lo que la hemos denominado **firma simple**. Este procedimiento conlleva la dificultad, ya comentada, de que la firma sería de tamaño similar al mensaje firmado, lo que representa un problema insalvable cuando se trata de mensajes muy largos, tanto por el tamaño de la firma como por la duración del proceso de firmado (los algoritmos de clave pública son relativamente lentos).

En concordancia con esta concepción del mecanismo, en la X.509 se definieron nuevos tipos ASN.1 que permiten expresar formalmente la estructura de las unidades de datos puestas en juego. Estos tipos, que se recogen con cierto detalle en el apartado 6.6, son: ENCRYPTED, HASHED, SIGNATURE y SIGNED.

Merced a estas características, desde un principio se vio que el mecanismo de firma digital podría jugar en el ciberespacio un papel similar al que durante siglos ha venido desempeñando la firma caligráfica en las comunicaciones convencionales mediante papel. Para que esta traslación al mundo digital sea más apropiada, y para conseguir que la firma digital tuviese el mismo reconocimiento legal que la firma manual, se ha considerado que es necesario que a la firma digital propiamente dicha se le añadan informaciones complementarias acerca de la fecha de firmado, algoritmos utilizados y certificados puestos en juego.

Ello ha motivado que se tratase de desarrollar y ampliar el esquema antes definido para dar satisfacción a todas las necesidades que se han detectado. De estas ampliaciones han surgido nuevos nombres y nuevas definiciones que tratan de recoger exigencias y características añadidas. En este planteamiento es en el que se inscribe el concepto de **firma electrónica**, como procedimiento que incluye y supera al más reducido de **firma digital**. La elección del nombre puede que no haya sido de lo más acertado, pero, en cualquier caso, nos sirve para establecer diferencias.

En este contexto se ha venido trabajando tanto en un marco normativo legal como en la elaboración de estándares técnicos en los que se especifica la estructura de los datos utilizados. En Europa se ha elaborado una Directiva del Parlamento Europeo y del Consejo que, en el caso de España, ha tenido su correspondiente traslación a la normativa legal interna. Paralelamente, el ETSI (*European Telecommunications Standard Institute*) ha desarrollado una propuesta que trata de adaptarse a las condiciones marcadas por la Directiva.

En resumen, podemos considerar tres tipos de firma:

- **Firma simple.** La firma simple de m sería el mero cifrado con la clave privada (secreta) del firmante.
- **Firma digital.** Se realiza a partir del resumen (*hash*) del mensaje m mediante un proceso de cifrado en el que se utiliza la clave privada del firmante.

- **Firma electrónica.** Consiste en una firma digital con informaciones añadidas para potenciar su validez.

En los tres casos el *mensaje firmado* es el conjunto de m y de la firma f .

Para verificar la validez de una firma, además de conocer la clave pública del firmante:

- En el caso de *firma digital* y *firma electrónica* siempre es necesario conocer m para poder obtener el resumen.
- En el caso de *firma simple* si m es inteligible (por ejemplo un contrato) bastaría sólo con la firma f . Si el mensaje es ininteligible (por ejemplo una clave criptográfica) para una verificación que aporte confianza al verificador es necesario (conveniente) disponer no sólo de la firma sino también de m .

A estas definiciones de firma habría que añadir la *firma a ciegas* o *firma opaca*, que será estudiada en el capítulo dedicado a los servicios de anonimato. Además, en las definiciones del ETSI se distingue, a su vez, entre distintas clases de Firma Electrónica (con sus distintos nombres), en la Directiva de la UE se introduce la categoría de *Firma Electrónica Avanzada* y en la legislación española se tipifica la *Firma Electrónica Reconocida*. Tanto unas como otra serán estudiadas en sucesivos apartados.

8.5. ALGORITMOS DE FIRMA DIGITAL

Dos son los algoritmos comúnmente usados en las redes telemáticas para conseguir el mecanismo de firma digital. Uno de ellos es el RSA, más sencillo, que utiliza los algoritmos de cifrado que le son propios en combinación con una operación hash del mensaje que se pretende firmar. Por ello, una determinada clave privada puede servirle a una entidad tanto para firmar mensajes como para descifrar un mensaje que otra entidad haya cifrado con su clave pública. El otro algoritmo, el DSA, es algo más elaborado y tiene la propiedad de que es específico para firma, no pudiendo ser usado para otras tareas de cifrado.

ALGORITMO DE FIRMA DIGITAL RSA

En la Figura 8.12 se muestra de nuevo (ya se presentó anteriormente como figura 5.7) el esquema de funcionamiento seguido para la obtención y verificación de la firma digital apoyándose en un algoritmo de cifrado RSA. Es conveniente distinguir el algoritmo de cifrado RSA propiamente dicho del *algoritmo de firma digital RSA*. En efecto, podremos obtener lo que antes hemos llamado *firma simple* mediante el cifrado, con RSA, de todo el mensaje con la clave privada del firmante, que será algo totalmente distinto a la firma digital con resumen (hash).

Dado un mensaje m , la entidad firmante A, para obtener la **firma** realiza dos operaciones:

- Realiza $h = H(m)$.
- Cifra $H(m)$ con su clave privada.

$$f = A_s[H(m)] = A_s[h] = h^d \bmod n$$

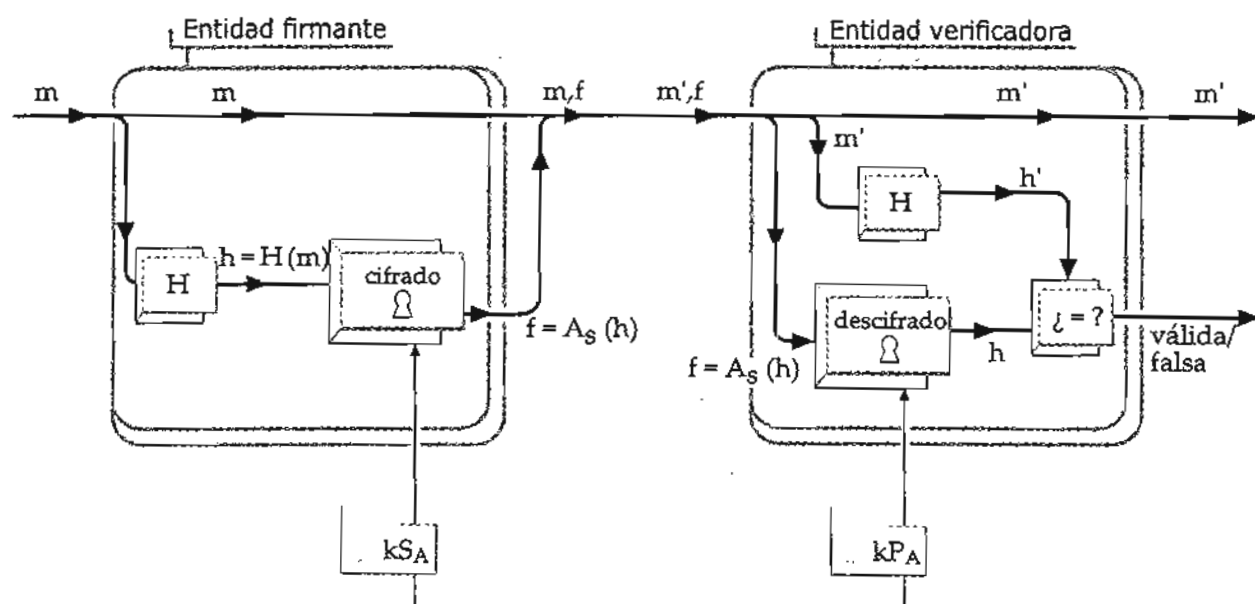


Figura 8.12. Esquema de la firma RSA.

donde n y d son los componentes de la clave privada de A ($kS_A = n, d$). Esto conlleva que el tamaño de f es reducido y similar al tamaño de h .

El **mensaje firmado** es el conjunto formado por m y f que puede ser **verificado** por cualquier entidad que conozca la clave pública de A, situación bastante sencilla de alcanzar si la entidad verificadora tiene acceso a un certificado válido de dicha clave.

En la Figura 8.12 hemos supuesto que es la entidad B la que realiza la verificación. Al mensaje recibido por B le llamamos m' porque, aunque es de esperar que sea el mismo m que envió la entidad A, hasta que no se verifique la firma existe la duda razonable de que haya sido alterado durante la transmisión de forma fortuita o intencionada. Si la firma resulta ser verdadera, sabremos, entre otras cosas, que $m' = m$.

Para efectuar la verificación, la entidad B realiza las siguientes operaciones:

1. Descifra f con la clave pública de A ($kP_A = e, n$). Como RSA es un algoritmo de cifrado reversible, lo que se cifre con una clave privada puede descifrarse con la correspondiente clave pública:

$$A_p(f) = f^e \bmod n = (h^d \bmod n)^e \bmod n = h$$

Es decir:

$$A_p(f) = A_p(A_s[h]) = h = H(m)$$

2. Realiza $H(m') = h'$. Es imprescindible que el algoritmo de *hashing* utilizado en la verificación sea exactamente el mismo que el utilizado para la obtención de la firma. Por ello es necesario que en la especificación de los mecanismos de firma se incluya, como veremos más adelante, una referencia explícita a dicho algoritmo.
3. Comprueba que $H(m') = H(m)$. Si esto es cierto, da por válida la firma y adquiere suficiente certeza acerca de la autenticidad del firmante y de la integridad del mensaje recibido ($m' = m$).

Este mismo procedimiento de firma podría utilizarse con cualquier otro algoritmo de clave pública que, al igual que el RSA, tuviese la propiedad, antes referida, con-

sistente en que lo que se cifre con una clave privada puede descifrarse con la correspondiente clave pública.

ALGORITMO DE FIRMA DIGITAL DSA

Como ya se comentó, el estándar SHS que define el SHA-1 y el DSS (Digital Signature Standard) donde se define el DSA (Digital Signature Algorithm) fueron concebidos por el NIST para ser usados de forma conjunta. El DSS, catalogado como FIPS PUB 186, se publicó inicialmente en 1991 y ha tenido algunas revisiones posteriores que no afectan a lo esencial de la propuesta.

En realidad, DSA es un nuevo algoritmo de cifrado asimétrico que no es reversible como RSA, es decir, permite descifrar con la clave pública aquello que se haya cifrado con la clave privada, pero no permite cifrar con la clave pública y descifrar con la privada. Es por tanto un algoritmo que posibilita la provisión de los servicios de autenticación e integridad pero que no sirve para proporcionar confidencialidad. Esta carencia no es casual porque se inscribe dentro de la misma corriente que el Key Escrow y, en general, de todos aquellos procedimientos que consideraban que los ciudadanos no deberían disponer de herramientas que les permitiesen blindar sus comunicaciones sin ninguna restricción externa sobre las claves de cifrado. Evidentemente, en la situación actual, el uso de DSA no representa ninguna merma para la privacidad, ya que se puede usar este algoritmo para la firma y RSA para obtener confidencialidad cifrando con la clave pública del receptor.

El algoritmo DSA está basado en el de El Gamal, que contempla procedimientos distintos para cifrar con la clave pública y con la clave privada. Consiste en hacer públicos una serie de parámetros y, naturalmente, las claves públicas de las entidades participantes.

Los datos públicos para todo el dominio son:

- **p** Número entero con un número de bits múltiplo de 64 y comprendido entre 512 y 1.024.
- **q** Número primo de 160 bits divisor de p .
- **g** Número entero cuyo valor es $g = j^{(p-1)/q} \bmod p$, donde j es un entero positivo elegido aleatoriamente menor que $p - 1$ y primo relativo con p que cumple con: $j^{(p-1)/q} \bmod p > 1$.

Para una entidad A signataria, las claves son:

- **Clave privada:** kS_A : número aleatorio mayor que uno y menor que q (para simplificar las expresiones, la denominaremos x).
- **Clave pública:** $kP_A = g^x \bmod p$ (para simplificar las expresiones, la denominaremos y).

Para la **firma** del mensaje m , el procedimiento seguido por la entidad A signataria (Figura 8.13) es el siguiente:

1. Realiza $h = H(m)$.
2. A partir de h , de los datos públicos p , q y g , de la clave privada x y de un número aleatorio k (mayor que cero y menor que q) obtiene la **firma** f :

$$f = r, s \quad \text{donde:}$$

$$r = (g^k \bmod p) \bmod q$$

$$s = [k^{-1} (h + xr)] \bmod q$$

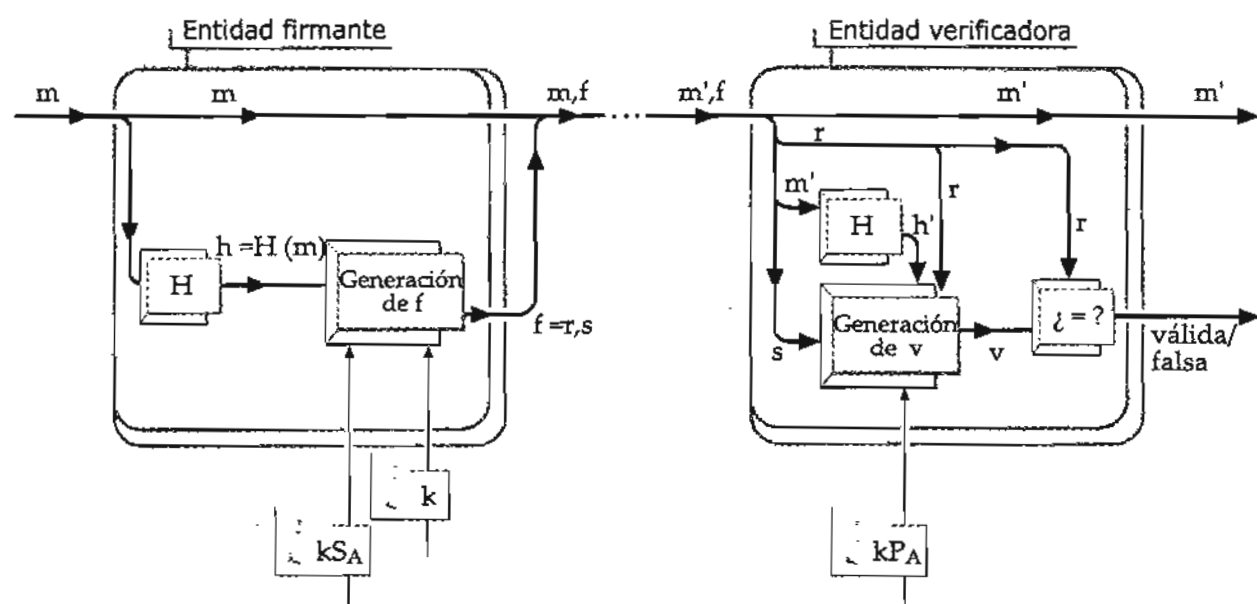


Figura 8.13. Esquema de la firma digital DSA.

donde k^{-1} es el inverso multiplicativo de k . Es decir: $(kk^{-1}) \bmod q = 1$.
El **mensaje firmado** es el conjunto formado por m y f :

$$m, r, s$$

En la Figura 8.13 hemos supuesto que B es la entidad verificadora. La entidad B recibe conjuntamente m' y f (que es una pieza de información ininteligible para ella). Al mensaje recibido le llamamos m' porque existe la duda razonable de que haya sido alterado durante la transmisión. Para efectuar la **verificación**, la entidad B realiza las siguientes operaciones:

1. Realiza $H(m') = h'$.
2. A partir de h' , de los datos públicos p, q y g , de la firma f y de la clave pública y obtiene el parámetro v conforme a la siguiente expresión:

$$v = [(g^{u1} y^{u2}) \bmod p] \bmod q \quad \text{donde:}$$

$$u1 = (h'w) \bmod q$$

$$u2 = (rw) \bmod q$$

$$w = s^{-1} \bmod q$$

3. Se puede demostrar que si $h' = h$, entonces $v = r$. Por ello, B comprueba que $v = r$ y si se produce dicha igualdad da por válida la firma y adquiere suficiente certeza acerca de la autenticidad del firmante y de la integridad del mensaje recibido ($m' = m$).

La demostración de que si $h' = h$ implica que el valor de v coincide con r , es larga y algo engorrosa, aunque perfectamente seguible apoyándose en los fundamentos matemáticos resumidos en el Capítulo 3. Veámoslo:

Como punto de partida, se puede demostrar que para cualquier entero no negativo N , se cumple que:

$$g^N \bmod p = g^{N \bmod q} \bmod p \quad (1)$$

En efecto, N puede representarse como:

$$\begin{aligned} N &= nq + z \Rightarrow z = N \bmod q \\ g^N \bmod p &= g^{nq+z} \bmod p = (g^{nq} \bmod p) (g^z \bmod p) = \\ &= (g^{nq} \bmod p) (g^{N \bmod q} \bmod p) \end{aligned} \quad (2)$$

Conforme a la definición de g :

$$\begin{aligned} g^{nq} \bmod p &= (j^{(p-1)/q} \bmod p)^{nq} \bmod p = j^{((p-1)/q)nq} \bmod p = j^{(p-1)n} \bmod p = \\ &= ((j^{p-1})^n \bmod p) \bmod p \end{aligned}$$

Debido a que h y p son primos entre sí, conforme al teorema de Fermat, $j^{(p-1)} \bmod p = 1$, por lo que:

$$g^{nq} \bmod p = 1$$

lo que conlleva que el resultado de la expresión (2) coincida con el de la expresión (1), conforme queríamos demostrar.

Apoyándonos en esto, tratemos de demostrar que v coincide con r . Conforme a la definición de v , $u1$ y $u2$:

$$v = [(g^{u1} y^{u2}) \bmod p] \bmod q = [(g^{hw} \bmod q y^{rw} \bmod q) \bmod p] \bmod q \quad (3)$$

Por definición de la clave pública y , simplificando conforme a las reglas de la aritmética modular y aplicando la expresión (1), el segundo factor de la expresión (3) queda:

$$\begin{aligned} y^{(rw) \bmod q} \bmod p &= (g^x \bmod p)^{(rw) \bmod q} \bmod p = g^{[x(rw) \bmod q] \bmod q} \bmod p = \\ &= g^{(xrw) \bmod q} \bmod p \end{aligned}$$

que sustituido en (3) y aplicando (1) resulta:

$$\begin{aligned} v &= [(g^{hw} \bmod q g^{(xrw) \bmod q}) \bmod p] \bmod q = [(g^{(hw + rxw) \bmod q}) \bmod p] \bmod q = \\ &= [(g^{[(h + rx)w] \bmod q}) \bmod p] \bmod q \end{aligned} \quad (4)$$

Conforme a la definición de s , el producto $(ks) \bmod q$ resulta:

$$\begin{aligned} (ks) \bmod q &= (k[k^{-1} (h + xr)] \bmod q) \bmod q = \\ &= [(k k^{-1}) \bmod q] [(h + xr) \bmod q] \bmod q = (h + xr) \bmod q \end{aligned} \quad (5)$$

El exponente de g en la expresión (4) una vez desglosado, sustituido el valor de la expresión (5) y tenido en cuenta que, por definición, $(sw) \bmod q = 1$, resulta:

$$\begin{aligned} [(h + rx)w] \bmod q &= [(h + rx) \bmod q] [w \bmod q] \bmod q = \\ &= [(ks) \bmod q] [w \bmod q] \bmod q = (ksw) \bmod q = \\ &= [(k \bmod q) ((sw) \bmod q)] \bmod q = k \bmod q = k \end{aligned}$$

que sustituido en (4) y aplicando la definición de r , resulta:

$$v = [g^k \bmod p] \bmod q = r$$

que es lo que queríamos demostrar.

8.6. FIRMA ELECTRÓNICA: UNA MEJORA DE LA FIRMA DIGITAL

Aunque hay muchos otros, los dos algoritmos de firma digital que acabamos de presentar son, con diferencia, los más aceptados y sobre los que más pruebas de seguridad se han realizado. En este epígrafe, al hilo de la descripción del «ataque del cumpleaños» se discute acerca de la robustez que, en la práctica, tienen estos algoritmos. A pesar de su robustez, un asunto a tener en cuenta es el de las insuficiencias que pueden detectarse en los servicios que se obtienen con los algoritmos de firma digital y la necesidad de buscar nuevas especificaciones que superen esas deficiencias.

VALIDEZ DE LA FIRMA DIGITAL: EL ATAQUE DEL CUMPLEAÑOS

De hecho, el «ataque del cumpleaños», que puede servir para falsificar una firma digital, es en realidad un ataque a la seguridad del resumen obtenido mediante la función hash. Por ello es efectivo en los algoritmos de firma que, como es el caso de la firma RSA, se basen sólo en el cifrado del resumen. Este nombre tan pintoresco procede, según parece, de la constatación de que si en una reunión de equis personas preguntamos si alguna de ellas nació en determinado día del año, la probabilidad de que alguien responda afirmativamente es muchísimo más baja que si lo que preguntamos es si dos de esas equis personas cumplen años el mismo día. Es decir, es mucho más difícil encontrar una persona que cumpla los años en una fecha concreta que buscar dos personas que cumplan años en el mismo día (sin importar de qué día se trate).

Del mismo modo, dado un mensaje $m1$, la probabilidad de encontrar otro mensaje $m2$ tal que $H(m1) = H(m2)$ es la inversa de 2^N , siendo N el número de bits del código hash, en tanto que la probabilidad de encontrar dos mensajes $m1$ y $m2$ tales que coincidan sus resúmenes es la inversa de $2^{N/2}$.

Esta particularidad se aplicaría al fraude que un usuario malicioso M quiere hacer a costa de otra persona bienintencionada, que llamaremos B . Para que las cosas sean más sencillas vamos a suponer que $N = 64$. Supongamos que el usuario M ha llegado a un acuerdo con B en relación con un determinado tema y ha redactado un contrato que recoge los términos pactados y que se supone va a enviar a B para que lo firme.

Redacta dos contratos: uno de ellos conforme a lo pactado, que vamos a denominar cfB , «contrato favorable a B »; y el otro modificando las cláusulas para que sea favorable a su intereses, que denominaremos cfM , «contrato favorable a M ».

Antes de que B pueda verlo por primera vez, M juega con los dos contratos introduciendo ligeras variaciones en las palabras, en los espacios, márgenes, etc., hasta conseguir que los resúmenes de ambos mensajes sean iguales: $H(cfM) = H(cfB)$. El esfuerzo que ha tenido que realizar para ello es proporcional a $2^{64/2}$, que es mucho pero realizable utilizando métodos y máquinas potentes.

Entonces, envía cfB a B para que lo analice y, si está conforme, lo firme. B está de acuerdo con los términos, lo firma y se lo devuelve a M . Si ha utilizado el algoritmo de firma digital RSA, la firma que recibe es $f = B_p[H(cfB)]$. Con posterioridad, presenta ante un juez el contrato cfM y la firma f que había recibido de B .

El juez analiza el caso, por sí mismo o a través de un perito realiza $H(cfM)$, luego cifra la firma f con la clave pública de B y obtiene $B_p(f) = B_p(B_p[H(cfB)]) = H(cfB)$, que es una pieza de información ininteligible pero comparable al resumen $H(cfM)$ que el mismo juez ha realizado sobre el mensaje que le ha sido presentado por M . Como

comprueba que ambos resúmenes son iguales, considera que B firmó el contrato *cfM* y obliga a B a cumplirlo.

Este ataque es más un divertido juego académico que un peligro real por varias razones:

Si B, antes de firmar y sin modificar lo acordado en modo alguno, realiza un ligero cambio, como añadir un espacio en blanco después de un punto y aparte, el resultado de la función hash sería totalmente distinto y el invento se vendría abajo. (Moraleja: cuando le pidan lo de «firme aquí» zarandee un poco el mensaje antes de firmarlo.)

Además, si se trata de algoritmos que producen resúmenes de 160 bits, el esfuerzo necesario sería proporcional a 2^{80} , que conllevaría unas tareas y un tiempo de ejecución que no estarían al alcance de cualquiera. Dentro de los mensajes (y las firmas) de la vida diaria, que se acogen bajo el paraguas de lo que hemos denominado Seguridad Cívica, la robustez que ofrecen esas funciones hash resulta más que suficiente. (Muchísimo más barato sería violar la resistencia personal de un ciudadano y obligarle a firmar lo que no desee.) En cualquier caso, resulta evidente que la firma digital es muchísimo más difícil de falsificar que lo que pudiera ser una firma caligráfica.

Por último, como ya se ha dicho, este ataque sólo es posible cuando se trate de algoritmos de firma con un esquema similar al RSA. Si la firma se realizase con, por ejemplo, DSA, ya no serviría para nada, porque en la firma intervienen otros parámetros además del valor hash.

INSUFICIENCIAS DEL MECANISMO DE FIRMA DIGITAL

Tal y como hemos planteado el mecanismo de firma digital, los actores o protagonistas del proceso son únicamente el signatario que firma el mensaje y el receptor que lo verifica. Podría haber otros receptores que aceptasen la validez de la firma por simple confianza, sin proceder a su verificación, pero esos participantes carecen de interés en el esquema que estamos analizando.

Un elemento de capital importancia en ese escenario es la validez del par de claves, privada y pública, del signatario. El medio más directo y seguro de que dispone el receptor del mensaje es obtener un certificado de la clave pública que no haya sido revocado antes de que la firma fuese emitida. Dependiendo de para qué haya sido firmado el documento, el receptor deberá conocer si la política bajo la cual fue emitido el certificado permite que esa operación tenga o no un respaldo externo. Esto significa que en el escenario de la firma deben existir, además, una o varias terceras partes de confianza: las CAs involucradas en el camino de certificación.

Pero hay más: aparte de la política de certificación propiamente dicha, dependiendo de la importancia de la firma, es razonable que interese definir una política de firmado en la que se regule, por ejemplo, si el certificado debe ser añadido o no al mensaje firmado, cómo ha de llevarse a cabo la constatación de la validez del certificado por parte del receptor del mensaje, quién o quiénes actuarán de árbitros en caso de conflictos, y otras muchas circunstancias que pueden presentarse.

Además, en la vida real se requiere a veces que un determinado mensaje sea firmado por más de una persona. Una forma de hacerlo es «en paralelo», es decir, todos los firmantes firman el mismo documento y el conjunto de todas las firmas es el que le otorga validez. La otra es firmar de forma «anidada»: cada nuevo signatario (generalmente de superior jerarquía) firma el documento ya firmado en el paso anterior. Este tipo de comportamientos es conveniente recogerlo de manera formal en una especificación que mejore la del simple mecanismo de firma digital antes descrito.

Otro elemento importantísimo en una firma es el instante de tiempo en el que fue realizada. En el caso de la firma digital esta circunstancia tiene especial relevancia debido al hecho de que, hoy por hoy, se admite que una determinada entidad, por motivos justificados, pueda cambiar su par de claves, repudiando por tanto la clave pública reconocida en el dominio de seguridad de que se trate y solicitando la revocación del certificado correspondiente incluso dentro del periodo de validez con el que éste ha sido emitido. Esto significa que un usuario malicioso puede conseguir una clave privada obsoleta de la entidad A y firmar un documento en el que aparezca una fecha en la que la clave estaba aún vigente.

Es decir, se requiere que existan procedimientos de sellado de tiempo que, mediante las correspondientes TTPs, garanticen, cuando ello sea necesario, la validez de las fechas que aparecen en los documentos firmados. En la descripción de los algoritmos de firma digital que se hizo en el epígrafe anterior se insistió en que la verificación de la firma proporcionaba una razonable confianza tanto en la autenticidad del origen de los datos como en la integridad del mensaje recibido. Para que, además, el autor de la firma no pueda repudiar su autoría, condición que recogíamos como uno de los requisitos de la firma electrónica, es necesario tener pruebas robustas, garantizadas por TTPs, acerca del instante en que fue firmado.

En general, la firma, por su propia naturaleza, debe poder utilizarse en procedimientos y escenarios en los que permanezca siendo válida durante periodos de tiempo bastante largos, y no sólo en situaciones en las que el mensaje sea válido durante un estrecho margen de tiempo, como es el caso del uso de la firma en los protocolos de autenticación de entidades.

Por todo ello, se llega a la conclusión de que el escenario elemental de la firma digital necesita ser mejorado y aumentado con nuevas piezas de información y nuevos agentes telemáticos que proporcionen, de forma conjunta, las necesarias garantías exigidas para que tenga la validez adecuada un documento firmado y para que el autor de la firma no pueda repudiar su autoría. En este contexto es en el que se concibe el concepto de **firma electrónica** que ya fue introducido en el epígrafe 8.4 y que será desarrollado en los siguientes. Antes vamos a tratar de recapitular cuáles han de ser los requisitos y características que debe tener la firma electrónica.

CARACTERÍSTICAS DE LA FIRMA ELECTRÓNICA

En comparación con las que tiene la firma manuscrita, de la cual pretende ser su plasmación digital, cabe seleccionar las siguientes características o peculiaridades:

1. Fácil de hacer

El usuario puede «estampar» su firma en un documento sin gran complicación ni esfuerzo. La firma caligráfica es, en la actualidad, más fácil de realizar que la firma electrónica, para la que se requiere cierto equipamiento. En un futuro próximo, el empleo masivo de tarjetas inteligentes puede facilitar enormemente el uso de la firma electrónica.

2. Susceptible de ser reconocida por el receptor

El receptor del mensaje firmado debe ser capaz de reconocer, de forma fiable, la validez de la firma, bien sea por sí mismo o con ayuda de terceros que actúen como validadores.

En el caso de la firma manuscrita, una persona corriente es incapaz de detectar una falsificación mínimamente habilidosa. Por otra parte, en infinidad de casos, los ciudadanos reciben mensajes firmados por personas a las que desconocen por completo, cuya firma es la primera vez que la ven y no tienen ni idea de cómo es la «firma verdadera», con lo cual difícilmente pueden establecer la más elemental evaluación sobre la validez del documento firmado. En caso de conflictos hay que recurrir a peritos calígrafos que, con excesiva frecuencia, emiten dictámenes contradictorios. A pesar de ello, miles y miles de firmas caligráficas flotan a diario en un escenario donde la más elemental falta de seguridad contrasta con la gran aceptación social que tienen y la prodigalidad con la que a diario se nos pide «firme aquí».

La firma electrónica, en cambio, se presta a que cualquier receptor que tenga acceso de forma fiable de la clave pública del signatario (si nos ceñimos al mecanismo de firma que estamos manejando) puede tener una razonable confianza en la validez de la firma. Además, la verificación está al alcance de cualquiera, con lo cual se trata de un mecanismo que refuerza en mucho la privacidad de los usuarios.

3. Dificil de falsificar

Ambos tipos de firma cumplen esta condición. No obstante, aunque es cierto que la seguridad de los algoritmos utilizados no es infinita, si comparásemos el esfuerzo tecnológico y económico que hay que realizar para romper un criptosistema que maneje una clave de, por ejemplo, 2.000 bits, con el que sería necesario realizar para encontrar y pagar a un falsificador virtuoso, llegaríamos a la conclusión de que es muchísimo más segura una firma electrónica que una firma caligráfica. Por ello, como prueba judicial debe ser considerada mucho más robusta una firma electrónica que una simple firma manual.

4. Identifica al firmante

Es decir, **auténtica** al signatario. Tanto un tipo de firma como el otro deben servir para identificar inequívocamente (dentro de lo razonable) al firmante o signatario. Sobre la base de los argumentos ya expuestos, podemos llegar a la conclusión de que este requisito lo cumple de forma mucho más eficaz la firma electrónica.

5. Va unida al mensaje formando un todo

Una firma que aparece en un documento no puede ser trasladada a otro. Esta condición la cumplen ambos tipos de firma. No se puede recortar una firma manuscrita y trasladarla a otro documento sin que sea detectable (salvo cuando se trata de faxes firmados, falsificación que está al alcance de un infante poco avisado y de corta edad, a pesar de lo cual, inexplicablemente, en muchos ambientes se da un estimable valor probatorio a este tipo comunicaciones). En el caso de la firma electrónica esto no es posible porque la firma depende tanto de la clave privada del firmante como del documento firmado.

6. Afecta a todo el mensaje

En otras palabras, sirve para garantizar la **integridad** del documento firmado. Una vez firmado un documento, no puede alterarse su contenido si no es realizando una falsificación. Cuando se firma un documento en papel contenido en varias páginas es necesario firmarlo en todas y cada una de ellas para garantizar la integridad. Si se trata de una firma electrónica, por las razones ya aducidas, una sola firma afecta a todo el documento. El simple cambio de un bit produciría una firma totalmente distinta.

7. El autor de la firma no puede repudiar su autoría

Quien accede al documento firmado adquiere una prueba, demostrable ante terceros, de que el mensaje ha sido firmado por quien dice ser el signatario. Junto con la autenticidad del origen de los datos y la integridad del mensaje, es una de las condiciones más importantes que cumplen ambos tipos de firma.

8. Permiten incluir información adicional al texto

Ambos tipos de firma pueden ser complementados con sellos de tiempo, información sobre la identidad de los destinatarios, etc., que sirvan para proteger la firma contra posibles falsificaciones y mal uso.

Como acabamos de ver, estas ocho condiciones las cumplen tanto la firma manual como la firma electrónica, aunque con una considerabilísima ventaja a favor de esta última. No parece fácil crear un criterio objetivo y cuantificable para establecer, de forma numérica, una valoración comparativa de la robustez de ambos tipos de firma, pero si ello fuese factible, con toda seguridad resultaría que la fortaleza de la firma electrónica es varios órdenes de magnitud superior a la de la manuscrita.

Existe, no obstante, una característica diferenciadora entre ambas:

9a. La firma manual es siempre la misma

Es decir, que aunque cambie el documento, la firma será la misma. Esto no entra en contradicción con que un usuario utilice firmas distintas dependiendo de si el documento es de tipo personal o es de carácter oficial, ya que, entre varios documentos del mismo tipo, reproduce siempre la firma de forma idéntica.

9b. La firma electrónica depende de la totalidad del mensaje

Como ya se ha dicho, mensajes distintos darán lugar a firmas digitales totalmente distintas. Con ello, la integridad del mensaje (característica número 6) queda muy reforzada. De forma análoga a lo comentado en el párrafo anterior, también una entidad signataria podrá disponer de pares de claves (pública y privada) especializadas en distintos tipos de firma, dependiendo del carácter del documento y del rol que soporte en el momento de la firma. Ello dependerá de la política de seguridad (política de firma) definida en el entorno en el que la entidad esté incluida.

8.7. PLANTEAMIENTOS DE LA FIRMA ELECTRÓNICA

Para que las firmas electrónicas puedan usarse masivamente en las comunicaciones entre ciudadanos y en transacciones comerciales que se produzcan a través de redes telemáticas es necesario que tengan un valor social equivalente al de las firmas manuscritas. La plasmación de esa equivalencia viene dada por el establecimiento de determinados requisitos que deben ser cumplidos por una firma electrónica para que tenga, respecto de los datos consignados en forma electrónica, el mismo valor jurídico que la firma manuscrita tiene en relación con los consignados en papel. Como consecuencia de ello, una firma electrónica que cumpla esos requisitos será admisible como prueba en juicio, valorándose en consonancia con la legislación vigente en el ámbito de que se trate.

Como quiera que las legislaciones son diferentes y específicas de cada país, en tanto y cuanto no se homogenice la legislación a nivel mundial, será necesario que en cada país se establezcan normativas concretas en las que se especifique con precisión cuáles han de ser los requisitos que deben cumplir las firmas electrónicas para que se produzca ese reconocimiento. Es por ello que en casi todos los países desarrollados existen iniciativas para cubrir esta demanda.

Habida cuenta del carácter transnacional que representan las comunicaciones mediante redes telemáticas, es fácil deducir la necesidad que existe de normas que garanticen la validez de las firmas electrónicas más allá de las fronteras concretas de un determinado estado. Aunque de forma parcial, en el ámbito de la ONU, para facilitar la necesaria homologación de la legislación sobre comercio electrónico, se están abordando iniciativas que pueden aportar criterios comunes a la hora de valorar jurídicamente la firma electrónica.

UN MARCO NORMATIVO PARA LA UNIÓN EUROPEA

El caso de la Unión Europea es particularmente interesante por concurrir dos circunstancias en principio contradictorias: de una parte, la necesidad de un espacio europeo sin fronteras interiores en el que esté garantizada la libre circulación de mercancías y, de otra, la heterogeneidad normativa en materia de reconocimiento legal de la firma manuscrita. Por este motivo, la Unión Europea ha promovido diversas iniciativas conducentes a generar documentos y especificaciones comunes que puedan, posteriormente, ser trasladados a las legislaciones específicas de los distintos países miembros.

Un hito importantísimo en este proceso lo constituye la publicación, en enero de 2000, de la Directiva 1999/93/CE del Parlamento Europeo y del Consejo por la que se establece un marco comunitario para la firma electrónica. En el caso de España esa Directiva se plasmó en la publicación, en septiembre de 1999¹, de un Real Decreto-Ley sobre firma electrónica y, poco más tarde, la Orden por la que se aprobó el Reglamento de acreditación de los prestadores de servicios, todo ello modificado y actualizado posteriormente en los trabajos preparatorios de la nueva Ley de Firma Electrónica².

De esta forma, desde los poderes públicos de muchos países se ha estado trabajando para generar normativas jurídicas que garanticen el valor legal de las firmas electrónicas para que puedan utilizarse tanto en el campo del comercio como de otras actividades sociales. Aunque es más que probable que la diligencia desplegada responda al interés de potenciar el comercio electrónico para favorecer a las empresas implicadas en ese sector, no cabe duda que estas regulaciones y aclaraciones pueden

servir para el mejor desarrollo de la seguridad de las comunicaciones que interesan a los ciudadanos en otras múltiples facetas de su vida diaria.

En la definición de firma recogida en la Directiva se ha procurado generalizar los términos de manera que no sea imprescindible la utilización del mecanismo de firma digital basado en hash y en criptografía de clave pública tal y como lo hemos definido antes. Esta independencia del concepto de firma digital justifica que se haya elegido una denominación, *firma electrónica*, que refleja también, desde un punto de vista técnico, una mayor generalización, dando cabida a otras soluciones que puedan vislumbrarse en el futuro.

Para los objetivos aquí perseguidos, lo que nos interesa no es la reseña exacta ni el análisis legislativo de estos documentos (que irán evolucionando y perfeccionándose gradualmente), sino las definiciones que en ellos se recogen y el análisis de los medios que serán necesarios para cumplir los requisitos que se exigen para la validez jurídica de la firma electrónica.

DEFINICIONES

La Directiva europea contiene un conjunto de definiciones que, con mínimas diferencias, también se recogen en la normativa española antes citada. En primer lugar aparece:

Firma electrónica: *los datos en forma electrónica anejos a otros datos electrónicos o asociados de manera lógica con ellos, utilizados como medio de autenticación.*

Como primera interpretación, cabe destacar que hace referencia a la existencia de unos datos (la firma) asociados a otros datos electrónicos (el documento firmado). Además, sólo exige que sirva como medio para identificar formalmente al autor o autores. Esta definición se completa con otra más matizada:

Firma electrónica avanzada: *la firma electrónica que cumple los requisitos siguientes:*

- a) *Estar vinculada al firmante de manera única.*
- b) *Permitir la identificación del firmante.*
- c) *Haber sido creada utilizando medios que el firmante puede mantener bajo su exclusivo control.*
- d) *Estar vinculada a los datos a que se refiere, de modo que cualquier cambio ulterior de los mismos sea detectable.*

Es oportuno analizar estas definiciones teniendo en mente el concepto de firma electrónica que hemos adoptado en el epígrafe 8.4. Básicamente, éste consiste en una firma digital a la que se le añaden una serie de parámetros, que sirven para determinar con precisión la validez de las claves empleadas y las condiciones bajo las cuales la firma ha sido llevada a cabo. Por su parte, la firma digital, tal y como la hemos definido anteriormente, está basada en algoritmos que utilizan como entrada la clave privada del firmante y el valor hash de aquello que se firma.

Es fácil comprobar que esta forma de concebir la firma concuerda plenamente con las definiciones recogidas en la Directiva, **representando un caso particular de éstas**, que están concebidas de forma más genérica, de tal manera que pueden seguir siendo válidas aunque en un futuro se creen otras tecnologías (basadas en PKIs o no) que también satisfagan los requisitos exigidos.

En la Directiva se recogen otras cuantas definiciones que sirven para matizar el significado de la firma:

Firmante: *la persona que está en posesión de un dispositivo de creación de firma y que actúa en su propio nombre o en el de la entidad o persona física o jurídica a la que representa.* En la norma española que traslada la Directiva se asigna el nombre de **Signatario**. Nosotros utilizaremos ambos términos indistintamente.

Para el proceso de generación de la firma se definen:

- **Datos de creación de firma:** *los datos únicos, tales como códigos o claves criptográficas privadas, que el firmante utiliza para crear la firma electrónica.* En el caso de apoyarse en la firma digital, se corresponde exactamente con la clave privada que el signatario utiliza para firmar.
- **Dispositivo de creación de firma:** *un programa informático configurado o un aparato informático configurado que sirve para aplicar los datos de creación de firma.* Si cumple con los requisitos enumerados en la Directiva recibe el calificativo de **seguro**.

Para la verificación se cuenta con los siguientes elementos:

- **Datos de verificación de firma:** *los datos, tales como códigos o claves criptográficas públicas, que se utilizan para verificar la firma electrónica.* La concreción de firma electrónica que hemos recogido en los apartados anteriores se centra en la clave pública pareja de la clave privada con la que se realizó la firma y en los certificados e informaciones complementarias que sirven para determinar la validez de ésta en el momento en que fue realizada la firma.
- **Dispositivo de verificación de firma:** *un programa informático configurado o un aparato informático configurado, que sirve para aplicar los datos de verificación de firma.* Si cumple con las condiciones enumeradas en la Directiva se calificará como **seguro**.
- **Certificado:** *la certificación electrónica que vincula unos datos de verificación de firma a una persona (o cualquier entidad signataria) y confirma la identidad de ésta.* Aunque no es precisamente un dechado de definición (casi incluye lo definido en la definición), para los efectos aquí perseguidos, queda claro que el Certificado X.509 cumple satisfactoriamente con esta exigencia. Se llamará **certificado reconocido** cuando el certificado cumple los requisitos establecidos en la Directiva y es suministrado por un **proveedor de servicios de certificación** que cumple los requisitos establecidos en un anexo de la misma.

Es menester llamar la atención sobre el hecho de que, aunque pueden considerarse implícitamente incluidos, la Directiva no hace mención expresa a dos conceptos de capital importancia que son condición *sine qua non* para tener garantías acerca de la validez de la firma. Nos referimos a la inclusión de marcas de tiempo fiables (fecha electrónica) y a la determinación de la política de seguridad concreta bajo la que se establecen los compromisos cuando se lleva a cabo el proceso de generación y verificación de la firma. En el caso del Estado español, la Ley de Firma Electrónica sí recoge estos dos conceptos e introduce la denominación de *Firma Electrónica Reconocida* para identificar con mayor precisión a la firma electrónica que cumple con los requisitos necesarios para ser considerada equivalente a la firma manuscrita.

La robustez de las piezas de información que han de utilizarse en el proceso depende de cuáles sean las condiciones que se exijan a las entidades u organismos que

expidan certificados o den por válidos los programas informáticos o equipos que sirvan para la generación y verificación de firmas. En este sentido se definen:

- **Proveedor de servicios de certificación:** *la entidad o persona física o jurídica que expide certificados o presta otros servicios en relación con la firma electrónica.*
- **Producto de firma electrónica:** *el programa informático o el material informático, o sus componentes específicos, que se destinan a ser utilizados por el proveedor de servicios de certificación para la prestación de servicios de firma electrónica o que se destinan a ser utilizados para la creación o la verificación de firmas electrónicas.*
- **Acreditación voluntaria:** *todo permiso que establezca derechos y obligaciones específicas para la prestación de servicios de certificación, que se concedería, a petición del proveedor de servicios de certificación interesado, por el organismo público o privado encargado del establecimiento y supervisión del cumplimiento de dichos derechos y obligaciones [...].*

En la normativa específica de cada uno de los Estados deberá decidirse qué organismo será el encargado de establecer los adecuados procedimientos de acreditación y supervisión que tendrán validez en todos los países de la Unión.

NECESIDAD DE ESTÁNDARES RECONOCIDOS

Realmente, desde un punto de vista técnico, el plato fuerte de la Directiva lo constituyen los cuatro anexos que especifican, con cierto detalle, los requisitos que han de cumplir los certificados reconocidos, los proveedores que expiden certificados y los dispositivos de creación y verificación de firmas.

En cuanto a los certificados reconocidos se dictamina que han de contener: a) *la indicación de que el certificado se expide como certificado reconocido*; b) *la identificación del proveedor de servicios de certificación y el Estado en que está establecido*; c) *el nombre y los apellidos del firmante o un seudónimo que conste como tal*; d) *un atributo específico del firmante, en caso de que fuera significativo en función de la finalidad del certificado*; e) *los datos de verificación de firma que correspondan a los datos de creación de firma bajo control del firmante*; f) *una indicación relativa al comienzo y fin del periodo de validez del certificado*; g) *el código indentificativo del certificado*; h) *la firma electrónica avanzada del proveedor de servicios de certificación que expide el certificado*; i) *los límites de uso del certificado, si procede*; y j) *los límites del valor de las transacciones para las que puede utilizarse el certificado, si procede*. Es decir, condiciones todas ellas perfectamente alcanzables mediante el Certificado X.509.

Los requisitos exigidos a los proveedores de servicios de certificación que expiden certificados reconocidos son muchos (doce en total) y muy diversos, y están relacionados con cuestiones como a) ofrecer garantías de un servicio rápido y fiable de forma que se tomen las medidas necesarias para la adecuada identificación de las personas y entidades a las que han de expedírseles o revocárseles certificados; b) utilizar sistemas, procedimientos y productos fiables, así como comprometerse a emplear a personal técnicamente cualificado en materia de seguridad informática; c) disponer de recursos económicos suficientes para hacer frente a las responsabilidades contraídas (que son objeto, a su vez, de una descripción pormenorizada dentro del articulado de

la Directiva); d) registrar y archivar adecuadamente la información relativa a los certificados expedidos durante el periodo de tiempo que se estipule, permitiendo a las personas autorizadas el acceso a la información en ellos contenida; y e) dejar claramente establecido con la entidad solicitante de un certificado las condiciones bajo las que éste será expedido.

En cuanto a los requisitos exigidos a los dispositivos seguros de creación y verificación de firmas, cabe decir que, obviamente, determinan de forma muy genérica las condiciones de seguridad y resistencia al fraude que deben ser recogidas en los estándares que se definan acerca de la Firma Electrónica.

En definitiva, la Directiva y las normas legales resultantes de su transposición en los distintos estados europeos dictaminan que la firma electrónica avanzada, basada en un certificado reconocido y creada por un dispositivo seguro homologado, tendrá, en relación con los datos consignados en forma electrónica, **el mismo valor jurídico** que la firma manuscrita tiene en relación con los datos consignados en papel y será admisible como prueba de juicio.

No obstante, para que estas condiciones y requisitos de seguridad puedan ser exigidas es necesario que existan estándares conformes con la Directiva que puedan ser aplicados de forma precisa y objetiva. La Directiva no indica el procedimiento para llegar a desarrollar estos estándares y solamente prevé publicar en su boletín oficial los números de referencia de aquellas normas que gocen de reconocimiento general para productos de firma electrónica.

Para conseguir este objetivo, las autoridades comunitarias han promovido la *European Electronic Signature Standardisation Initiative*, EESSI, dentro de cuyo ámbito se recogen, entre otros, los trabajos de especificación que lleva a cabo el ETSI (*European Telecommunications Standard Institute*) y el CEN (*Comité Europeo de Normalización*) acerca de los Formatos de Firma Electrónica y procesos de validación, a los que se hará reiterada referencia en los párrafos siguientes³.

8.8. PROPUESTAS TÉCNICAS: FORMATOS DE FIRMA ELECTRÓNICA

Ya en 1988, en los anexos aclaratorios de la ISO/IEC 7498-2 *Security Architecture* se contemplaba la conveniencia de que existiesen terceras partes en el escenario donde se llevasen a cabo los procesos de firma y verificación. Así (Figura 8.14), se establecía una distinción en cuanto a la actuación de éstas. Por una parte se tendría el *Esquema de firma directa*, en el que la tercera parte es simplemente un juez (dicho en sentido genérico, puede ser simplemente alguien aceptado por ambas partes). El juez se mantiene al margen de la comunicación y solamente verifica la firma cuando aparece un conflicto entre los participantes, ya que alguno de ellos puede no decir la verdad. En la otra opción, denominada *Esquema de firma arbitrada*, la tercera parte (o terceras partes) autentica la fuente de los datos y puede establecer registros de la operación realizada para utilizarlos en caso de litigio.

Desarrollando este segundo esquema se pueden llegar a definir escenarios de comunicación bastante evolucionados en los que una TTP puede actuar fuera de línea (*off-line*), como es el caso de una CA que genera certificados, o puede estar directamente involucrada en el intercambio de información. En esta segunda alternativa pueden darse, a su vez, otras dos situaciones. La primera de ellas se corresponde con el esquema representado en la Figura 8.14 b) y consiste en que la TTP está interpuesta en el camino (*in-line*) de forma que el mensaje firmado es enviado a la TTP y es ésta

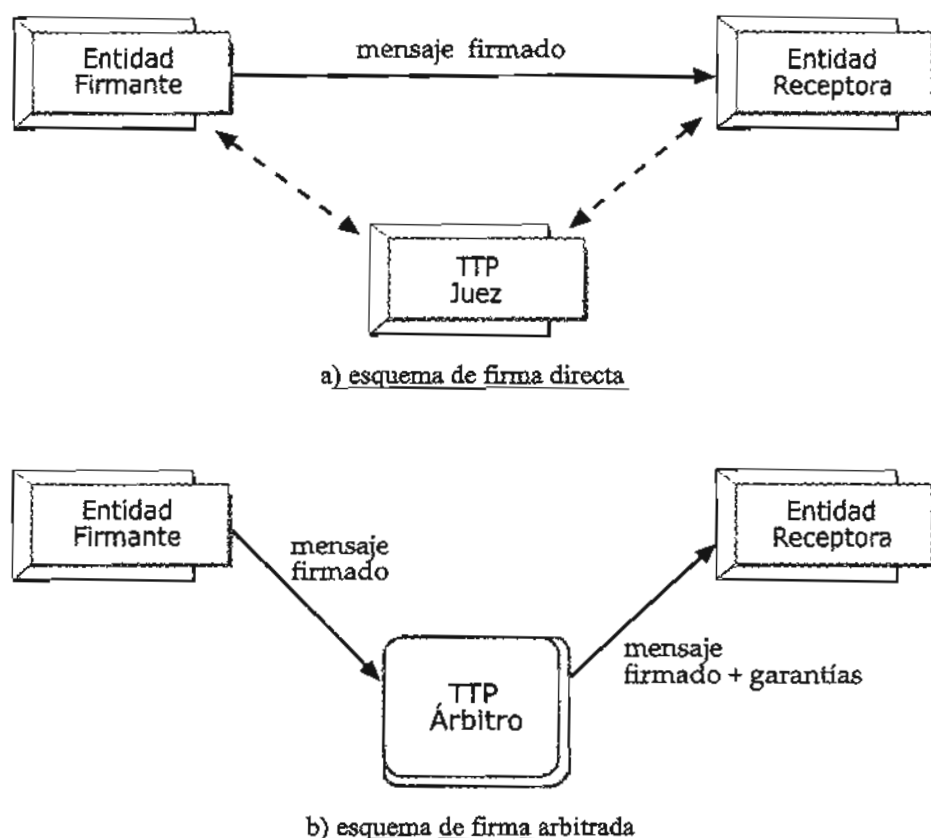


Figura 8.14. Dos esquemas de utilización de la firma digital.

la que lo transfiere al receptor (añadiendo, por ejemplo, marcas de tiempo). La segunda situación posible es que la TTP esté activamente involucrada en la comunicación (*on-line*), pero sin que pase a través de ella el mensaje que se envía a la entidad receptora.

Este segundo caso es el que se representa en la Figura 8.15 en el que el conjunto de TTPs sirven de ayuda tanto para la generación de la firma como para su verificación. En este contexto se han desarrollado múltiples soluciones y propuestas [MPW92], entre ellas la llevada a cabo por el ETSI. Así pues, lo que ha venido en llamarse **firma electrónica**, independientemente de lo acertado o no del término, no es otra cosa que una *firma digital arbitrada* en la que se introducen una serie de mejoras que tratan de resolver las insuficiencias del mecanismo clásico de firma digital, mediante el concurso de distintas TTPs que ayuden a garantizar la validez de las claves públicas y privadas que intervienen en el proceso, a fijar de forma fiable la fecha y hora en que la firma se realiza, a definir la política de firma bajo la que tiene validez, etc.

La descripción de Firma Electrónica que vamos a analizar a continuación está basada sustancialmente en la ya citada especificación del ETSI sobre *Electronic Signature Formats*⁴, que está concebida para que pueda servir para muy diversos tipos de transacciones: comerciales, contractuales, personales, administrativas, etc., al tiempo que pretende ser conforme con la Directiva europea sobre Firma Electrónica antes referida. Para su especificación se apoya en otros estándares, tanto RFCs como normas generadas por ISO e ITU-T, que abordan aspectos relacionados con la firma y con las TTPs que le sirven de apoyatura. De forma especial, para la definición formal de las distintas estructuras necesarias, el estándar de ETSI asumió las definiciones y especificaciones contenidas en la RFC 2630 *Cryptographic Message Syntax* (posteriormente dividida en dos RFCs: la 3369 *Cryptographic Message Syntax* y la 3370 *Cryptographic Message Syntax Algorithms*).

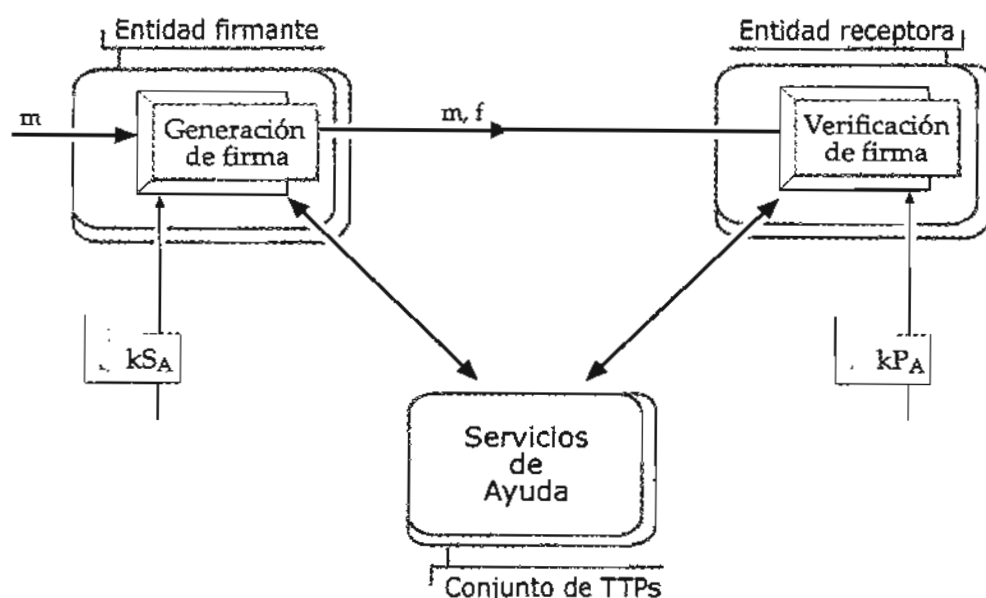


Figura 8.15. Firma electrónica usando los servicios de una Infraestructura de Seguridad.

Con buen criterio, la especificación abarca dos cosas distintas y complementarias:

- El formato de la Firma Electrónica propiamente dicha. Es decir, la sintaxis y la semántica de la pieza de información que es emitida por el signatario y recibida por la entidad verificadora.
- El formato de la Política de Firma. Recoge el conjunto de reglas exigidas en la creación y validación de la firma para que ésta resulte aceptable en ese dominio.

ESCENARIO DE LA FIRMA ELECTRÓNICA

Las entidades y agentes telemáticos involucrados en el proceso de firma y verificación son los siguientes:

- Entidades Firmantes o Signatarias. Un mismo mensaje, concatenado con otros datos adicionales que son requeridos, puede ser firmado por una o más entidades usando cada una de ellas su clave privada.
- Entidad Verificadora.
- Conjunto de TTPs que cooperan para dar validez al proceso.
- Árbitro. Es la entidad que resuelve en caso de conflicto entre la entidad signataria y la verificadora.

Al conjunto de TTPs se le denomina también por el nombre genérico de TSP, Proveedores de Servicio Confiables. Dependiendo de cada contexto y de la política de seguridad allí definida, pueden servir para dar apoyo tanto a la entidad firmante como a la verificadora. Se utilizan las siguientes TTPs, todas ellas descritas en el capítulo anterior y precedentes:

- Autoridades de Certificación, CAs, para la generación de certificados.
- Autoridades de Registro, RAs, para facilitar la identificación y el registro previo de las entidades del dominio antes de serles emitido un certificado.

- Autoridades de emisión de CRLs y proveedoras de servicios OCPS.
- Autoridades de Repositorio. Servicios de información sobre certificados, CRLs y políticas de firma. Un ejemplo de ello puede ser el Directorio.
- Autoridad de emisión de la política de firma.
- Autoridades de Sellado de Tiempo, TSAs.
- Autoridades de Atributos AAs.

Algunas TTPs serán de utilidad para la creación de la firma electrónica debido a que, para facilitar la evaluación y verificación de la firma, se debe añadir información complementaria a los datos ya firmados (el mensaje en sí más ciertos parámetros). De otra parte, durante el proceso de verificación también se requerirá el concurso de algunas TTPs para aceptar o rechazar la firma.

ESTRUCTURA DE LA FIRMA ELECTRÓNICA

Recordemos la definición de firma digital que adelantamos en el epígrafe 8.4: «una pieza de información añadida a una unidad de datos, que es el resultado de una transformación criptográfica de ésta en la que se ha usado una información privada del signatario, que permite a una entidad receptora probar la autenticidad del origen y la integridad de los datos recibidos».

Para la firma electrónica, ES, el documento de ETSI propone la siguiente definición: *Evidencia en formato digital que puede ser procesada para obtener confianza de que determinado compromiso ha sido explícitamente sostenido bajo una Política de Firma, en una fecha y hora concreta, por un signatario bajo un identificador (por ejemplo un nombre o un seudónimo), y opcionalmente un rol.*

Podemos añadir que, como comprobaremos más adelante, la recepción de esta pieza de información permite a la entidad receptora probar no sólo autenticidad del origen y la integridad de los datos recibidos, sino también asegurarse de que el autor de la firma no pueda repudiar su autoría.

Mensaje firmado

Bajo este planteamiento, teniendo en cuenta tanto las definiciones del ETSI a que nos estamos refiriendo como las especificaciones de la RFC 3369, *Cryptographic Message Syntax* (en que se apoyan), dado unos datos de usuario o mensaje m , podemos definir un **mensaje firmado** como un conjunto de datos que incluye:

- El mensaje m que es firmado.
- Información sobre algoritmos utilizados.
- Información sobre certificados.
- Información sobre revocaciones de certificados.
- Información por **cada uno** de los firmantes:
 - Certificado del firmante.
 - Identificación de los algoritmos utilizados.
 - Atributos que son firmados (incluida la identificación de la Política de Firma).
 - Otros atributos no firmados.
 - Firma digital (*Signature Value*) de m y de los atributos.

Como puede comprobarse, se contempla la posibilidad de que sean más de una las entidades signatarias del documento. Será en la Política de Firma donde se determine el número máximo de posibles firmantes (si es que existe) y las reglas que han de cumplirse para la estampación de las firmas sucesivas.

Según iremos analizando, alguno de estos parámetros (tanto los globales como los específicos de cada firmante) son optativos, ya que su misión es ayudar a la adecuada verificación de las firmas que aparecen en el documento y pueden ser añadidos en el instante mismo de la realización de la firma por el signatario, o en un momento posterior.

Formato básico de firma electrónica

La especificación del ETSI define varios formatos para la firma electrónica, en función de la información complementaria que contenga. El formato básico (Figura 8.16), al que denomina simplemente **ES** (*Electronic Signature*), lo considera constituido por los siguientes parámetros:

- Identificador de la Política de Firma (*Signature Policy Id*).
- Otros atributos que son firmados.
- Firma digital (*Signature Value*) de m y de los atributos.

Es decir, la firma electrónica estará constituida por un conjunto de atributos, en claro, y por la firma digital tanto del mensaje que se firma como de los atributos. Además de la firma y del mensaje en claro, en el **mensaje firmado** aparecen otros atributos que ayudan en el proceso de validación.

Entre los atributos que son firmados deben estar presentes, al menos, los siguientes:

- Identificador preciso de la Política de Firma (ya citado).
- Identificación precisa del certificado de la clave pública del firmante.
- Tipo de información contenida en el mensaje m firmado.
- Resumen o valor hash del mensaje m .
- Identificador del algoritmo de firma digital utilizado.
- Hora y fecha de la firma.

Política de Firma. Cada contexto legal o contractual reconocerá una política concreta como reflejo de los requisitos allí exigidos, y dicha política estará emitida por la entidad autorizada para ello. La validez de la firma está condicionada por el marco ahí especificado: por definición, una firma electrónica válida es aquella que pasa un proceso de validación conforme con la Política de Firma bajo la que ha sido emitida.

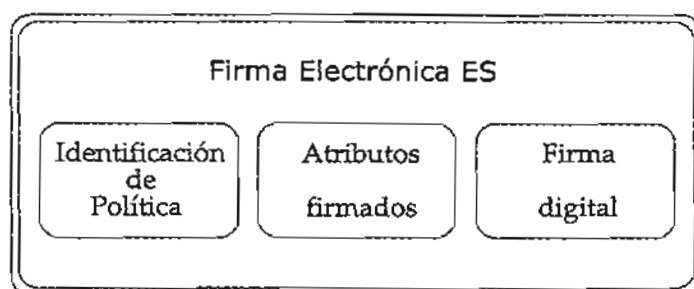


Figura 8.16. Formato básico de la Firma Electrónica. (Fuente: ETSI.)

Por eso se exige que se identifique de forma no ambigua el documento donde se describe esa política.

En la especificación formal de la firma, la identificación de la política se lleva a cabo mediante un identificador de objeto (OBJECT IDENTIFIER) y la descripción de la política consiste en un conjunto de reglas que deberán ser expresadas tanto formalmente (usando ASN.1), para que puedan ser interpretadas automáticamente por los agentes involucrados en la recepción y validación, como de forma legible para un ser humano, para que las personas que intervienen en el proceso puedan conocer las implicaciones legales y contractuales que conlleva.

Identificación del certificado. El mismo grado de concreción cabe exigir para la identificación del certificado de la clave pública del firmante. Además de la identificación del certificado de la clave que debe ser usada imprescindiblemente para la verificación de la firma, se puede añadir la identificación de otros certificados que ayuden en el proceso de determinación del camino de certificación. Una forma de reconocer el certificado es mediante la identificación de la CA emisora y del número de serie con que fue emitido.

Contenido del mensaje. Este parámetro especifica el tipo de contenido de los datos de usuario que son firmados, reconocible mediante la asignación de un valor concreto de entre un conjunto de valores posibles.

Valor hash. Se trata de una ristra de octetos que son el resultado de aplicar el algoritmo hash especificado al mensaje firmado. Obviamente, este resumen o valor hash también interviene en el algoritmo de firma digital.

Algoritmo de firma. Otro de los atributos que deben ser firmados es la indicación precisa del algoritmo de firma utilizado. En la especificación se recogen dos posibles: el RSA y el DSA. No obstante, está abierta para que puedan ser incluidos nuevos algoritmos de firma digital que vayan siendo desarrollados.

Hora y fecha. Cada uno de los signatarios, cuando apliquen el algoritmo de firmado, deberán incluir la fecha y la hora en la que la operación ha sido realizada, con vistas a la determinación de su validez en el contexto contractual del dominio donde ha sido llevada a cabo.

FIRMA CON SELLO DE TIEMPO

Es importante destacar que el formato básico de firma contiene información de tiempo sobre el momento en el que el signatario **dice** que firmó el documento. En principio, esta información no está respaldada por ninguna Autoridad de Sellado de Tiempo (TSA), por lo que la validez de la ES, tal y como está definida, no cumple con ese requisito imprescindible de garantizar, mediante una prueba demostrable ante terceros, que la firma ha sido realizada *en una fecha y hora concreta*. Será necesario, por tanto, requerir el concurso de una TSA externa para conseguir este objetivo.

Para satisfacer este requisito se define la firma electrónica con sello de tiempo, **ES-T** (*ES with Timestamp*), en la que se añade una marca de tiempo a la ES básica para posibilitar la validez de la firma durante periodos de tiempo largos. En la Figura 8.17 aparecen, resumidamente, los elementos que la constituyen.

En efecto, aunque la firma puede ser creada sin acceder en ese momento a ninguna TSA, esta firma, sin la adición de un sello de tiempo fiable, no protege contra la amenaza de que el signatario, pasado un tiempo, niegue haberla generado (es decir, no provee el servicio de no-repudio).

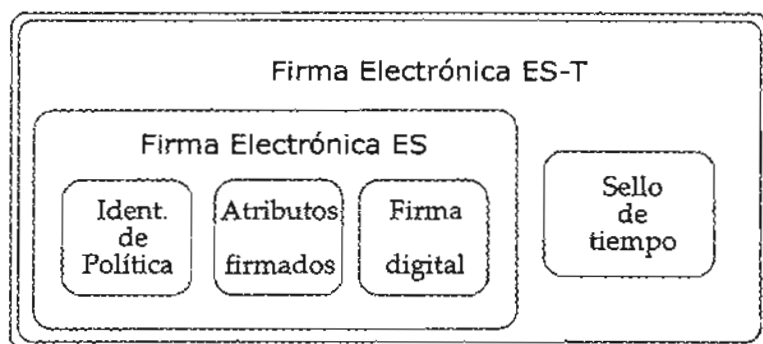


Figura 8.17. Esquema de la ES-T. (Fuente: ETSI.)

Esto es así porque si, en un momento dado, el par de claves de una entidad queda obsoleto a causa de que se ha decidido renovarlas o porque haya sido comprometida su seguridad, tiempo después, un usuario malicioso que ha conseguido ese par de claves antiguas puede generar un documento en el que la entidad en cuestión se comprometa a algo que le beneficie. Para ello podría incluir en la firma una fecha antigua, perteneciente al periodo de tiempo en el que la clave privada utilizada era válida y el correspondiente certificado no había sido aún revocado. Por esa misma razón, si no existiese un sello de tiempo fiable, un usuario que sí firmó un compromiso tiempo atrás puede, pasado el tiempo, negar la validez de la firma aduciendo que ha sido falsificada usando la clave vigente en aquel momento, y que en la actualidad ya está repudiada.

La entidad signataria puede, una vez obtenida la ES básica, decidir proveer la ES-T mediante su envío a una TSA reconocida en la política de firmado para que realice la estampación y la garantice con su firma digital. No obstante, si el firmante no proporciona la ET-S, una entidad verificadora, para poder garantizar la validez de la firma, debe crear *motu proprio* la correspondiente ES-T sobre la ES recibida acudiendo a una TSA reconocida o generando de por sí un registro seguro firmado con indicación del momento en que se ha procedido a la verificación (si es que la política de seguridad autoriza a esta entidad para ello, lo que la convierte de hecho en una TSA). Ambos mecanismos protegen contra un posible repudio de existencia de la firma.

En resumen, para que la firma sea válida se requiere la existencia de dos indicadores de tiempo y dos firmas:

- Una fecha y hora de firma indicada por el signatario (que forma parte de los datos que éste firma digitalmente).
- Una fecha y hora de firma indicada por una Autoridad de Sellado de Tiempo, TSA, que es firmado por esta TTP sobre el valor de la firma digital de la ES básica.

Para que un verificador califique como válida una firma bajo una determinada política es necesario que ambos indicadores de tiempo estén suficientemente próximos entre sí, es decir, estén dentro de unos límites de tiempo aceptables. En cada caso concreto, lo que signifique el genérico «suficientemente próximos» (minutos, horas o incluso días) será lo que se defina en la política de firmado bajo la que se esté realizando la operación.

MÚLTIPLES FIRMAS

En el uso establecido dentro del ámbito de los compromisos y comunicaciones convencionales mediante documentos en papel es frecuente el uso de múltiples firmas manuscritas estampadas en un mismo documento. En unos casos se trata de un conjunto de personas que rubrican un escrito «en paralelo» y desde un equivalente nivel jerárquico. En otros, de lo que se trata es de que alguien añada su firma como réplica o respuesta a otro documento previamente firmado, o bien firme un documento ya firmado para otorgarle su «visto bueno» o su autorización.

Para la proyección de estas prácticas al mundo de las comunicaciones telemáticas, los procedimientos de firma electrónica prevén también la existencia de múltiples firmas sobre un mismo documento o mensaje. La multiplicidad de firmas se lleva a cabo dentro de dos categorías básicas:

- Mediante firmas independientes.
- Mediante firmas encastradas (*embedded signatures*).

Las **firmas independientes** son firmas en paralelo en las que el orden de aparición no es relevante. Su plasmación en la estructura se lleva a cabo añadiendo en el formato que hemos visto del **mensaje firmado**, en el campo «por cada uno de los firmantes», tanto los atributos que son firmados como los que no lo son, añadiendo la correspondiente firma digital de los datos y de los atributos.

Las **firmas encastradas** se aplican una a continuación de otra, de forma que cada nueva firma engloba e incluye a la anterior, es decir, se firman datos ya firmados. Aquí el orden de firmado queda meridianamente reflejado en la firma resultante. Dentro de la estructura de la firma electrónica esta circunstancia puede reflejarse mediante un atributo optativo que especifique que se han realizado una o más firmas añadidas sobre la ristra de octetos resultante de la codificación del campo «firma digital» (*Signature Value*) de la firma electrónica.

En la vida real se pueden presentar gran cantidad de esquemas en los que se requiera la existencia de varias firmas realizadas por distintos agentes, algunas de las cuales puede ser necesario que se realicen en paralelo y otras que se hagan en respuesta a un documento previamente firmado o tengan el carácter de una autorización, de forma que el orden relativo de aparición de todas ellas presente diversas variantes. Todos estos esquemas pueden llevarse a cabo mediante la combinación de los dos tipos de firma múltiple que acabamos de describir.

IDENTIFICACIÓN DE LOS SIGNATARIOS

En la definición de firma electrónica que hemos recogido se dice que la operación será llevada a cabo *por un signatario bajo un identificador (por ejemplo un nombre o un seudónimo), y opcionalmente un rol*. El nombre usado por el signatario en la operación concreta de firma recogida en la especificación está contenido en el campo correspondiente del certificado. Si, como es el caso más robusto, se trata de un certificado X.509, esa información aparece (véase epígrafe 6.6) en el campo *subject*. Como ese atributo tiene que ser obligatoriamente del tipo *DN*, *Distinguished Name*, puede darse el caso de que el signatario prefiera otra de las múltiples posibilidades que le ofrece la extensión *subjectAltName*. Si el signatario no tiene *DN*, el campo *subject* del certificado puede estar vacío, en cuyo caso la extensión de nombre alternativo tiene que

aparecer obligatoriamente como crítica. Si el signatario decide usar un seudónimo, este identificador deberá estar debidamente registrado y certificado para garantizar el no-repudio en caso de conflicto.

En relación con este asunto, cabe señalar que en un entorno PKI evolucionado, una determinada entidad puede ser capaz de obtener desde diferentes CAs, o incluso de la misma, diferentes certificados X.509 conteniendo la misma clave pública para distintas formas de su nombre. Esto es así porque una entidad puede disponer de distintos pares de clave para distintos propósitos, pero también puede utilizar la misma clave privada para múltiples usos, lo cual puede suponer una ventaja en cuanto a su protección (por ejemplo si lo que usa es una tarjeta inteligente). En este caso, cada certificado estará emitido bajo una política de certificación concreta y diferenciada. Cuando use un determinado certificado en el proceso de firma, esa política de certificación, en conjunción con la política de firma, reflejarán con precisión el contexto en el cual la clave privada puede ser usada.

Debido a la importancia que tiene dejar fijada con precisión la identidad del firmante, no basta con que se adjunte a la firma el certificado del signatario, sino que se exige que la identificación precisa del certificado sea uno de los parámetros que tienen que ser obligatoriamente firmados.

En cuanto a la utilización de roles, una práctica común en el uso que se hace de la firma caligráfica sobre documentos (de forma especial cuando se trata de contratos o acuerdos) consiste en que las personas firmantes actúan en calidad del cargo que desempeñan dentro de una organización, o como propietarias de algo, o como receptoras de una determinada delegación, etc. Frecuentemente, algunos contratos sólo serán válidos si los firma alguien que ostente un cargo concreto, por lo que en muchos casos es más importante el puesto que ocupa el firmante que su propio nombre.

Estas mismas circunstancias se pueden plasmar en el esquema de firma electrónica a través de dos vías alternativas:

- Añadiendo un atributo firmado en el que se recoge el rol que el signatario dice tener.
- Añadiendo un atributo firmado en el que se identifique un **Certificado de Atributo**.

La primera de las soluciones emula la práctica común que consiste en incluir en un escrito el cargo de la persona que lo firma. La garantía de que ello es así cae bajo la responsabilidad de quien lo emite, aunque el hecho de que este atributo vaya firmado implica bastante compromiso por su parte.

En el segundo caso, lo que se adjunta es la identificación firmada (por el signatario) de un Certificado de Atributo emitido y firmado por una Autoridad de Atributos, AA. Naturalmente, esa Autoridad de Atributos deberá estar reconocida y ser de confianza (una TTP) tanto para el emisor de la firma como para su receptor. Este mecanismo es equivalente al utilizado en los contratos establecidos mediante la firma caligráfica en los que se recoge una referencia a un acta notarial en la que se confiere a una persona la prerrogativa de representar a determinada corporación o grupo. La diferencia es que, en el caso de la firma electrónica, la entidad verificadora dispone de mecanismos para comprobar en tiempo real la validez de esa certificación (cosa que debe hacer antes de dar por válida la firma), mientras que en el caso convencional de documentos de papel esa verificación requeriría de pasos y gestiones complementarias que sólo suelen llevarse a cabo en caso de conflictos.

FIRMA CON INFORMACIÓN COMPLETA PARA VALIDACIÓN

La estructura definida para la ES básica recoge los elementos mínimos imprescindibles para que la firma reúna las condiciones exigidas en la definición que hemos adoptado para la firma electrónica, a excepción de la marca de tiempo que debe proporcionar una TSA de confianza. La ES con sello de tiempo añadido es la que hemos denominado ES-T.

La especificación del ETSI que estamos utilizando como referencia propone la denominación ES-C (*ES with Complete validation data*) a la estructura resultante de añadir a la ES-T referencias al conjunto completo de datos que soportan la validez de la firma electrónica. En la Figura 8.18 se refleja esquemáticamente su estructura, en la que se añaden las referencias completas a los certificados y las correspondientes informaciones sobre revocación.

Al igual que ocurre con la ES-T, la entidad signataria del documento puede también crear la ES-C cuando haya obtenido todos los datos sobre revocación de certificados y otros parámetros necesarios para la validación. Lo que ocurre es que para obtener esos datos complementarios necesita cierto tiempo, que depende de la velocidad de respuesta de las TTPs consultadas.

Debido a ello, lo más frecuente será que la ES-C no pueda ser creada al mismo tiempo que la ES. Si la entidad signataria no proporciona la ES-C a la entidad validadora, ésta puede (y debe) recolectar los datos necesarios para construir la correspondiente ES-C y llevar a cabo con suficientes garantías el proceso de validación.

8.9. ESPECIFICACIÓN FORMAL DE LA FIRMA ELECTRÓNICA

Como ya hemos referido, la especificación formal de la ES se apoya, para la definición de sus componentes más importantes, en los planteamientos de la RFC 3369 *Cryptographic Message Syntax* (CMS). Esta norma describe una sintaxis para la protección de datos que permite encapsulaciones múltiples y sucesivas, de forma que una encapsulación de datos pueda ser anidada dentro de otra. Proporciona definiciones que permiten expresar las operaciones de firma y las operaciones criptográficas a ellas asociadas, de forma que se pueda firmar cualquier información previamente determinada.

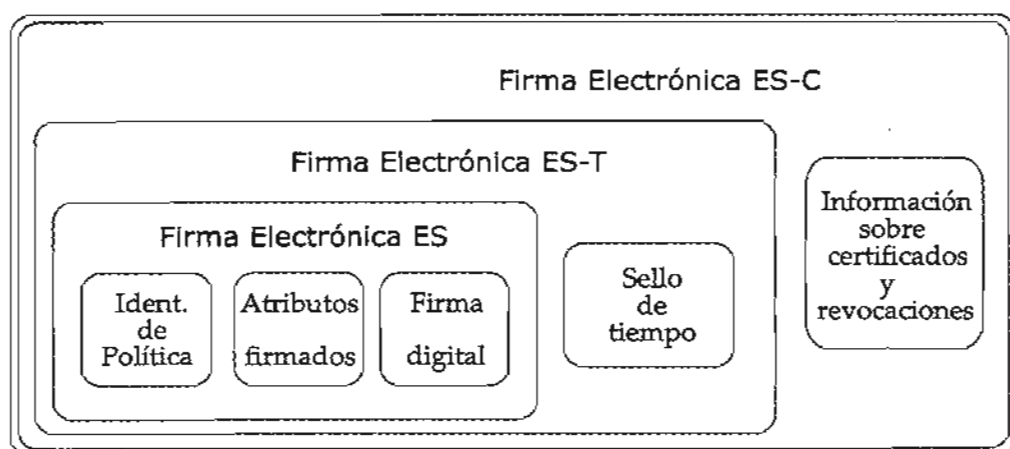


Figura 8.18. Firma electrónica con información completa para validación. (Fuente: ETSI.)

La sintaxis general de CMS consiste en asociar el identificador de tipo de contenido (*content type*) con un contenido (*content*). De esta forma se define, en ASN.1, el tipo *ContentInfo*:

```
ContentInfo ::= SEQUENCE {
    contentType          ContentType,
    content              [0] EXPLICIT ANY DEFINED BY contentType }
```

```
ContentType ::= OBJECT IDENTIFIER
```

El tipo *ContentInfo* es el más global de todos los que se definen en la sintaxis, del cual derivan todos los tipos restantes, cada uno de ellos reconocido por un identificador de objeto, OBJECT IDENTIFIER, que consiste en una cadena de números enteros asignada por una Autoridad que identifica unívocamente el tipo del contenido. El campo *content* indica, como su nombre sugiere, el contenido asociado. Su estructura interna queda determinada de forma precisa por el *contentType*. En la norma se recogen seis tipos de datos: *data* (para referirse a una cadena de octetos cualquiera), *signed-data* (para especificar un mensaje firmado), *enveloped-data*, *digest-data* (para resúmenes hash), *encrypted-data* y *authenticated data*.

Para los objetivos perseguidos en este apartado, nos centraremos en las definiciones que sirven para la especificación de la firma electrónica. La especificación CMS no ofrece una especificación particularizada de lo que, conforme a las definiciones del ETSI, hemos venido llamando ES (desde la estructura básica a las otras diversas extensiones), sino que se centra en la especificación del mensaje firmado (*signed-data*), del cual la firma electrónica constituye una parte.

MENSAJE FIRMADO: EL TIPO SIGNEDDATA

El *ContentType* del *signed-data* consiste en una estructura de datos, parte de cuyos componentes están firmados por cero o más signatarios (el valor cero es puramente formal) que realizan la operación en paralelo. Está definido por el siguiente identificador de objeto:

```
id-signedData OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs7(7) 2 }
```

En el anterior epígrafe ya adelantábamos los campos que constituyen la estructura del **mensaje firmado**. En la especificación del tipo *SignedData* aparecen en el siguiente orden:

- Versión.
- Información sobre algoritmos *hash* utilizados.
- El mensaje *m* que es firmado.
- Información sobre certificados.
- Información sobre revocaciones de certificados.
- Información sobre los firmantes (la misma estructura repetida tantas veces como firmantes haya).

La especificación ASN.1 del tipo *SignedData* es la siguiente:

```
SignedData ::= SEQUENCE {
    version          CMSVersion,
```

```

digestAlgorithms      DigestAlgorithmIdentifiers,
encapContentInfo      EncapsulatedContentInfo,
certificates          [0] IMPLICIT CertificateSet          OPTIONAL,
crls                  [1] IMPLICIT CertificateRevocationLists
                                                                    OPTIONAL,

signerInfos SignerInfos
}
    
```

Cada uno de los campos pertenece a un tipo concreto cuya especificación precisa está recogida en la norma. A continuación pasaremos a explicar su significado añadiendo parte de su especificación ASN.1, lo que nos servirá de ayuda para entender mejor su funcionalidad.

El **primero** de ellos, el campo *version*, indica la versión de la sintaxis. Para que tengan cabida las definiciones que necesitamos para la firma electrónica debe valer 3.

El **segundo de los campos** está destinado a informar acerca de los algoritmos hash utilizados. Se define:

DigestAlgorithmIdentifiers ::= SET OF DigestAlgorithmIdentifier

Cada uno de los elementos del conjunto es el identificador de uno de los algoritmos hash reconocidos que haya sido utilizado por alguno de los firmantes en su operación de firma digital. En principio se cuenta con que sean MD5 o SHA-1, pero a efectos de la especificación valdría cualquier otro que más adelante sea reconocido y que pueda ser referenciado mediante un identificador de objeto. Por ejemplo, el identificador de SHA-1 es:

sha-1 OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) 26 }

(La información acerca de los algoritmos de firma utilizados está incluida dentro de la estructura que recoge la información sobre los firmantes, definida mediante el tipo *SignerInfo*, que se describe en el siguiente apartado.)

El campo **encapContentInfo** representa el mensaje o documento sobre el que se ha realizado la firma digital. Esta información viaja en claro, y consiste en el identificador del tipo de contenido más el contenido propiamente dicho. El tipo de encapsulación se define como:

EncapsulatedContentInfo ::= SEQUENCE {
 eContentType **ContentType,**
 eContent **[0] EXPLICIT OCTET STRING OPTIONAL }**

ContentType ::= OBJECT IDENTIFIER

El tipo de contenido, *eContentType*, está determinado por un identificador de objeto, y *eContent* es el contenido en sí, es decir, la cadena de caracteres mediante la cual se representa el mensaje o documento que se está firmando. Cabe llamar la atención sobre el hecho de que este campo es optativo, en contra de la más elemental concepción de lo que es un mensaje firmado, que exige que siempre el mensaje viaje en claro además de la firma (sea ésta del tipo que sea). La razón de ello es posibilitar la construcción de «firmas externas», o firmas encastradas en cuyo caso el mensaje firmado está ausente en el nuevo proceso de firmado. Más adelante se insistirá en esta situación.

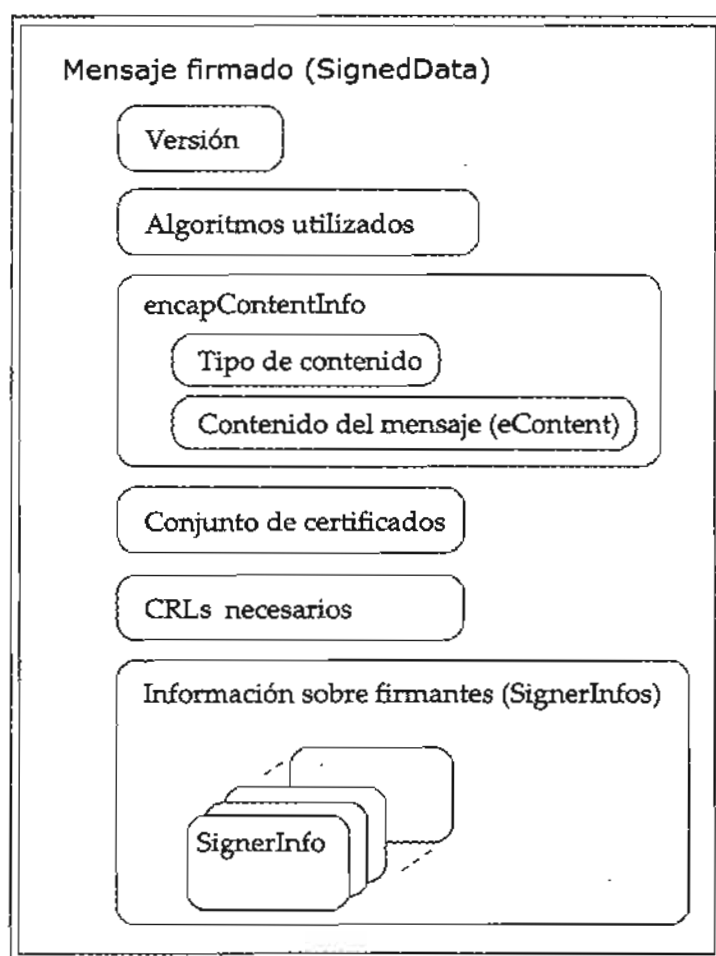


Figura 8.19. Estructura de la especificación del Mensaje Firmado.

El cuarto campo, *certificates*, define un conjunto de certificados que se entiende que son suficientes para que la entidad evaluadora pueda recorrer el camino de certificación hasta encontrar un punto común de confianza (para todos y cada uno de los firmantes). Para ello, en la especificación se define, mediante SET y CHOICE, una serie de alternativas acerca de cómo pueden referenciarse esos certificados. Puede darse el caso de que se reflejen menos certificados de los necesarios si se espera que la entidad evaluadora tenga medios para obtener los que necesite. En cualquier caso, será la política de firma la que deberá regular esas circunstancias.

El quinto campo, *crls*, tiene un significado similar al anterior, sólo que referido a la colección de CRLs necesarias para que pueda conocerse el estado de revocación o suspensión de los certificados que aparecen en el campo anterior. También pueden referenciarse menos CRLs de las que serían necesarias si en el proceso de evaluación puede suplirse esa carencia.

El último de los campos, *signerInfos*, se define como:

SignerInfos ::= SET OF SignerInfo

Es decir, representa un conjunto de estructuras (una por cada uno de los signatarios), cada una de las cuales está a su vez definida por el tipo *SignerInfo*, que pasaremos a describir en el siguiente apartado. En la Figura 8.19 se muestra un cuadro-resumen de la estructura de la especificación del Mensaje Firmado donde se refleja esta situación de multiplicidad de firmantes.

INFORMACIÓN PARA CADA UNO DE LOS FIRMANTES: EL TIPO SIGNERINFO

Como acabamos de decir, este tipo sirve para determinar la información que debe aparecer por cada uno de los signatarios en el caso de que varios de ellos firmen el documento en paralelo. En la especificación del tipo *SignerInfo* aparecen los siguientes campos:

- Versión.
- Información sobre la clave pública del firmante.
- Identificación de los algoritmos hash y los algoritmos de firma utilizados.
- Atributos que son firmados y otros atributos no firmados.
- Firma digital (*Signature Value*) del mensaje y de los atributos que deben ir firmados.

Su definición ASN.1, según la RFC 3369, es la siguiente:

```
SignerInfo ::= SEQUENCE {
    version          CMSVersion,
    sid              Signer Identifier,
    digestAlgorithm  DigestAlgorithmIdentifier,
    signedAttrs      [0] IMPLICIT SignedAttributes OPTIONAL,
    signatureAlgorithm SignatureAlgorithmIdentifier,
    signature        SignatureValue,
    unsignedAttrs    [1] IMPLICIT UnsignedAttributes OPTIONAL
}
```

El significado y estructura de estos siete campos es el siguiente:

El campo *version*, como en el caso anterior, indica la versión de la sintaxis. Para que tengan cabida las definiciones que necesitamos para la firma electrónica debe valer 3.

El campo *sid* especifica la clave pública que se corresponde con la clave privada con la que el signatario ha firmado el documento. A través del tipo *SignerIdentifier* provee dos alternativas:

```
SignerIdentifier ::= CHOICE {
    issuerAndSerialNumber IssuerAndSerialNumber,
    subjectKeyIdentifier  [0] SubjectKeyIdentifier
}
```

La primera opción es indicar el nombre de la CA que ha emitido el certificado y el número de serie que le asignó. La segunda opción es proporcionar la extensión *subjectKeyIdentifier* que (véase epígrafe 6.6) sirve para ayudar a conocer con puntualidad cuál es la clave certificada. El interés de esta segunda opción radica en que el signatario mantenga varios pares de claves distintas, para usos distintos, bajo políticas de seguridad distintas. En este caso la extensión nos ayudará a identificar sin ambigüedad cuál de ellas es la clave pública para la que se ha emitido el certificado.

Los campos *digestAlgorithm* y *signatureAlgorithm* identifican, respectivamente, el algoritmo hash y el algoritmo de firma, junto con los parámetros a ambos asociados.

Los campos *signedAttrs* y *unsignedAttrs* representan, en claro, sendos conjuntos de atributos, de los cuales solamente los del primer grupo formarán parte de la información que se firma. En principio, dentro de la especificación aparecen como opcionales, pero la política de seguridad concreta bajo la que se desarrolle la firma podrá indicar cuáles de ellos deben aparecer obligatoriamente. En el siguiente apartado se

comenta más espaciadamente el significado de los atributos más significativos que pueden aparecer en la información que se aporta por cada signatario. Se definen como:

SignedAttributes ::= SET SIZE (1..MAX) OF Attribute

UnsignedAttributes ::= SET SIZE (1..MAX) OF Attribute

Attribute ::= SEQUENCE {
 attrType **OBJECT IDENTIFIER,**
 attrValues **SET OF AttributeValue**
}

AttributeValue ::= ANY

El campo *signature* es, obviamente, la estrella invitada de la especificación, la firma propiamente dicha, en torno a la cual giran todos los restantes campos, para dejar claro y preciso cuál es la información que se firma y las circunstancias bajo las cuales la operación se realiza. Su especificación es muy simple:

SignatureValue ::= OCTET STRING

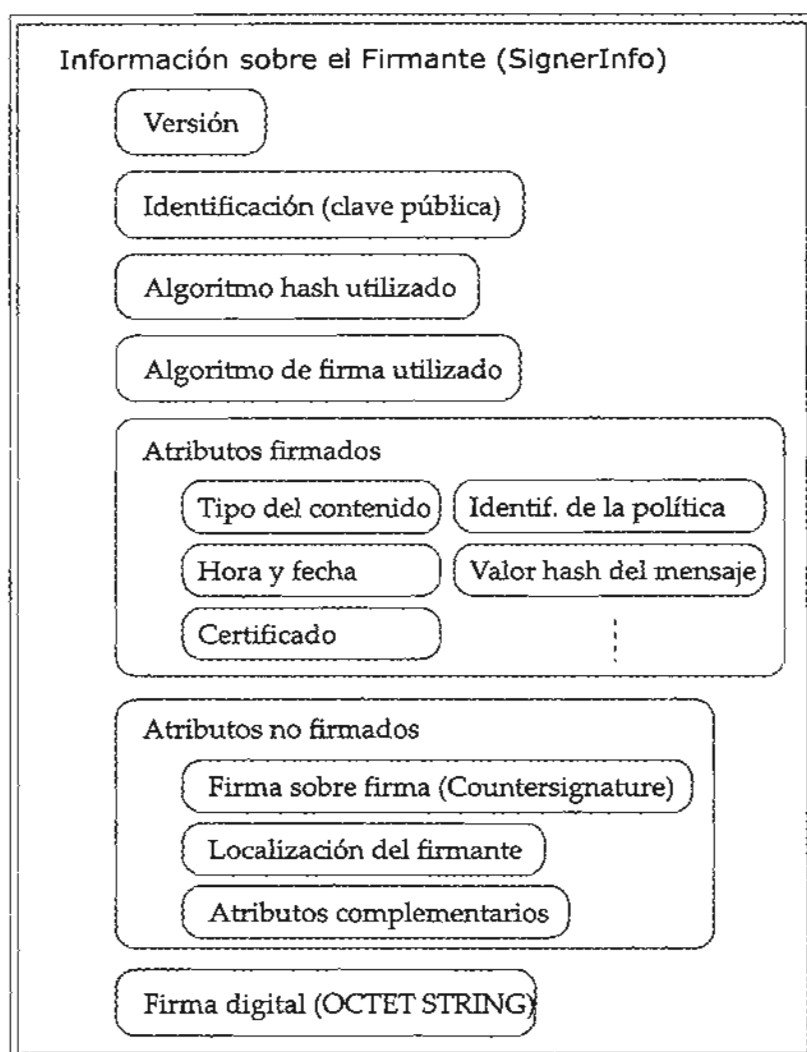


Figura 8.20. Estructura de la especificación de la Información sobre el Firmante.

El *SignatureValue* es el resultado de la generación de la firma digital a partir del resumen del mensaje y de los atributos que se firman, usando la clave privada del signatario.

En la Figura 8.20 se muestra un cuadro-resumen con la estructura de esta definición de la Información sobre el Firmante, en la que se ha alterado ligeramente el orden de los campos para facilitar su representación.

El proceso seguido para el cálculo de la firma es el siguiente: Para empezar, se realizará el cálculo del valor hash del contenido (mensaje *m*) junto con los atributos que se firman. En primer lugar se realizará el cálculo del resumen a partir del contenido del mensaje, concretamente la cadena de octetos mediante la que se representa *eContent*. Solamente el valor de esa cadena se tomará como entrada, sin incluir ni el campo etiqueta ni el campo longitud que aparecerán en su representación ASN.1. Obviamente, si no existiesen atributos firmados, este sería el valor hash resultante.

En caso de existir obligatoriamente atributos firmados, que es la situación que verdaderamente nos interesa, uno de esos atributos será, precisamente, el resumen del contenido del mensaje (compruébese la definición de la ES que asumíamos en el epígrafe anterior y el significado del atributo *MessageDigest* que comentaremos en el siguiente apartado). Por ello, en este caso, el resumen final será el resultado de aplicar la función hash a la codificación DER completa del campo *SignedAttributes*. (La especificación CMS preconiza, para la mayoría de las estructuras que define, la codificación DER.)

Una vez obtenido el resumen, éste se utilizará como entrada del proceso de cálculo de la firma digital, junto con la clave privada del signatario y los restantes parámetros que defina el algoritmo utilizado. El identificador de objeto, junto con los parámetros que son requeridos, se proporciona en el campo *signatureAlgorithm* que antes hemos referido. En principio, se tienen en cuenta solamente los algoritmos RSA y DSA, pero nada impide que se puedan incluir otros que resulten fiables a los actores que intervienen en el proceso de firma y evaluación. A título de ejemplo, el identificador de objeto del algoritmo DSA es:

```
id-dsa-with-sha1 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
x9-57 (10040) x9cm(4) 3 }
```

El proceso de verificación de la firma digital se llevará a cabo obteniendo garantías de la validez de la clave pública asociada a la clave con que se ha firmado, mediante el proceso de análisis de los certificados y CRLs presentados por el signatario (u obtenidos por la entidad verificadora) y de todos los parámetros y atributos que se determinen en la Política de Firma bajo la que se esté operando.

ATRIBUTOS QUE DEBEN IR FIRMADOS

Aunque en la especificación descrita en el apartado anterior se refleja que tanto los atributos firmados como los no firmados son opcionales, para que estas definiciones sean de utilidad para lo que en el anterior apartado hemos denominado firma electrónica básica, ES, se requiere que al menos seis de ellos aparezcan siempre entre los atributos firmados. Además del *identificador del algoritmo de firma*, estos son los siguientes:

1. Tipo del contenido (*ContentType*)

El identificador de objeto de este tipo está recogido en la CMS de la RFC 3369. Sirve para especificar el tipo de la información contenida en el mensaje. Está definido

a su vez mediante un identificador de objeto y ya se comentó su significado al describir el campo *EncapsulatedContentInfo* del tipo *SignedData*.

2. Hora y fecha de la firma (*SigningTime*)

Este atributo está definido también en la CMS de la RFC 3369 y especifica la fecha y hora a la que el signatario dice haber realizado la firma. Está definido para que se pueda elegir entre las dos maneras de expresar la hora que vimos cuando se describió este atributo en ASN.1:

SigningTime ::= Time

**Time ::= CHOICE {
 utcTime UTCTime,
 generalizedTime GeneralizedTime }**

3. Certificado del signatario (*SigningCertificate*)

Este tipo está definido en la RFC 2634 *Enhanced Security Services for S/MIME*, donde se recoge también su identificador de objeto. Se incluye entre los parámetros firmados para prevenir un ataque por sustitución de la información sobre certificación y para permitir un conjunto restringido de certificados en la verificación de la firma. Se define como una SEQUENCE de certificados.

4. Identificador de la Política de Firma (*SignaturePolicyIdentifier*)

Está definido (incluido su identificador de objeto) en la especificación técnica del ETSI antes citada. Dada la importancia de la política de firma bajo la que se lleve a cabo el proceso de firma y validación, se requiere que este atributo esté presente en la firma y se encuentre entre los que son firmados.

La especificación permite no hacer referencia explícita a una definición concreta de política, dejando que ésta quede determinada de *forma implícita*. En caso contrario, que sería la opción más recomendable, se aporta el identificador de la política mediante tres campos:

- El primero de ellos identifica, mediante un OBJECT IDENTIFIER, la versión concreta de política de firma bajo la que se opera.
- El segundo contiene el valor hash de la especificación de esa política, para que no haya ningún género de dudas acerca de cómo estaba definida en el momento de realizarse la firma.
- El tercero recoge un conjunto de elementos calificadores que permiten aportar información a las entidades que acepten o validen la firma en cuestión.

La especificación formal (en ASN.1) de este atributo y una descripción más pormenorizada de la Política de Firma se posponen para ser abordadas en el Capítulo 10, en el que se pretende recoger y sistematizar todo lo relativo a políticas de certificación y firma.

5. Valor hash del mensaje (*MessageDigest*)

Como se comenta en el apartado anterior al describir el procedimiento de obtención de la cadena de octetos que representa el *SignatureValue*, el resumen del mensaje o documento que se está firmando es también uno de los atributos que conviene que sean firmados para introducir mayor robustez y confianza en la validez de la firma

electrónica. Este tipo está definido en la RFC 2630, donde se recoge también su identificador de objeto. Su definición en ASN.1 es bien simple:

MessageDigest ::= OCTET STRING

ATRIBUTOS QUE NO NECESITAN SER FIRMADOS

Un caso particularmente interesante es cuando lo que se desea es firmar sobre un documento ya firmado. En ese caso, al no existir contenido, no debe aparecer el atributo *contentType*, aunque en una firma convencional sea obligatorio. Del mismo modo, el atributo *MessageDigest* que acabamos de definir tiene un significado diferente. Para este tipo de firma se define un atributo específico que pertenece al conjunto de atributos que **no deben** ir firmados. Este atributo es:

1. Firma sobre otra firma (*Countersignature*)

Sirve para especificar una nueva firma sobre el conjunto de octetos resultantes de la codificación DER del campo *SignatureValue* de la estructura *SignerInfo* de un mensaje ya firmado.

Está definido en la RFC 2630, donde se recoge también su identificador de objeto. Su definición en ASN.1 es, aparentemente, muy sencilla:

Countersignature ::= SignerInfo

Es decir, repite la estructura completa del tipo en el cual está incrustado. No obstante, para tener en cuenta las especificidades que antes señalamos, presenta algunas particularidades respecto del tipo *SignerInfo*:

- El conjunto de atributos firmados deberá contener el atributo de resumen del mensaje si tiene otros atributos firmados, pero no necesita, como ya hemos dicho, el atributo del tipo de contenido.
- La entrada para el cálculo del resumen que interviene en la firma es el conjunto de octetos que representan al *SignatureValue* del *SignerInfo* con el cual el atributo de nueva firma está asociado.
- El atributo *Countersignature*, por ser de tipo *SignerInfo*, puede, a su vez, contener otro atributo *Countersignature*, con lo cual es posible construir una estructura de firmas sucesivas tan larga como se desee. A esto es a lo que en el epígrafe 8.7 hemos denominado **firmas encastradas** (*embedded signatures*).

Recordemos el otro tipo de firmas, las **firmas independientes** que se estampan en paralelo sobre el mismo documento; se consiguen mediante el SET de *SignerInfo* que constituye el último campo del tipo *SignedData*.

Pueden añadirse otros nuevos atributos optativos que, dependiendo de la política de firmado que se siga, en algunos casos puede que se exija que vayan firmados. En realidad, tal y como está definida, la sintaxis CMS de la que partimos está totalmente abierta a la inclusión de nuevos atributos que los distintos organismos de normalización y especificación puedan considerar conveniente añadir. A continuación señalaremos de forma resumida dos de ellos:

2. Localización del firmante (*SignerLocation*)

A veces entre las reglas que rigen la política de firmado (o las condiciones exigidas para la firma de un contrato o convenio) está que se consigne información complementaria respecto a la localización del firmante. La especificación del ETSI recoge el identificador de objeto de este tipo, el cual extrae de la Recomendación F.1 del ITU-T. Se define:

SignerLocation ::= SEQUENCE { -- at least one of the following shall be present

countryName	[0] DirectoryString	OPTIONAL,
	-- As used to name a Country in X.500	
localityName	[1] DirectoryString	OPTIONAL,
	-- As used to name a locality in X.500	
postalAddress	[2] PostalAddress	OPTIONAL }

PostalAddress ::= SEQUENCE SIZE(1..6) OF DirectoryString

3. Atributo del signatario (*SignerAttribute*)

Con frecuencia se requiere de la presencia de atributos complementarios acerca de la personalidad de la entidad signataria. Este puede ser el caso del rol o cargo que desempeña en el momento de la firma. Como ya dijimos en el epígrafe anterior, podemos distinguir dos casos: que sea un atributo del cual ella se reclama (*claimed attribute*) o sea algo certificado por una Autoridad de Atributos (AA). Este tipo está recogido en la especificación del ETSI y se define como:

SignerAttribute ::= SEQUENCE OF CHOICE {
 claimedAttributes **[0] ClaimedAttributes,**
 certifiedAttributes **[1] CertifiedAttributes }**

ClaimedAttributes ::= SEQUENCE OF Attribute

CertifiedAttributes ::= AttributeCertificate -- As defined in X.509

ESPECIFICACIÓN DE OTROS FORMATOS DE FIRMA ELECTRÓNICA

La especificación del ETSI sobre formatos de firma electrónica recoge, como ya hemos visto, otras estructuras con parámetros añadidos sobre el formato de la ES básica. Un atributo que es necesario incorporar en la especificación de estas estructuras es el relativo a la determinación del instante de tiempo en el que se lleva a cabo una determinada operación. Se define como:

Atributo de sello de tiempo (Timestamp Attribute)

Conforme a la definición de firma electrónica que hemos asumido, ésta no será totalmente fiable si no lleva incorporado un sello de tiempo proporcionado por una TSA de confianza. Además, en la Política de Firma se deberá recoger el valor máximo permitido para el periodo de tiempo comprendido entre el indicado por el signatario en la firma y el que determina la TTP. En la especificación del ETSI se recoge el identificador de objeto del atributo y su definición en ASN.1. Estos son:

id-aa-signatureTimeStampToken OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 14 }

SignatureTimeStampToken ::= TimeStampToken

El tipo *TimeStampToken* es un tipo genérico que se define en la RFC 3161 *Time Stamp Protocol* reflejando en un campo específico el dato sobre el que se aplica el sello de tiempo. En este caso, representa el valor hash del campo de firma incluido en *SignerInfo*.

En el epígrafe anterior comentamos que la especificación ETSI define, además de la ES y ES-T, la estructura ES-C para incorporar la información completa que se necesita para la validación. Para muchas entidades, el acceso a todos los datos de certificación y revocación cuyas referencias están contenidas en la ES-C puede resultar muy difícil, por lo que puede crearse una estructura complementaria a ES-C que los contenga. La especificación del ETSI propone varios nuevos nombres para estas estructuras dependiendo de si llevan o no marca de tiempo adicional y sobre qué datos está producido el sellado de tiempo. También propone la existencia de otra nueva estructura, denominada ES-A (*Archive Validation Data*) que, como su nombre sugiere, está destinada a ser archivada para su posible recuperación tiempo después de haber sido generada y validada la firma. Para ello sería necesario añadir un nuevo sello de tiempo, usando, si ello es posible, un algoritmo y un tamaño de claves aún más robusto que el utilizado en estampaciones previas.

8.10. VALIDACIÓN DE LA FIRMA ELECTRÓNICA

El valor de una firma debe ser duradero, por lo que una vez que ha sido probada su validez, debe seguir siendo considerada válida meses y años después. Un firmante o un verificador tienen que ser capaces de proporcionar pruebas robustas de que la firma fue creada o verificada durante el periodo de validez de todos los certificados involucrados en el camino de certificación. La firma seguirá siendo válida aunque tiempo después la seguridad de alguna de esas claves hubiese quedado en entredicho. Para garantizar que la firma o la verificación fueron realizadas antes de una posible revocación, la inclusión de sellos de tiempo proporcionados por TSAs fiables es la mejor solución. Una vez realizada la ES, cuanto antes se obtenga la correspondiente ES-C, tanto mejor. Por el mismo motivo es conveniente que la entidad verificadora, una vez comprobada la corrección de todos los datos, añada una marca de tiempo fiable y firme el conjunto de datos resultantes.

Como en otros casos, el principal problema lo introduce la información de revocación. La entidad signataria debe tener confianza de que **en el momento de la firma** ninguno de los certificados involucrados estaba revocado o suspendido. Pero puede ocurrir que esa situación se produzca cuando se lleve a cabo la verificación si ha transcurrido entre ambas operaciones un periodo de tiempo considerable. En estos casos, será la política de seguridad la que determine el grado de validez que debe otorgarse a la firma en cuestión.

Si para asegurarse de la validez de los certificados se utilizan CRLs, el valor de éstas en el momento de la firma o de la verificación debe quedar adecuadamente reflejado. Más problemática resulta la situación cuando lo que se usa es información OCSP, porque las autoridades proveedoras de este servicio no proporcionan información acerca de cuál era el estado en un momento ya pasado.

El caso de la suspensión es aún más escurridizo porque la decisión sobre la validez o no de la firma ha de posponerse hasta que se resuelva la situación transitoria de suspensión, ya que llegado el caso el estado del certificado puede pasar a revocado o a activo. Además, no se sabe con puntualidad en qué momento se va a resolver esa situación de provisionalidad.

Para poder llevar a cabo la verificación de la firma, la entidad verificadora tiene que ser capaz de acceder y analizar la siguiente información:

- El certificado del signatario.
- La cadena completa de certificados que constituyen el camino de certificación hasta alcanzar un punto de confianza común entre la entidad signataria y la entidad verificadora.
- Todo el conjunto de información sobre el estado de revocación de los certificados involucrados.
- Todos los atributos, firmados o no, que matizan la firma.
- La política de seguridad bajo la que la firma ha sido realizada.

Como resultado del proceso de validación de una firma electrónica⁵ puede obtenerse uno de los tres siguientes «veredictos»:

- La firma es válida.
- La firma no es válida.
- La validación realizada es incompleta.

Válida indica, obviamente, que la firma ha pasado con éxito el proceso de verificación y que cumple con todos los compromisos que establece la política bajo la que fue generada.

No válida indica que la firma no cumple las restricciones impuestas por la política de validación de firma. Esta situación puede producirse porque el formato de la firma no es el adecuado, el valor de la firma es incorrecto, alguno de los certificados involucrados en el camino de certificación ha sido revocado, etc.

Validación incompleta indica que las verificaciones del formato de la firma electrónica y del valor de la firma digital no han fallado, pero no se dispone de suficiente información para determinar que la firma sea completamente válida conforme a la política bajo la que ha sido generada. Otra situación que puede haberse producido es que la información adicional incluida en la firma es incompleta o no está disponible para ser comprobada. En estos casos, como resultado del proceso de validación se generará un informe de las carencias para su posible subsanación o para que la entidad receptora de la firma le dé a ésta el valor que considere adecuado.

A modo de recapitulación, conviene reflexionar acerca de la viabilidad de que un usuario corriente esté en condiciones de verificar un documento firmado que ha recibido a través de una comunicación telemática.

- En el esquema más sencillo, el representado en la Figura 8.13, nos fijábamos en las características y cualidades de la firma digital. Ahí contemplábamos sólo dos entidades, la *entidad firmante* y la *entidad verificadora*, que es la receptora del mensaje firmado, y enfatizábamos el hecho de que el proceso de validación de la firma es sencillo y robusto, sin más requisito que el disponer de la clave pública del firmante, o lo que es lo mismo, de un certificado fiable emitido a su favor.

- Posteriormente introdujimos el concepto de *firma electrónica* como un aditamento que no supone mucha novedad conceptual y como algo que no resulta excesivamente complejo (cosa que es totalmente cierta). Como la fortaleza y la seguridad de la firma electrónica recaen en la *firma digital* que constituye su núcleo conceptual, al resumir las características de la firma electrónica se decía que «el receptor del mensaje firmado debe ser capaz de reconocer, de forma fiable, la validez de la firma, bien sea por sí mismo o con ayuda de terceros que actúen como validadores». No obstante, no dudamos en incluir que «la firma electrónica [...] se presta a que cualquier receptor que tenga acceso de forma fiable de la clave pública del signatario [...] puede tener una razonable confianza en la validez de la firma», lo que significa que «... la verificación está al alcance de cualquiera...».
- A pesar de esa poca complejidad conceptual que añade la *firma electrónica*, para su generación y validación sí que vimos, como se refleja en la Figura 8.15, que era necesaria la ayuda de múltiples TTPs integradas en una determinada infraestructura de seguridad. Aunque ya parece claro en qué se materializa lo de *ayuda de terceros*, en este esquema se sigue manteniendo que la entidad receptora es la encargada de establecer las consultas necesarias para completar el proceso de validación.
- Posteriormente hemos venido presentando para la firma electrónica diversas estructuras y definiciones formales. En lo que se refiere a validación, hemos visto en los párrafos anteriores toda la parafernalia necesaria para la determinación de conformidad con la política de seguridad, búsqueda de caminos de certificación, consultas de CRLs, llamadas a servidores OCSP, consultas de atributos generados por AAs, etc. Llegados a este punto cabe preguntarse cuál puede ser la cara que ponga un usuario corriente si recibe un mensaje firmado y para aceptar o no su validez tiene que realizar toda esa retahíla de funciones. Es razonable pensar que pueda llevar a cabo la validación en un escenario sencillo en el que el firmante y el receptor pertenezcan a un mismo entorno organizativo, pero si el firmante es alguien organizativamente distante, lo razo-

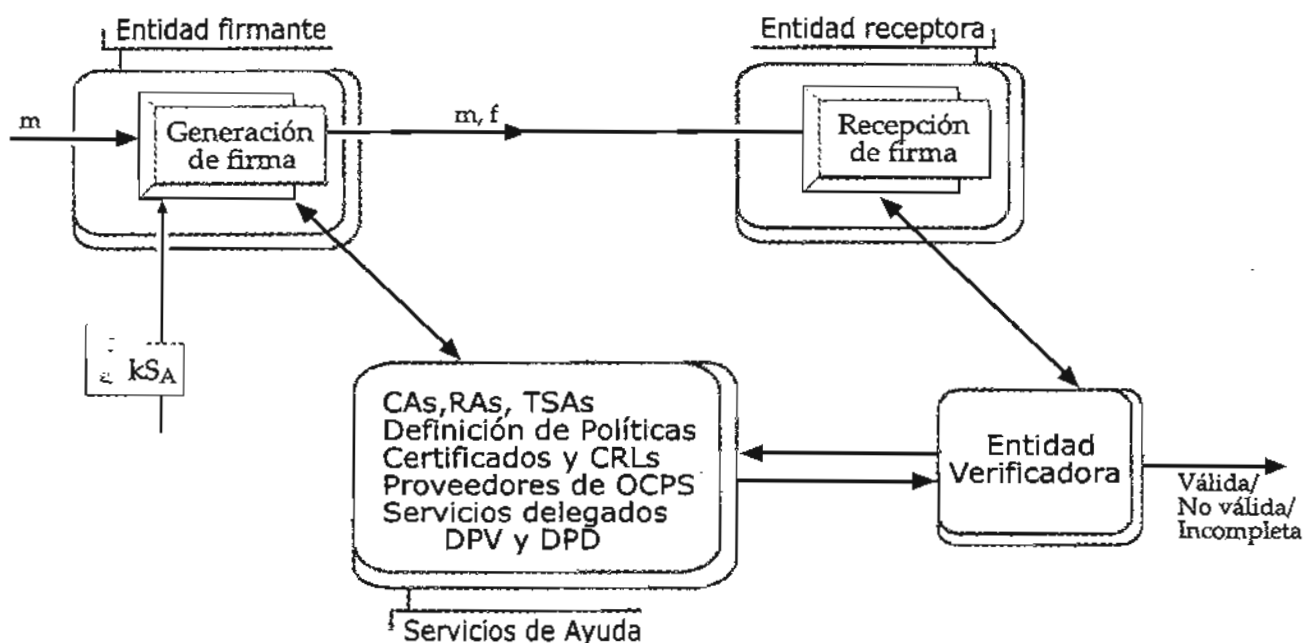


Figura 8.21. Presencia de entidad verificadora en el escenario de firma electrónica.

nable es que la entidad receptora pida ayuda, para que realice el proceso completo, a alguna entidad de su confianza especializada en esas tareas.

Esta última situación es la que trata de reflejarse en la Figura 8.21. En ella se ha previsto la existencia de una **entidad verificadora**, cuyo veredicto es de absoluta confianza para la entidad receptora, que se encarga de todas las tareas de consulta necesarias. Conforme a este esquema, en un determinado entorno organizativo, bien sea una empresa o un organismo público, las distintas entidades comunicantes allí presentes sí se harían cargo de las funciones necesarias para la generación de firma, pero delegarían las tareas de validación en una sola entidad verificadora en la que estarían centralizados estos servicios.

En la Figura 8.21, entre los servicios de ayuda se han incluido los denominados **servicios delegados DPV y DPD**, cuyos acrónimos se corresponden con *Delegate Path Validation* y *Delegate Path Discovery* que están siendo especificados en normativa emergente⁶. Se trata, como su nombre indica, de nuevos servicios que servirían de ayuda para las tediosas tareas de búsqueda de los caminos de certificación y la validación de certificados. La entidad que demande estos servicios debe suministrar al servidor la información necesaria acerca de los certificados que desee evaluar, incluido el identificador de la correspondiente política de verificación. El servidor se encargará de acudir a los repositorios de certificados que crea conveniente y consultar las CRLs y las respuestas de los servidores OCSP que puedan proporcionarle la información requerida.

Conforme a las propuestas de CEN y ETSI, los pasos a seguir en el proceso de validación dependerán del formato de la firma recibida. En principio, para que la firma sea robusta cabe esperar que entre sus atributos firmados aparezca la fecha y hora en la que el firmante dice haber firmado el mensaje, además de un sello de tiempo generado por una TSA fiable (formato ES-T). Si no se cumpliese este requisito (y la política de seguridad allí vigente permitiese esta deficiencia) la entidad verificadora procedería a solicitar un sello de tiempo para añadirse. A continuación obtendría toda la información adicional correspondiente al estado de los certificados involucrados y a su posible revocación. Si la firma ha sido generada añadiendo información sobre certificados y revocaciones (formato ES-C) la entidad verificadora utilizará esos datos y los completará en la medida que sea necesario. En cualquier caso, reconfigurará una nueva ES-C con el resultado de las pesquisas finales y emitirá el veredicto de *válida*, *no válida*, o *incompleta*. En caso de que esté previsto, procederá a firmar y sellar la firma recibida para su correspondiente archivo.

NOTAS AL CAPÍTULO 8

¹ Puede observarse que la fecha de publicación en España del Real Decreto-Ley 14/1999 es anterior a la de la publicación de la Directiva europea. No obstante, la concordancia entre ambas normativas se produce gracias a que las ideas contenidas en la Directiva ya estaban acordadas, en esas fechas, en otras instancias comunitarias. Como consecuencia de ese adelanto, se hizo necesario tramitar la transposición de la Directiva al ordenamiento jurídico del Estado español, situación que se aprovechó para preparar los borradores de lo que debería ser en el futuro la Ley de Firma Electrónica.

² El Real Decreto-Ley de 1999 fue aprobado con la intención de fomentar el crecimiento de la economía y la incorporación de las nuevas tecnologías de seguridad a las comunicaciones telemáticas dentro de las empresas y la Administración Pública, debido a la confianza que esa norma generaría en las transacciones que se hiciesen mediante redes telemáticas. Después

de transcurridos cuatro años, cuando se presentó el nuevo anteproyecto de Ley de Firma Electrónica, poco o nada se había conseguido con la publicación del mencionado Real-Decreto. La Ley de Firma Electrónica se aprobó finalmente en diciembre de 2003. En nuestra opinión, si bien la cobertura jurídica es imprescindible, no lo es menos disponer de estándares conformes con esa normativa (como así lo recoge la Directiva europea) que puedan aplicarse de forma no ambigua. Si la aventura del Real-Decreto ha demostrado que *no por mucho madrugar amanece más temprano*, no hay muchos elementos que hagan pensar que el futuro de la nueva ley vaya a ser más halagüeño. Ello es así porque pocas actualizaciones tecnológicas recoge la nueva ley, cuyo más notorio avance parece consistir en rebajar las exigencias a los prestadores de servicios y rebajar el papel del Estado en favor de la consabida cantinela neoliberal de la «autorregulación del mercado».

³ Con objeto de desarrollar los correspondientes estándares conformes con la Directiva, la Comisión Europea encargó al ICTSB (*Information and Communications Technologies Standard Board*) la tarea de analizar las normas existentes y el desarrollo de las que fuesen necesarias. El ICTSB es una organización creada conjuntamente por las tres entidades de normalización europeas: ETSI, CEN y CENELEC (*Comité Europeo de Normalización de Electrotecnia*). De dicho encargo fue de donde surgió la iniciativa EESSI que trata de reunir los esfuerzos de normalización de diversos actores pertenecientes tanto a la administración como a la industria privada. En la web de la EESSI puede encontrarse referencia detallada de los múltiples documentos elaborados tanto por ETSI como por CEN de cara a la consecución de estándares tanto para los procesos de creación y verificación de firmas como para la determinación de las características que han de cumplir los *dispositivos de creación de firma* a los que hacen referencia los anexos de la Directiva.

⁴ La descripción que aquí se presenta está basada en la especificación técnica ETSI TS 101 733 V1.4.0, de septiembre de 2002, *Electronic Signature Formats*. Esta versión consta de unas 90 páginas, entre las que se incluye un anexo de 18 páginas que contiene la especificación formal completa usando ASN.1 (en su doble versión de X.208 y X.680). En ese extenso documento se recoge de forma minuciosa y detallada la estructura y el significado de los distintos elementos que concurren en la firma electrónica ahí definida. Por ello, la descripción que aquí se presenta es un resumen de los conceptos más importantes que se recogen en dicha norma, siendo responsabilidad del autor tanto la consideración de lo que es más sustancial como las interpretaciones de lo que en ella se especifica. Al igual que en otras referencias a estándares, el lector que necesite hacer uso preciso de dicha especificación deberá recurrir a la fuente original.

⁵ Tanto en la norma ETSI TS 101 733 como en la CWA 1471 *Procedures for Electronic Signature Verification* desarrollada por CEN se contempla la existencia de tres estados de salida posibles (*Valid, Invalid, e Incomplete Validation*) y se describen con cierto detalle los pasos a seguir en el proceso de validación, así como las posibles alternativas en caso de que, como resultado de la verificación, se determine que la firma es válida sólo en parte.

⁶ La RFC 3379 *Delegate Path Validation and Delegate Path Discovery. Protocol Requirements* es un trabajo emergente de cara a la definición de los correspondientes protocolos que han de soportar los servicios delegados. En ella se trata de especificar la información que es necesario suministrar al servidor y el tipo de respuestas que éste debe proporcionar. También existen propuestas de cara al denominado *Delegate Signature Validation, DSV*, cuyas tareas se corresponden con lo que en este texto hemos denominado *Entidad Verificadora*. En cierto modo, OSCP es también un servicio delegado porque trata de aliviar de tareas a la entidad que necesita comprobar la validez de un certificado. Quizás por ello, algunas propuestas emergentes tratan de relacionar estos servicios entre sí. Tal es el caso del documento del PKIX WG *DPV and DPD over OSCP* elaborado por M. Meyer (draft-ietf-pkix-ocsp-dpvdvd, January 2003).

SERVICIOS DE CONTROL DE ACCESO Y NO REPUDIO

De los seis servicios de seguridad que destacamos en el Capítulo 1 hemos abordado ya con cierto detalle los tres primeros: *confidencialidad, integridad y autenticación*. Aunque, en principio, para la provisión de estos servicios no son necesarias TTPs especializadas, la necesidad de obtener garantías acerca de la validez de las claves públicas utilizadas hace que solamente sea posible ofrecerlos con cierta garantía si se apoyan en infraestructuras de certificación robustas.

Asimismo, hemos comprobado que para la obtención de las garantías de autenticación e integridad que proporciona la *firma electrónica* resulta casi imprescindible el concurso de *Autoridades de Sellado de Tiempo*, TSAs, por lo que resulta necesario que exista este tipo de TTPs especializadas dentro de la infraestructura de seguridad en que se apoye el dominio de seguridad en el que se está aplicando el mecanismo de firma.

Continuando con el apoyo que ofrecen las infraestructuras de seguridad que estudiamos en el Capítulo 7, en el presente capítulo se abordan otros dos servicios de seguridad: el servicio de *Control de Acceso* y el servicio de *No Repudio*, dejando pendiente para el Capítulo 11 el servicio de *Anonimato*.

Para la adecuada provisión del primero de ellos es necesario utilizar (en algunos casos) *certificados de atributos*, pertenecientes a las infraestructuras PMI, que también forman parte, como vimos, de las infraestructuras de seguridad. Para el estudio del servicio de *No Repudio* nos apoyaremos fuertemente en las facilidades y garantías de seguridad que ofrece la *firma electrónica* estudiada en el anterior capítulo, al tiempo que definiremos nuevas TTPs especializadas en la provisión de este servicio.

9.1. GENERALIDADES SOBRE CONTROL DE ACCESO

El *Control de Acceso* es un servicio de seguridad que ya estaba presente en los sistemas informáticos aislados, es decir, no interconectados a través de redes. En estos casos, el control de acceso servía, y sirve, para limitar las operaciones o acciones que un usuario autorizado puede llevar a cabo cuando se conecta a un computador. En sentido contrario, el control de acceso sirve para disuadir a los potenciales clientes que pretendan hacer un uso inadecuado de ese computador.

Esta concepción del servicio es heredada y asumida en el ámbito de los dominios de seguridad sobre redes telemáticas, de forma que los protocolos y mecanismos que sustentan el *Servicio de Control de Acceso* sirven para determinar cuál es la utilización permitida de los recursos (sistemas, servidores, otros servicios, etc.), así como para evitar accesos no autorizados. Por tanto, este servicio definirá las piezas de información que esos protocolos necesitan presentar para obtener el acceso y la secuencia de operaciones necesarias para solicitarlo. Como siempre que se trata de la interconexión de sistemas abiertos, el servicio de Control de Acceso en redes telemáticas sólo abarca las operaciones que se realizan de forma remota, no contemplando aquellas que solamente están relacionadas con accesos locales dentro de un sistema.

Así pues, de forma muy resumida podríamos decir que este servicio permite especificar **qué puede hacer quién**. Ese «quién» hay que entenderlo de forma amplia y flexible, ya que puede tratarse de una persona, un agente telemático, un programa, etcétera. Para generalizar, nosotros hablaremos de *entidades comunicantes*. Además, puede darse el caso de que la entidad que pretende realizar el acceso al recurso no lo haga en nombre propio sino en representación de otra entidad que le ha transferido, a los solos efectos del acceso a ese recurso, los derechos que posee.

En el caso más sencillo, se tratará de regular el acceso a un sistema completo. Pero en otros casos será necesario regular pormenorizadamente el acceso a determinados recursos **dentro** del sistema (una vez que se ha superado el acceso global a éste) o incluso a **partes** de ese recurso o a partes de esa parte. Esta cualidad de segmentación selectiva y anidada tanto del recurso como de las operaciones es lo que se conoce como *granularidad*.

Debido a la «veteranía» de este servicio, existe bastante literatura especializada tanto en su definición global como en los mecanismos de seguridad en los que se sustenta. También los organismos internacionales que generan normas acerca de redes telemáticas han abordado esta cuestión¹. Para la descripción del *servicio de Control de Acceso* aquí recogida se ha tenido muy en cuenta, como en otras ocasiones, la formulación proveniente de estos organismos.

AMENAZAS, MECANISMOS Y POLÍTICAS

De forma global, el servicio de Control de Acceso trata de contrarrestar la amenaza que supone el que alguna entidad realice operaciones no autorizadas sobre un recurso de la red, sea éste un servidor, una base de información o un sistema de comunicación. La concreción de esa amenaza puede estar relacionada con:

- Utilización indebida de los recursos de la red a causa de una posible suplantación de personalidad.
- Divulgación o repetición del contenido (*replay*) de un mensaje o parte de él para dirigirlo a un destinatario no autorizado.
- Modificación, alteración o destrucción del contenido de un mensaje o de una información almacenada en un determinado sistema.
- Denegación de un servicio (*denial of service*) a causa de que alguien que acceda indebidamente altere sustancialmente la funcionalidad que debiera ofrecer un determinado sistema o servicio de la red.

Como puede comprobarse, las amenazas que debe contrarrestar el servicio de Control de Acceso abarcan casi todo el espectro de posibles ataques que resumimos

en el Capítulo 1, lo que demuestra la gran importancia que tiene este tipo de protecciones de cara al buen funcionamiento de las redes telemáticas.

Para conseguir estas protecciones y definir los protocolos que las proporcionen es necesario:

- a) El empleo de un conjunto de *mecanismos de seguridad* específicos.
- b) La cooperación con otros servicios de seguridad.

En cuanto a los *mecanismos de seguridad*, cabe decir que la utilización que de estos se haga está directamente relacionada con la *política de control de acceso* que se defina en ese entorno de seguridad. Al igual que ocurre con otros servicios de seguridad, en el caso del control de acceso lo realmente determinante es la política de seguridad que se establezca. La política de seguridad estará constituida por un conjunto de directrices de alto nivel que definan cómo estará controlado el acceso y cómo se tomarán las decisiones que lo determinen. Los mecanismos, en cambio, serán unas funciones de bajo nivel que deben servir para implementar los protocolos y los servicios conforme a lo especificado en la política. A pesar de esta interdependencia, la definición y caracterización de los mecanismos de control de acceso (que abordaremos en el epígrafe 9.2) debe ser llevada a cabo de tal forma que haga que éstos sean fuertemente independientes de la política de control de acceso para la cual son utilizados. Las distintas políticas que pueden presentarse serán estudiadas en el epígrafe 9.3.

Por otra parte, para la adecuada provisión del servicio de control de acceso es muy importante la cooperación con los restantes servicios de seguridad que hemos venido estudiando en los capítulos precedentes. Si bien es cierto que esta cooperación es algo que aparece sistemáticamente en la provisión de cualquier servicio, en el caso del control de acceso esta dependencia se presenta de forma más notoria. De hecho, aunque pueden darse casos en los que no sea así, por lo general la provisión del control de acceso está directamente relacionada con la identidad de la entidad que realiza el acceso al recurso, por lo que casi siempre será necesario realizar previamente una autenticación de esa entidad. En la Figura 9.1, que describiremos con cierto detalle en el siguiente apartado, se refleja, entre otras puntualizaciones, la utilización que el servicio de control de acceso hace del servicio de autenticación y de otros servicios de seguridad.

FUNCIONES DE CONTROL DE ACCESO

El esquema de comportamiento de las funciones de Control de Acceso se recoge en la Figura 9.1. En ella aparece:

- El *iniciador*. Es una entidad comunicante cualquiera. Puede tratarse de un ser humano, una entidad física (por ejemplo, un sistema real) o una entidad lógica (por ejemplo, un programa o un agente telemático).
- El *objetivo*. Se trata del recurso de la red telemática al que se pretende acceder. Puede ser otra entidad, un servidor de información o cualquier sistema de comunicación.
- Las *funciones de Control de Acceso*. Son funciones especializadas que sirven para tomar decisiones sobre el acceso pretendido.

El *iniciador* tendrá asignada cierta información de control de acceso en forma de *privilegios* o *credenciales* que le habrán sido asignados por parte de una autoridad de

seguridad. Por otra parte, el objetivo tendrá definida también una información de control de acceso en forma de *etiquetas* o restricciones acerca de quiénes y bajo qué condiciones están autorizados para usarlo. La forma de asignar estos privilegios y restricciones (y la relación que pueda existir entre ambos) es muy variada y dependerá de la política de control de acceso elegida.

Las *funciones de Control de Acceso* lo que harán es, a partir de la petición de acceso que pretende realizar el iniciador ante el objetivo, tomar las *decisiones* acerca de cuál es la *petición autorizada*. Estas decisiones serán tomadas (Figura 9.1) aplicando las reglas de la política de control de acceso allí establecidas en base a la información de control de acceso asignada tanto al iniciador como al objetivo. Además, para interpretar adecuadamente la aplicación de las reglas definidas en la política será necesario también tener en cuenta cierta información *contextual*, como puede ser la ubicación del iniciador, la hora de acceso o el trayecto de comunicación utilizado.

En la norma de ISO e ITU-T sobre marcos de seguridad para control de acceso se llama ACI, en general, a la *Información de Control de Acceso* (*access control information*). En cuanto a las funciones de control de acceso distingue entre:

- ADF: Función de Decisión de Control de Acceso (*access control decision function*), especializada en la toma de decisiones.
- AEF: Función de Imposición de Control de Acceso (*access control enforcement function*), que impone la decisión tomada por la ADF.

Si trasladásemos este distingo a la Figura 9.1, ello supondría descomponer en dos la caja que representa las tareas llevadas a cabo por las *funciones de acceso* estableciendo una comunicación de decisión entre ellas. Como hemos considerado un poco artificiosa esta descomposición, hemos optado por simplificar la descripción en el sentido que venimos comentando (según se recoge en la Figura 9.1) al tiempo que incluimos la presencia de otros servicios de seguridad.

En cuanto a la cooperación con otros servicios de seguridad, cabe resaltar la estrecha relación existente entre el servicio de Control de Acceso y el servicio de Autenticación, hasta el punto de que, con frecuencia, se produce confusión con res-

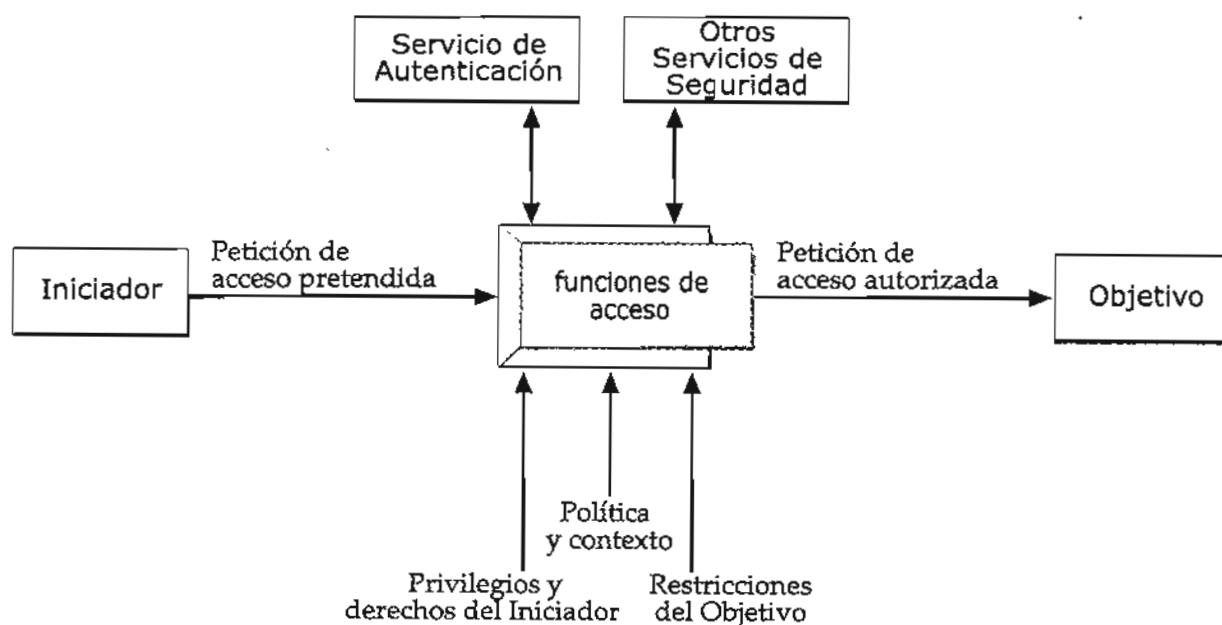


Figura 9.1. Funciones de Control de Acceso.

pecto a la independencia entre ambos. Es cierto que para la aplicación de determinados mecanismos de control de acceso se requiere tener garantías acerca de la identidad del iniciador, pero no es menos cierto que otras veces se puede acceder al recurso de forma anónima.

Un ejemplo de este último caso que se presenta en la «vida real» puede ser el control de acceso que se lleva a cabo para entrar en un cine o en cualquier espectáculo: el pretendiente a espectador lo que tiene que hacer es comprar la entrada o boleto en la correspondiente taquilla de venta y después mostrárselo a la persona que controla el acceso a la sala en cuestión (tener el dinero suficiente para ello es lo que le proporciona la «autorización»).

Situaciones como esta se presentan también en las comunicaciones seguras a través de redes telemáticas. En el sistema Kerberos, según se describió en el capítulo anterior, el cliente accede primero a un servidor de autenticación y a un servidor de credenciales. Una vez obtenida la credencial que le permite el acceso a un servidor concreto, lo único que tiene que hacer es presentarla ante dicho servidor en el momento que desee hacer uso de sus servicios. El control de acceso de ese servidor es lo único que se apoya es en la garantía y robustez de la credencial que le presentan, no en la identidad del cliente que le solicita el servicio.

Otros servicios que sirven para sustentar el servicio de Control de Acceso son el de integridad y el de confidencialidad. El primero de ellos es imprescindible para garantizar que la información de seguridad presente en los distintos componentes que intervienen en el servicio sea exactamente la misma que les asignó la autoridad de seguridad competente, sin que haya sufrido alteraciones malintencionadas por parte de agentes no autorizados. El servicio de confidencialidad es necesario para garantizar que la información de Control de Acceso asignada a las distintas entidades y recursos (que con frecuencia constituye una información muy sensible) solamente sea conocida por los responsables de la gestión de esa entidad o ese recurso.

Además, para una adecuada implantación del servicio de Control de Acceso es muy conveniente recopilar registros de las peticiones realizadas por las distintas entidades que participan en ese entorno telemático para poder llevar a cabo labores de auditoría y contabilidad que permitan conocer, a posteriori, las circunstancias en las que se han permitido o denegado los accesos. Ello permitirá también conocer en qué medida las entidades autorizadas hacen un uso apropiado de los privilegios que les han sido asignados.

9.2. MECANISMOS DE CONTROL DE ACCESO

Para caracterizar los mecanismos de Control de Acceso cabe hacer referencia a un conjunto reducido de *esquemas de control de acceso* que hacen uso a su vez de mecanismos de apoyo más concretos. Podemos distinguir tres esquemas distintos:

- Esquemas basados en Listas de Control de Acceso (*Access Control Lists*, ACLs).
- Esquemas basados en capacidades (*capabilities*).
- Esquemas basados en etiquetas (*labels*).

Entre los mecanismos de apoyo que pueden ser utilizados dentro de cualquiera de estos tres esquemas podemos citar como más significativos:

- Los mecanismos que permiten la *autenticación* de los iniciadores y los objetivos.

- Los *testigos* de seguridad (como puede ser una tarjeta inteligente) que pueden contener información de utilidad para la adecuada provisión del servicio de Control de Acceso.
- Los *certificados* de control de acceso que garantizan (mediante la firma de una autoridad de seguridad) la certeza de la información contenida en determinados atributos. (El epígrafe 9.4 se dedica al estudio del *certificado de atributos*, uno de los mecanismos de mayor relevancia presentes en el control de acceso.)

LISTAS DE CONTROL DE ACCESO (ACLs)

Para obtener una idea simplificada pero clara de lo que son y representan las Listas de Control de Acceso vamos a plantear la *matriz de acceso* vinculada a un conjunto reducido de recursos [SaPi94] tal y como se representa en la Figura 9.2. De una parte aparecen cuatro recursos (que podemos considerar como objetivos de control de acceso) y de otra parte tres iniciadores. En las distintas casillas de la matriz se reflejan las operaciones que cada uno de los iniciadores tiene permitido llevar a cabo sobre los distintos recursos.

Se ha supuesto la existencia de solamente tres operaciones posibles: *leer*, *modificar* y *suprimir* (podemos imaginarnos, por ejemplo, que cada uno de los recursos representa un conjunto de atributos asignados a un determinado sistema o entidad comunicante). Para simplificar más las cosas vamos a suponer que *todo lo que no está permitido está prohibido*, es decir, cada iniciador solamente puede realizar sobre el recurso aquellas operaciones que *explícitamente* se señalen. Echando un vistazo a la matriz se aprecia que el Iniciador A tiene permitido realizar todas las operaciones sobre el Recurso 1 (lo que hace suponer que tiene derechos de propiedad sobre ese recurso) mientras que solamente puede leer el Recurso 2, no tiene permitido realizar ninguna operación sobre el Recurso 3 y puede leer y modificar el contenido del Recurso 4. El Iniciador B solamente tiene acceso al Recurso 2 y el Iniciador C es, de los tres, el que más derechos de acceso tiene, estándole permitido incluso suprimir el contenido de dos de los recursos.

Para reflejar esta situación podemos optar por asignar una **Lista de Control de Acceso (ACL)** a cada uno de los recursos indicando para cada uno de los posibles iniciadores las operaciones para las que está autorizado. En el caso de que dispusié-

	Recurso 1	Recurso 2	Recurso 3	Recurso 4
Iniciador A	◦ leer ◦ modificar ◦ suprimir	◦ leer		◦ leer ◦ modificar
Iniciador B		◦ leer ◦ modificar ◦ suprimir		
Iniciador C	◦ leer	◦ leer ◦ modificar ◦ suprimir	◦ leer ◦ modificar ◦ suprimir	◦ leer

Figura 9.2. Matriz de Control de Acceso.

semos de una matriz de Control de Acceso semejante a la representada en la Figura 9.2, esta opción se corresponde con una representación en columnas de la información de control contenida en dicha matriz. En un caso tan sencillo como este, la implementación del esquema de control de acceso podría consistir en cuatro listas enlazadas tal y como se recoge en la Figura 9.3.

Con estas listas está perfectamente cubierto, y es fácil de llevar a cabo, el control de acceso a cada uno de los recursos. Cuando un iniciador pretenda acceder a uno de los recursos, éste lo que tiene que hacer es proceder a la autenticación del iniciador y posteriormente consultar en su lista los derechos que tiene asignados.

Para la autoridad de seguridad encargada de asignar la información de seguridad en ese entorno resulta también muy sencillo modificar o revocar los derechos de acceso relacionados con un recurso concreto: sólo tiene que modificar la ACL correspondiente e informar a los distintos usuarios o iniciadores del cambio que se ha producido. Por contra, para cambiar los derechos de acceso de un usuario concreto sería necesario revisar y, en su caso, modificar todas y cada una de las ACLs definidas en ese entorno telemático.

En una situación similar a la reflejada en la Figura 9.3, con un elemento de la lista por cada iniciador, cuando el número de usuarios o hipotéticos iniciadores fuese muy

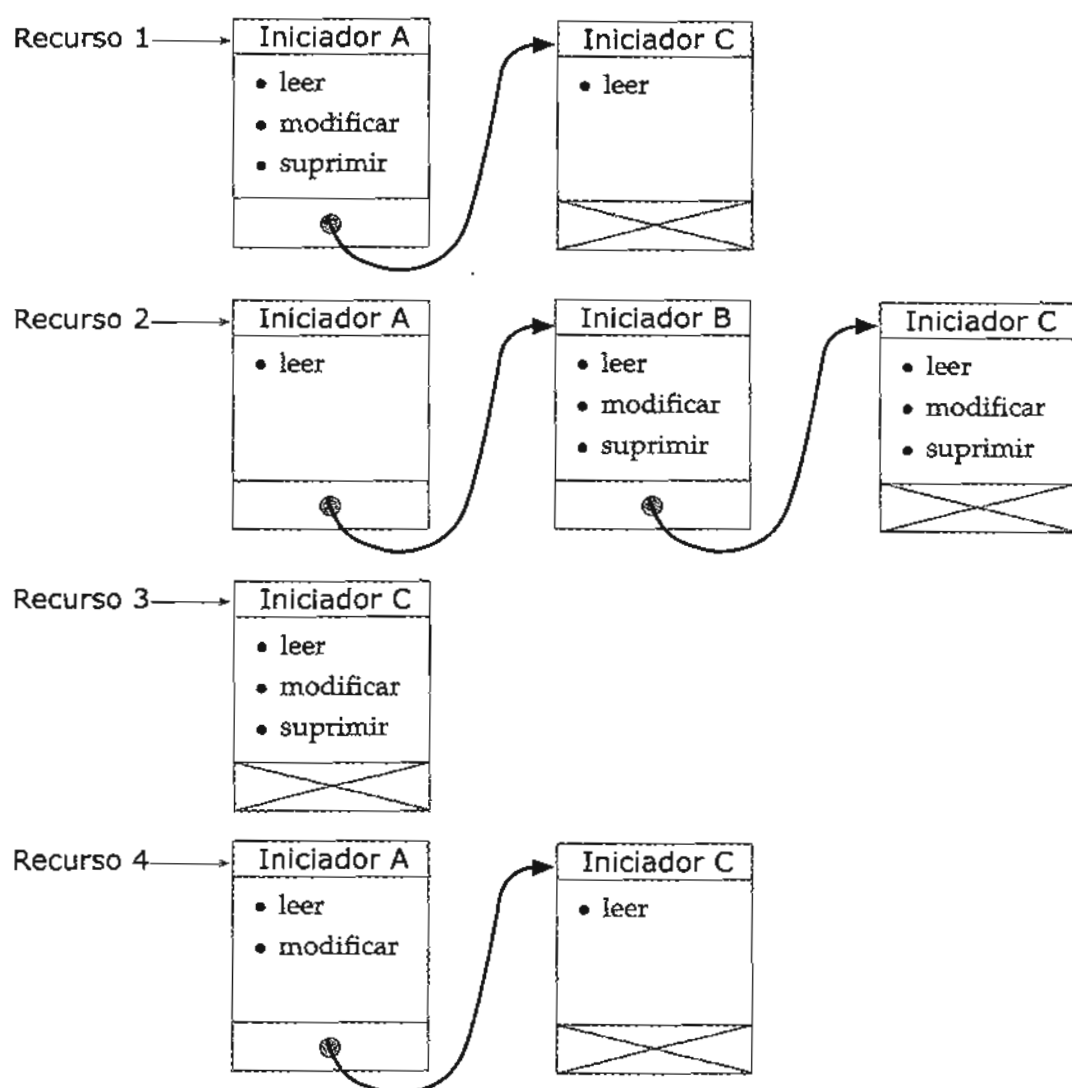


Figura 9.3. Listas de Control de Acceso para los permisos reflejados en la Figura 9.2.

grande, las ACLs serían también excesivamente largas y difíciles de manejar. Por ese motivo, en situaciones reales donde existan muchos hipotéticos usuarios, para que las ACLs sean de utilidad, es conveniente considerar a los usuarios como pertenecientes a determinados grupos (por ejemplo, todas las personas que pertenecen a una organización o departamento) o por categorías o roles que se hayan definido en ese esquema. De esta forma, en las listas de control de acceso no aparecerían muchos nombres de forma individual, sino que los derechos de acceso se asignarían en función del grupo o categoría que el iniciador tuviese, reduciendo así el tamaño necesario de las ACLs asignadas a los recursos.

En estos casos, cuando un iniciador pretenda acceder a un recurso, además de autenticarse debidamente, tendrá que demostrar que pertenece a un grupo establecido o que ostenta determinada categoría o rol. Para ello, sería conveniente que su nombre estuviese construido de forma que reflejase su ubicación organizativa jerárquica (como, por ejemplo, los nombres X.500) o que dispusiese de un documento digital (un certificado) firmado por una autoridad de seguridad solvente en que, mediante atributos de diversos tipos, se garantizase su inclusión en un grupo o categoría concreto.

Es decir, las situaciones reales en las que se aplican esquemas basados en ACLs son mucho más ricas y complejas que la representada en la Figura 9.2. Además de catalogación de los iniciadores por grupos o categorías, es posible hacer referencia a situaciones contextuales como puede ser la hora o la ubicación del punto de acceso. Para definir las ACLs suelen usarse expresiones lógicas que permiten diversidad de matices y particularidades. Así, mediante este tipo de expresiones, determinado permiso puede asignarse a «alguien» que perteneciendo al grupo X no pertenezca al subgrupo X_2 o bien que pertenezca al grupo Y siempre y cuando ostente una categoría W o Z, etc. Es decir, es posible refinar los permisos de acceso de forma bastante más completa que la que se refleja en las casillas que aparecen en la Figura 9.3.

Un caso de estudio interesante para analizar y evaluar la potencia y posibilidades de las ACLs lo constituye el control de acceso definido en el Directorio². Allí (véase Capítulo 7), la información almacenada en la DIB (*Directory Information Base*) está agrupada a través de entradas o inserciones organizadas mediante el DIT (*Directory Information Tree*). Es posible, por ejemplo, que un usuario que accede al directorio tenga derechos para leer (o incluso modificar) algunos atributos de una determinada entrada al tiempo que tiene vedado el acceso a otros. Incluso dentro de un atributo, pueden colocarse ACLs en los distintos valores de ese atributo (por ejemplo, dentro del atributo que recoge el número de teléfono puede darse un acceso amplio al valor que refleja el teléfono del puesto de trabajo, mientras que el valor que recoge el número de teléfono particular puede quedar restringido para que sólo sea accesible para un reducido número de personas). Esta propiedad de refinamiento en los controles de acceso que permite asignar información de control sobre un conjunto de objetivos relacionados entre sí, de forma que sea posible ir refinando las restricciones o permisos a medida que se avanza hacia las distintas partes o subdivisiones de un recurso, recibe el nombre de **granularidad**.

El hecho de que los nombres en el Directorio estén constituidos por componentes (RDNs, *Relative Distinguished Names*) que determinan su posición jerárquica en el DIT facilita el establecimiento de **reglas de precedencia** y el establecimiento de restricciones o permisos sucesivos. Así, resulta sencillo definir un conjunto de ACLs sobre un conjunto de elementos (entradas, atributos, valores) relacionados jerárquicamente de forma que cada uno de ellos pueda **heredar** los controles impuestos al elemento del que jerárquicamente depende pero añadiéndole nuevas restricciones o permisos específicos.

En general, y no sólo para el caso de las ACLs, las *reglas de precedencia* especifican tanto el orden en el que se aplican las reglas de control de acceso como cuáles son las reglas que tienen prioridad sobre otras. Las *reglas de herencia*, como su nombre sugiere, hacen referencia al hecho de que, cuando un elemento está contenido en otro, o es creado copiando parte de la estructura de otro, la información de control de acceso que se le asigne puede establecerse a partir de la información de control de acceso de ese otro elemento del que procede.

En resumen, las características más relevantes de un esquema basado en Listas de Control de Acceso podemos recapitularlas en las siguientes:

- En esta clase de esquemas de control de acceso el establecimiento de restricciones y permisos se realiza en base a cada objetivo, en lugar de hacerlo en base a otorgar derechos a cada uno de los iniciadores.
- Estos esquemas son muy convenientes cuando lo que se requiere es modificar o revocar fácilmente los derechos de acceso asignados a un determinado recurso u objetivo (lo único que hay que hacer es modificar la ACL asignada a ese objetivo).
- Las ACLs son muy adecuadas cuando lo que se requiere es una *granularidad* muy fina en el establecimiento de restricciones y permisos.
- Este esquema es fácilmente implementable siempre y cuando el tamaño de las ACLs no sea excesivo. Esto se consigue si el número de iniciadores no es muy grande, o cuando los iniciadores están organizados en agrupaciones no muy numerosas, o cuando en una misma ACL el número de iniciadores o de grupos que aparecen de forma individual es reducido.

ESQUEMA BASADO EN CAPACIDADES

Podemos decir que el esquema de control de acceso basado en *capacidades* (*capabilities*) es un esquema dual o «simétrico» del basado en ACLs. Esto es, si en este último la información de control de acceso se asigna a los recursos y los iniciadores sólo tienen que probar su identidad o la identidad del grupo al que pertenecen, en el esquema basado en capacidades lo que se hace es asignar a cada usuario o hipotético iniciador una información de control de acceso que indique cuáles son los derechos de acceso que posee ante cada uno de los recursos u objetivos presentes en el entorno telemático de que se trate.

Cuando un iniciador pretenda acceder a un recurso concreto, lo que tendrá que hacer es mostrar los permisos (las *capacidades*) que le han sido asignados y demostrar que quien se los asignó es una autoridad de seguridad con competencias para ello (lo más sencillo será que esa autoridad firme esas credenciales). En la mayoría de los casos cuando acceda a un recurso tendrá que demostrar también su identidad para garantizar que es el titular de esos derechos, pero en otros casos lo único que importa es que sea portador de una credencial que le habilite para el acceso al recurso, con lo cual podría incluso transferir esa credencial a una tercera entidad.

Para fijar la diferencia existente entre el esquema basado en capacidades y el basado en ACLs vamos a suponer un ejemplo de control de acceso [Ford94] en un entorno no telemático. Supongamos la existencia de un determinado organismo de investigación que tiene sus dependencias repartidas entre diversas estancias de un mismo edificio o un conjunto de ellos. Vamos a suponer también que dada la naturaleza de los trabajos que allí se llevan a cabo, el acceso a las distintas áreas está

cuidadosamente reglado para garantizar que sólo puedan penetrar en ellas las personas autorizadas. Una manera de resolver el problema es el colocar una persona en el punto de entrada de cada una de estas áreas con una lista conteniendo los nombres de las personas autorizadas a entrar. Cuando alguna persona desee entrar en una estancia concreta lo que tendrá que hacer es mostrar su tarjeta de identificación a quien guarda la entrada para que éste compruebe que su nombre está incluido en la lista de personas autorizadas: esta sería una solución similar a un esquema basado en ACLs. Otra alternativa consistiría en que a cada empleado se le proporcionase una credencial en la que se especificase a qué áreas o departamentos tiene autorizado el acceso. En este caso, similar a un esquema basado en capacidades, quien guarda la entrada lo único que tiene que comprobar es, además de la identidad, si entre los accesos permitidos en esa credencial se encuentra el área o estancia de cuyo acceso es responsable. Piénsese además que dentro de esta segunda alternativa podría decidirse que para la entrada a determinadas áreas reservadas no fuese necesario autenticarse, sino simplemente mostrar una credencial que autorice para ello «al portador de la presente».

Volviendo al mundo de los entornos telemáticos y al ejemplo simplificado de matriz de acceso que mostrábamos en la Figura 9.2, la asignación de capacidades a los distintos iniciadores allí presentes puede llevarse a cabo mediante un conjunto de listas encadenadas tal y como se muestra en la Figura 9.4, dual en cierta forma del conjunto de ACLs representado en la Figura 9.3.

En el caso de un esquema basado en capacidades, la información de control de acceso contenida en la matriz de acceso se representa por filas, en tanto que en el caso de las ACLs lo que se representan son las columnas.

Como antes apuntábamos, de cara a la aplicación de controles de acceso, las situaciones que se presentan en el mundo real son mucho más ricas y complejas que la que se recoge en la matriz de control de acceso representada en la Figura 9.2, que si bien resulta de interés para aclarar conceptos, representa un escenario excesivamente simple para sacar conclusiones genéricas.

Debido al carácter dual que la solución basada en ACLs representa en relación con la alternativa basada en capacidades, las principales características de este último esquema podemos extraerlas por comparación con las que para las ACLs resumíamos al final del apartado anterior. Estas características son:

- En los esquemas de control de acceso basados en capacidades el establecimiento de restricciones y permisos se realiza en base a otorgar derechos a cada uno de los iniciadores, en lugar de hacerlo en base a los objetivos.
- Estos esquemas no son convenientes cuando lo que se requiere es modificar o revocar los derechos de acceso asignados a un determinado recurso u objetivo, salvo que sea posible identificar de forma individual las capacidades asignadas a todos los iniciadores implicados en ese recurso. En contrapartida, estos esquemas basados en capacidades son muy convenientes cuando lo que se desea es revocar los derechos a un usuario concreto.
- Este esquema es fácilmente implementable siempre y cuando el tamaño de la información sobre capacidades asignada a cada iniciador no sea muy voluminosa. Por ello, este esquema de control de acceso es adecuado cuando existen muchos usuarios o muchos grupos de usuarios diferenciados que acceden a un número limitado de objetivos, o bien que, aunque existan muchos objetivos, el número de objetivos a que acceda cada usuario sea, por término medio, no muy elevado.

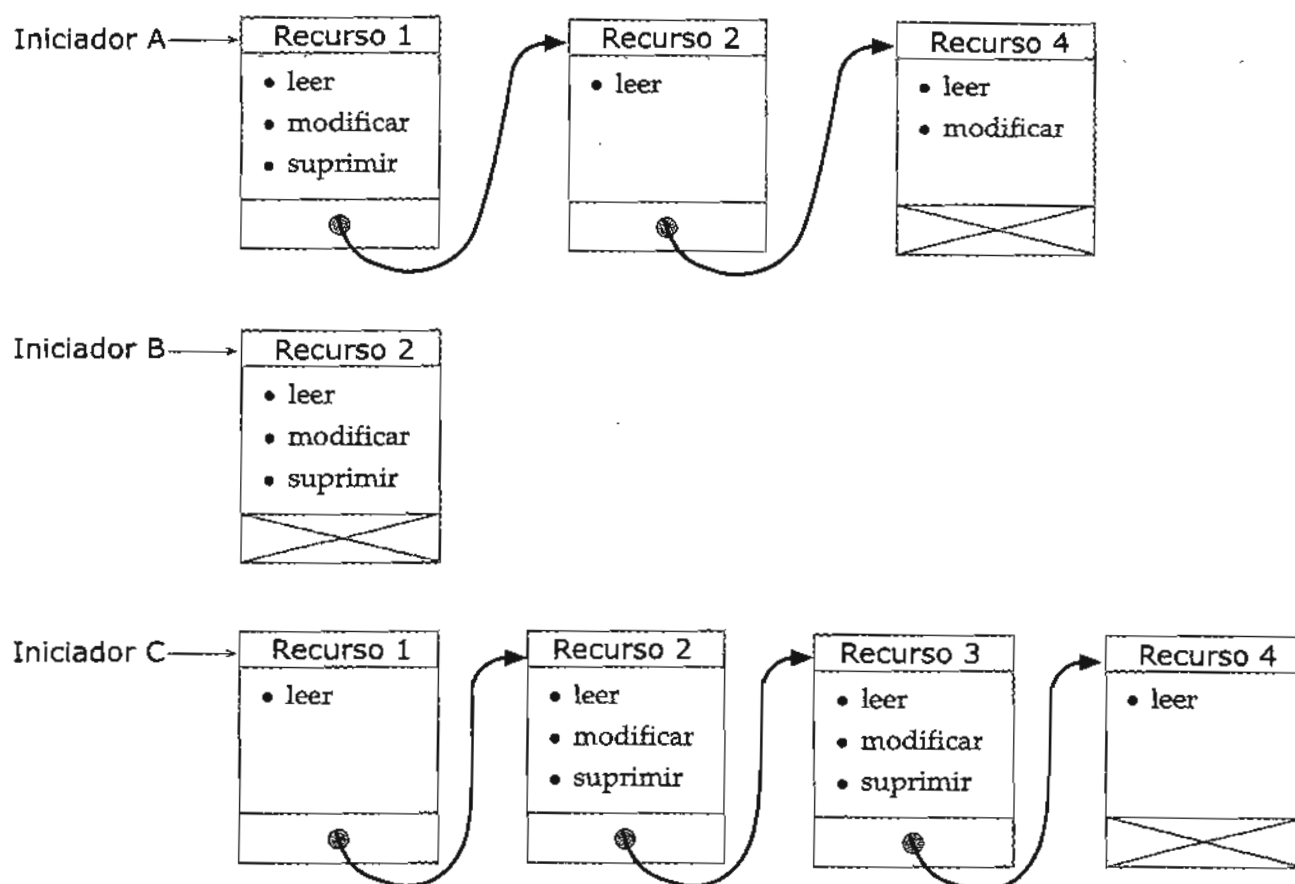


Figura 9.4. Información sobre capacidades para los permisos reflejados en la Figura 9.2.

Con un poco de ligereza algunos autores han aventurado que para los entornos de comunicación telemáticos la solución basada en capacidades es menos adecuada que la basada en ACLs. La variedad que presentan los entornos telemáticos tanto en tamaño como en complejidad geográfica y organizativa es muy grande, por lo que es difícil pronosticar cuáles son, de forma genérica, las mejores soluciones. Bien es cierto que tanto el servicio de Directorio como el FTAM de transferencia de ficheros han optado por las listas de control de acceso, pero eso no significa que sea dado obtener conclusiones generales. Antes bien, la solución que se adopte deberá depender de cada caso concreto.

ESQUEMAS BASADOS EN ETIQUETAS

Estos esquemas se presentan como una generalización de los dos esquemas anteriores, ya que en lugar de poner el énfasis en el ámbito de los recursos o en el de los usuarios o iniciadores, lo que plantean es asignar *etiquetas de seguridad* (*security labels*) tanto a los iniciadores como a los objetivos, así como a los datos o mensajes que se transfieren entre sistemas. El control de acceso se llevaría a cabo mediante la comparación de la información contenida en las etiquetas relacionadas en cada caso.

El énfasis se pone, por tanto, en la *relación* existente entre la información de control de acceso asignada a cada uno de los elementos presentes en la comunicación. En este caso, el ejemplo de la matriz de control de acceso representada en la Figura 9.2 es demasiado simple como para que nos pueda servir, ya que lo que implicaría

sería incluir, redundantemente, tanto capacidades en los iniciadores como listas de control de acceso en los recursos.

En cierto modo, el primitivo mecanismo de contraseña (*password*) sí se podría considerar incluido en este esquema de información relacionada. En efecto, el control de acceso está basado en información de seguridad que comparten el iniciador y el objetivo, así como en el hecho de que el mensaje que transporta esa información se mantenga confidencial para que no pueda ser copiado por terceros no autorizados. Precisamente es en la dificultad de mantener la confidencialidad de la contraseña (sobre todo cuando se pretende cambiar) en donde reside la principal vulnerabilidad de este mecanismo tan inseguro.

En términos generales, una etiqueta o *label* consiste en un conjunto de atributos de seguridad asociados a un «objeto» que puede ser un usuario, un recurso, un mensaje, una petición de acceso o una información de control de acceso que está siendo transferida entre dos entidades comunicantes. Cuando las etiquetas se asocian a usuarios suelen llamarse *acreditaciones* o *credenciales* (*credentials*) y cuando se asocian a recursos de la red se conocen como información de *clasificación*.

Como se ha dicho, el mecanismo de control de acceso basado en etiquetas se puede aplicar también a los datos que son transferidos. Un caso en el que se aplica esto es en el servicio de correo electrónico X.400 (*Sistema de Gestión de Mensajes* o *Message Handling System*, MHS). A cada uno de los *mensajes* que se transfieren se le adjunta una etiqueta que será evaluada en el sistema receptor antes de entregar el mensaje al destinatario. Como es fácil de suponer, la información de control de acceso contenida en las etiquetas de seguridad hace referencia frecuentemente a la política de seguridad que determinará la relación entre el iniciador y el objetivo. Por ese motivo, los atributos definidos para la etiqueta que se adjunta a los mensajes X.400 contienen bastante información relacionada con la política como son, entre otros, el OID (identificador de objeto) de la Política vigente, o el nivel de clasificación del mensaje (no marcado, no clasificado, restringido, confidencial, secreto o muy secreto).

En comparación con los otros dos esquemas (el de capacidades y el de ACLs), dentro del planteamiento de «muchos»/«pocos» se podría considerar como una característica de los esquemas basados en etiquetas el que son muy convenientes cuando existen «muchos» usuarios o «muchos» grupos de usuarios que acceden a «muchos» objetivos y sólo se requiere una granularidad poco fina del control de acceso. Pero esta afirmación tendría que ser muy matizada, ya que lo que realmente ofrecen los esquemas basados en etiquetas de seguridad es una gran versatilidad y multiplicidad de escenarios en los que resulta interesante aplicarlos.

9.3. POLÍTICAS DE CONTROL DE ACCESO

La política de control de acceso estará definida por un conjunto de reglas bajo las cuales los iniciadores pueden tener acceso a los distintos objetivos o recursos presentes en el entorno telemático de que se trate. Como en otros casos, pueden establecerse mediante lenguaje natural en forma de principios genéricos, o bien (como es la tendencia que paulatinamente va imponiéndose) pueden formularse mediante lenguajes formales que permitan que las reglas sean entendidas y cumplidas por parte de agentes telemáticos automáticos sin intervención humana.

Una división «clásica» de las políticas de control de acceso es la que procede de los planteamientos inicialmente formulados en el llamado *Libro Naranja* del Departamento de Defensa de los Estados Unidos sobre criterios de evaluación de seguridad,

aunque posteriormente han sido utilizados en entornos civiles y comerciales. Según esta clasificación, se pueden distinguir tres tipos de políticas:

- **DAC**, *Discretionary Access Control*, que permite a los usuarios garantizar o revocar los derechos de acceso sobre los objetos que se encuentran bajo su control.
- **MAC**, *Mandatory Access Control*, que gobierna el acceso en base a la clasificación de los sujetos (iniciadores) y los objetivos presentes en el entorno.
- **RBAC**, *Role-Based Access Control*, que, como su nombre indica, se basa en la asignación de roles a los usuarios y puede considerarse como una evolución de las dos anteriores³.

Dentro del marco OSI, la tantas veces aquí citada ISO/IEC 7498-2 (idéntica en todo a la Recomendación ITU-T X.800) y las normas de control de acceso que de ellas se derivan (entre otras, la antes citada ISO/IEC 10181-3 (ITU-T X.812)) no utilizan esta nomenclatura que hemos denominado «clásica», sino que proponen la siguiente distinción entre políticas:

- *Política de seguridad basada en la identidad (identity-based policy)*, que se basa en reglas específicas para cada iniciador o grupo de iniciadores.
- *Política de seguridad basada en reglas (rule-based policy)*, en la que, según recoge la norma antes citada, las reglas «se aplican a todas las peticiones de acceso de cualquier iniciador sobre cualquier objetivo en un dominio de seguridad».

Dentro de este marco normativo, las políticas de control de acceso que se establecen en términos de considerar *grupos* de iniciadores o considerar a los iniciadores actuando conforme a un *rol* preestablecido se consideran como casos particulares de las políticas *basadas en la identidad*.

Para Warwick Ford [Ford94], a efectos prácticos, podemos considerar como equivalentes *identity-based* y DAC, así como también serían equivalentes *rule-based* y MAC. Además, dentro de cada una de ellas considera otras nuevas divisiones teniendo en cuenta la existencia de grupos, la formulación de distintos niveles de seguridad o la asignación de categorías a los distintos objetivos. Según este criterio, la política RBAC reúne características de los dos anteriores grupos, esto es, el de las políticas basadas en la identidad y el de las basadas en reglas.

Conforme a ese criterio, para cumplir con los objetivos que aquí nos hemos propuesto, en los siguientes párrafos trataremos de describir de forma muy resumida lo más significativo de cada una de esas tres agrupaciones de políticas: basadas en la identidad (semejantes a DAC), basadas en reglas (semejantes a MAC) y basadas en roles (RBAC). Cabe suponer, además, que las dos primeras representan planteamientos más anticuados y que la tercera es una evolución de las otras dos que satisface mejor los actuales requisitos que demandan tanto las organizaciones gubernamentales como las del mundo de la industria, la ciudadanía y el comercio.

DACS Y POLÍTICAS BASADAS EN LA IDENTIDAD

En la política DAC, la más elemental e insuficiente [SaPi94], se considera a los usuarios como propietarios de los recursos del entorno. Debido a ello, un usuario, o grupo de usuarios, controla el acceso a un determinado recurso, por lo que puede

imponer sobre éste, de forma discrecional, los control de acceso que considere convenientes mediante sentencias o estipulaciones. Lo más sencillo es que estas sentencias se apoyen en el principio de *autorización explícita* de forma que sobre cada recurso se especifique qué operaciones concretas puede llevar a cabo cada usuario (o grupo de usuarios) que hayan demostrado previamente su *identidad* (esto recuerda claramente los planteamientos de las listas de control de acceso, aunque no necesariamente esas sentencias tengan que establecerse aplicando este mecanismo concreto).

La principal objeción a este tipo de planteamientos, en el que son los usuarios los que imponen discrecionalmente los controles, es que en muchas organizaciones los usuarios finales no son propietarios de los recursos a los que acceden, sino que todos estos son propiedad de la organización en su conjunto. Esta objeción podría no afectar a una política basada en identidad si quien impone las sentencias de control de acceso sobre los recursos es una autoridad de seguridad y no los usuarios, lo que evidencia que, si bien las políticas DAC y las basadas en la identidad son similares, no tienen por qué ser en todo coincidentes.

Bajo el principio de autorización presente en las sentencias que aplican el control de acceso bajo las políticas basadas en la identidad, lo que no está permitido está prohibido. Otros planteamientos, llamados de *política abierta*, lo que especifican en las sentencias son prohibiciones en lugar de permisos. En este caso, para cada usuario lo que se especifica es el conjunto de recursos u operaciones a los que se le niega el acceso. Puede plantearse también el uso combinado de autorizaciones y prohibiciones, pero este tipo de interacciones resultan muy difíciles de gestionar y pueden aparecer fácilmente inconsistencias en el sistema de control de acceso global.

POLÍTICAS MAC

En las políticas MAC (*Mandatory Access Control*) cada usuario y cada objetivo dentro del sistema tiene asignados un nivel de seguridad. El nivel de seguridad asociado con el objetivo (*sensitivity*) refleja el grado de perjuicio que puede derivarse de un uso no autorizado del mismo, mientras el nivel de seguridad asociado con el usuario o iniciador (también llamado *acreditación* o *clearance*) refleja el grado de confianza que se ha depositado en él de cara a hacer un uso adecuado de la información y los recursos presentes en el sistema. Por tanto, la información de seguridad asociada a los recursos y a los usuarios establece una cierta *clasificación* de éstos. Por lo general, el nivel de seguridad se representa por un elemento incluido en un conjunto de valores ordenados jerárquicamente (por ejemplo: muy secreto, secreto, confidencial o no clasificado).

Cuando un iniciador pretende acceder a un recurso, dicho acceso queda garantizado solamente si existe una concordancia entre los niveles de seguridad asociados con ambos, en función de la operación de que se trate y de las **reglas** que estén establecidas en la política de control de acceso de que se trate, teniendo en cuenta las condiciones de contexto en las que se lleve a cabo la petición de acceso.

Por ejemplo, cuando de lo que se trata es de acceder a una información clasificada cuya confidencialidad se desea proteger, puede aplicarse una política consistente en que al usuario se le permite leer aquellas informaciones cuyo nivel de seguridad sea igual o inferior al que dicho usuario tiene asignado (*read down*). De otra parte, en lo que se refiere a la «escritura», esta misma política establece el principio de que el nivel de seguridad asignado al recurso predomina sobre el asignado al usuario (*write up*).

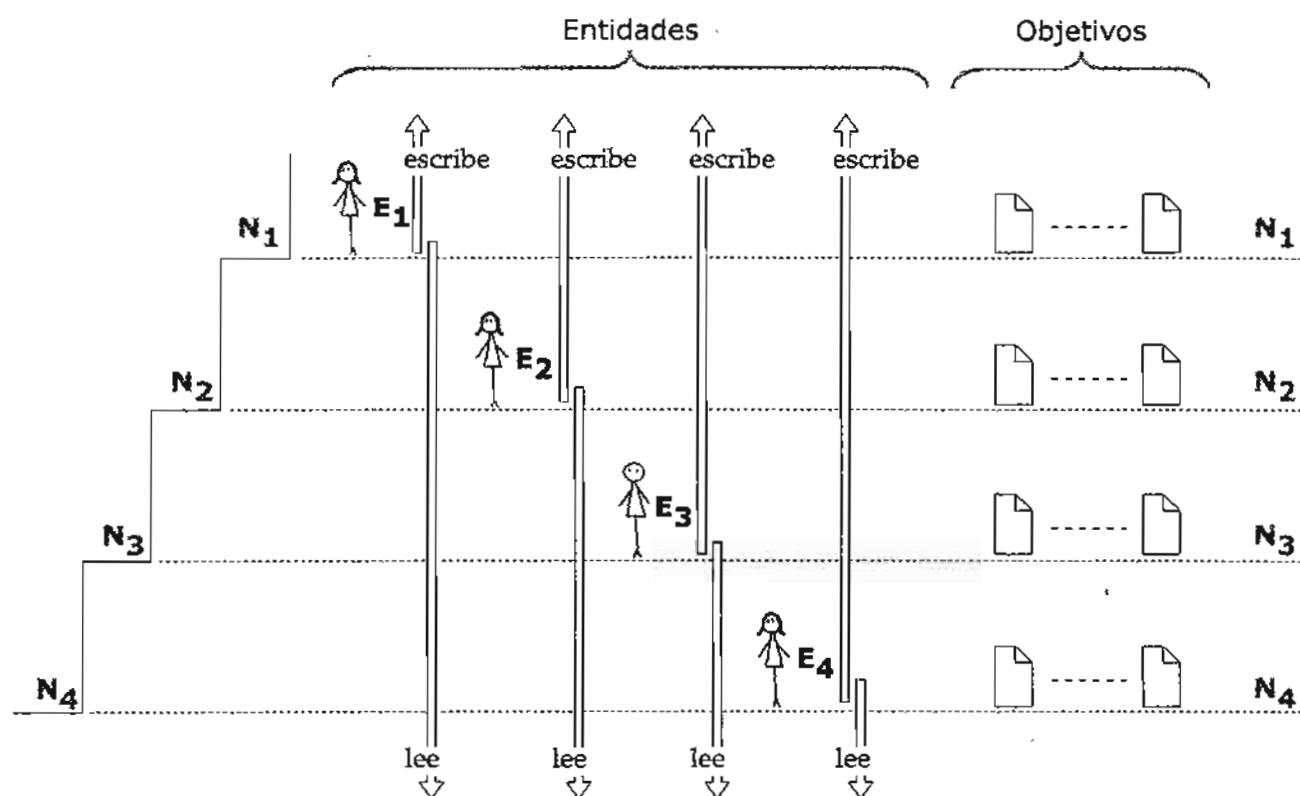


Figura 9.5. Reglas de acceso para garantizar la confidencialidad.

En la Figura 9.5 (que es una adaptación de la representada en [SaPi94]) se ilustra este tipo de comportamiento. En esta figura se representan cuatro entidades usuarias de objetos de información. Para mayor sencillez se ha supuesto que los niveles de seguridad asignados a los iniciadores y a los objetivos son los mismos: cuatro niveles comprendidos entre N_1 , el más secreto, hasta N_4 , que representa el menor nivel de confidencialidad. Si nos fijamos en la entidad E_2 , la regla de lectura hacia abajo lo que significa es que puede leer todos los objetos situados en su mismo nivel, el N_2 , o en niveles inferiores, pero no puede leer los clasificados con N_1 , el nivel más alto de secreto.

En cuanto a la escritura, la regla de permisión de escritura hacia arriba lo que conlleva es que la entidad E_2 puede crear (escribir) objetos y ubicarlos tanto en su nivel como en el nivel inmediatamente superior. Como no puede leer los objetos previamente existentes clasificados como N_1 (ni tener conocimiento de ellos) ese permiso de escritura no le permitirá añadir información en ninguno de ellos, lo que supondría, de hecho, su destrucción. No obstante, puede resultar algo contradictorio el hecho de que la entidad E_1 no esté capacitada para escribir en niveles inferiores al suyo. Esta contradicción puede superarse permitiendo, por ejemplo, que la entidad E_2 a la hora de crear (escribir) objetos nuevos pueda ostentar su nivel de seguridad (el N_2) o cualquier otro de los niveles inferiores al suyo, con lo cual sí que podría ubicar objetos en los niveles inferiores. Para seguir protegiendo la confidencialidad, las reglas de *write up* pueden imponer que cuando se trate de objetos clasificados como N_2 , que previamente la entidad E_2 ha leído, no le esté permitido transferir (escribir) esos mismos objetos a niveles inferiores: para ello la marca de N_2 que tengan esos documentos tiene que estar protegida contra borrado o manipulación.

Si de lo que se trata es de proteger la integridad de la información, se podrían establecer otros niveles de seguridad (relacionados, por ejemplo, con la importancia

del documento) y establecer nuevas reglas que en este caso podrían ser del tipo *read up* y *write down* que permitirían a una entidad iniciadora leer documentos con nivel de integridad superior al suyo, pero no escribirlos.

Lo que acabamos de describir debemos interpretarlo a título de ejemplo sin poner excesivo énfasis en los detalles, ya que resulta claro que en una situación real las reglas tendrían que ser menos simples, contemplar diversos casos y matices en las operaciones y tener en cuenta condiciones de contexto. Para lo que sí sirven estos razonamientos es para clarificar el significado de que tanto los iniciadores como los objetivos estén calificados mediante niveles de seguridad y el papel que juegan las reglas definidas en la política a la hora de establecer la relación de concordancia que tiene que existir entre esos niveles asignados.

POLÍTICAS RBAC

En las políticas de control de acceso basadas en roles, **RBAC** (*Role-Based Access Control*) las decisiones de acceso a la información y los recursos se toman en base al rol que el usuario tiene asignado como miembro de una organización. La definición de roles en una organización concreta se llevará a cabo mediante el análisis pormenorizado de la forma de operar de dicha organización a partir de las distintas tareas que tienen asignadas todos y cada uno de sus miembros.

Un rol quedará definido como un conjunto de actuaciones, tareas y responsabilidades asociadas con una actividad concreta. Consecuentemente, RBAC [SaPi94] [SCFY96], en lugar de especificar los derechos de acceso que tiene asignado cada usuario en particular, lo que hace es que las autorizaciones sobre los distintos objetivos se especifican mediante roles. Esta forma de ver las cosas simplifica bastante las definiciones que es necesario hacer y se adapta bastante bien a las necesidades detectadas en muchas organizaciones, sobre todo cuando de lo que se trata es del acceso a informaciones. Evidentemente, en entornos más abiertos y menos jerarquizados, en los que sea muy difícil catalogar a los hipotéticos usuarios mediante roles, este tipo de técnicas tienen menos cabida, siendo necesario recurrir a otras más clásicas y sencillas.

Las características más significativas de este tipo de políticas se pueden resumir en las siguientes:

- a) *Distintos roles.* Todo usuario tiene que tener asignado, al menos, un rol. Pero será muy frecuente que un determinado usuario pueda tener asignados varios roles dependiendo de las distintas tareas que pueda llevar a cabo. Del mismo modo, varios usuarios distintos tendrán asignado el mismo rol (en eso se basa esta política).
- b) *Derechos de acceso para cada rol.* A los efectos de control de acceso, un usuario que tenga asignados varios roles, cuando desee llevar a cabo una operación sobre un recurso protegido lo hará exhibiendo uno de los roles que tiene asignados, ya que los derechos de acceso sobre ese recurso estarán definidos solamente en función de roles (esa es la gracia de esta propuesta). Los gestores del control de acceso están capacitados para regular qué rol puede llevar a cabo qué acción, cuándo, en qué orden y bajo qué circunstancias.
- c) *Jerarquía y especialización de roles.* Los roles definidos pueden ser disjuntos, es decir, pueden tener asignadas tareas y actuaciones totalmente distintas unas

de otras. Pero puede darse el caso de que al asignar responsabilidades a varios roles distintos sea necesario definir para cada uno de ellos un conjunto común de operaciones y privilegios. En este caso, lo que procede hacer es definir un rol con todas los atributos comunes y establecer una jerarquía de roles de tal forma que los otros roles especializados posean, además de los propios, los privilegios asignados al rol del que proceden. Por ejemplo, en una organización universitaria, un rol puede ser «profesor» y tener asignados unos privilegios; para definir el rol «profesor de matemáticas», «profesor de electrónica» y «profesor de telemática» no es necesario repetir los atributos del rol «profesor», sino definirlos como procedentes de éste y añadirles los privilegios particulares de cada uno de ellos.

- d) *Herencia de privilegios.* Pueden definirse roles que hereden los privilegios de otros roles previamente establecidos (las facilidades de herencia y jerarquía de roles sólo estarán presentes en el modelo jerárquico de RBAC).
- e) *Separación de obligaciones.* Existen opciones (*constrained RBAC*) que permiten establecer restricciones en la asignación de roles (permanentemente o de forma dinámica), de manera que ciertos roles puedan declararse como incompatibles, lo que conlleva que un mismo usuario no pueda poseer esos roles simultáneamente. Por ejemplo, ante los resultados de un examen, una misma persona no puede tener al mismo tiempo los roles de alumno (examinando) y profesor (examinador).
- f) *Principio del menor privilegio.* En el caso de que un usuario posea distintos roles, cuando pretenda realizar una operación sobre un recurso no debe usar más privilegios que los estrictamente necesarios para llevar a cabo esa tarea, es decir, debe exhibir el mínimo rol imprescindible. Este principio del *menor privilegio* (*least privilege*) sirve para mejorar la gestión y minimizar riesgos. Por ejemplo, si en un departamento universitario está definido el rol «miembro del departamento» que poseen todos los estudiantes y personas que trabajan allí, si el director o directora de ese organismo quiere leer un documento que está a disposición de todos los miembros de ese departamento, no debe usar el rol «director» para acceder a esa información, sino simplemente el rol «miembro del departamento» que también posee.
- g) *Facilidad en la gestión de las autorizaciones.* A la hora de gestionar la asignación de autorizaciones a los usuarios, una ventaja de este tipo de políticas consiste en la separación que hace, de una parte, en la asignación de usuarios a roles y, de otra, en la asignación de derechos de acceso a los distintos roles. De esta manera, si un usuario cambia de actividad y de privilegios, no es necesario revocarle los derechos de acceso que tuviese antes y sustituirse los por otros nuevos, sino que lo que se necesita es asignarle el nuevo rol (con todas las herencias y dependencias jerárquicas que ya tenga establecidas).
- h) *Clasificación de los objetos y recursos.* De igual forma que se permite una catalogación de los usuarios mediante roles, la política RBAC posibilita clasificar también los recursos a los que se accede. Esta clasificación puede ser llevada a cabo en función del tipo de objeto de que se trate (documento, base de información, servicio de comunicación, sistema físico, etc.), o bien en función de las distintas aplicaciones que puede tener. El establecimiento de *clases de objeto* (que pueden a su vez subdividirse jerárquicamente) permite que la asignación de autorizaciones a los roles se haga en base a clases de objeto y no solamente en base a objetos específicos. Por todo ello, esta política se adapta bien a sistemas y aplicaciones en los que se usan tecnologías

orientadas a objeto, en las que el concepto de operación RBAC se puede asociar al concepto de «método» presente en esas tecnologías.

- i) *Facilidades de distribución.* En sistemas distribuidos, la protección y las responsabilidades de gestión pueden dividirse entre un dominio central y otros dominios locales, de tal forma que se puede definir una política de protección global y dejar a las organizaciones locales la responsabilidad sobre asuntos de su incumbencia.

Todas estas propiedades y ventajas hacen que las políticas RBAC tengan una aceptación creciente en muy diversos entornos, tanto telemáticos como en sistemas aislados, y que las actividades de normalización vayan concretando cada vez más las características, modelos y extensiones de este tipo de políticas y sistemas de control de acceso.

9.4. CERTIFICADOS DE ATRIBUTOS PARA EL CONTROL DE ACCESO

En el epígrafe 7.9, dentro del capítulo que hemos dedicado a las infraestructuras de seguridad, se ha definido ya el concepto de *Certificado de Atributos* y el papel que juega esta pieza segura de información en el contexto de las Infraestructuras de Gestión de Privilegios (PMIs).

Allí centramos nuestra atención en la infraestructura PMI y en la semejanza y relación que tiene con la infraestructura PKI «clásica». En el presente epígrafe nos centraremos en la información que puede aportar dicho certificado a través de los distintos campos que posee. Para ello comentaremos su especificación formal y discutiremos sobre las extensiones y los distintos tipos de atributos que pueden incluirse.

Aunque el certificado de atributos puede también servir para la provisión de otros servicios de seguridad, la principal intención perseguida con la incorporación de este nuevo tipo de certificado se centra en la aportación de información que sirva para facilitar los procesos de *autorización*. Es decir, los *certificados de atributos* sirven para aportar información de control de acceso que se puede asignar a los distintos elementos que intervienen en la obtención de autorizaciones para realizar determinadas operaciones sobre un recurso.

También destacamos en el epígrafe 7.9 la posibilidad de incluir en el certificado de atributos una conexión con uno o varios certificados de clave pública que, como ya hemos visto reiteradamente, constituyen un mecanismo muy potente para garantizar la autenticación de la entidad que pretende llevar a cabo el acceso sobre un recurso.

En realidad, los certificados de atributos son mecanismos concretos que pueden ser aplicados en cualquiera de los tres tipos de política que antes hemos analizado. En efecto, en las políticas tipo DAC los certificados de atributos pueden servir para asignar autorizaciones o prohibiciones a los distintos iniciadores presentes en el entorno telemático y en las políticas tipo MAC pueden servir para asignarles los niveles de seguridad requeridos. En el caso de RBAC⁴ los certificados de atributos pueden servir para asignar roles a un usuario (cuando éste es el titular del certificado) o para asignar permisos a un rol (cuando el rol es el titular del certificado).

ESPECIFICACIÓN FORMAL DEL CERTIFICADO DE ATRIBUTOS

En la Figura 7.19, que para facilitar su lectura representamos de nuevo como Figura 9.6, se recoge, de forma simplificada, la estructura de un certificado de atributos, que tiene bastante parecido con la de un certificado de la clave pública (al que venimos denominando también Certificado X.509).

En la RFC 3281, titulada *An Internet Attribute Certificate Profile for Authorization* [FaHu02], se describe formalmente su estructura (definida previamente, como ya dijimos, en la Recomendación X. 509 de ITU-T), se proponen elementos para configurar un perfil realista y se discuten diversos aspectos relativos a este tipo de certificados. La especificación formal es la siguiente:

```

AttributeCertificate ::= SEQUENCE {
    acinfo                AttributeCertificateInfo,
    signatureAlgorithm     AlgorithmIdentifier,
    signatureValue         BIT STRING
}

AttributeCertificateInfo ::= SEQUENCE {
    version               AttCertVersion -- version is v2,
    holder

```

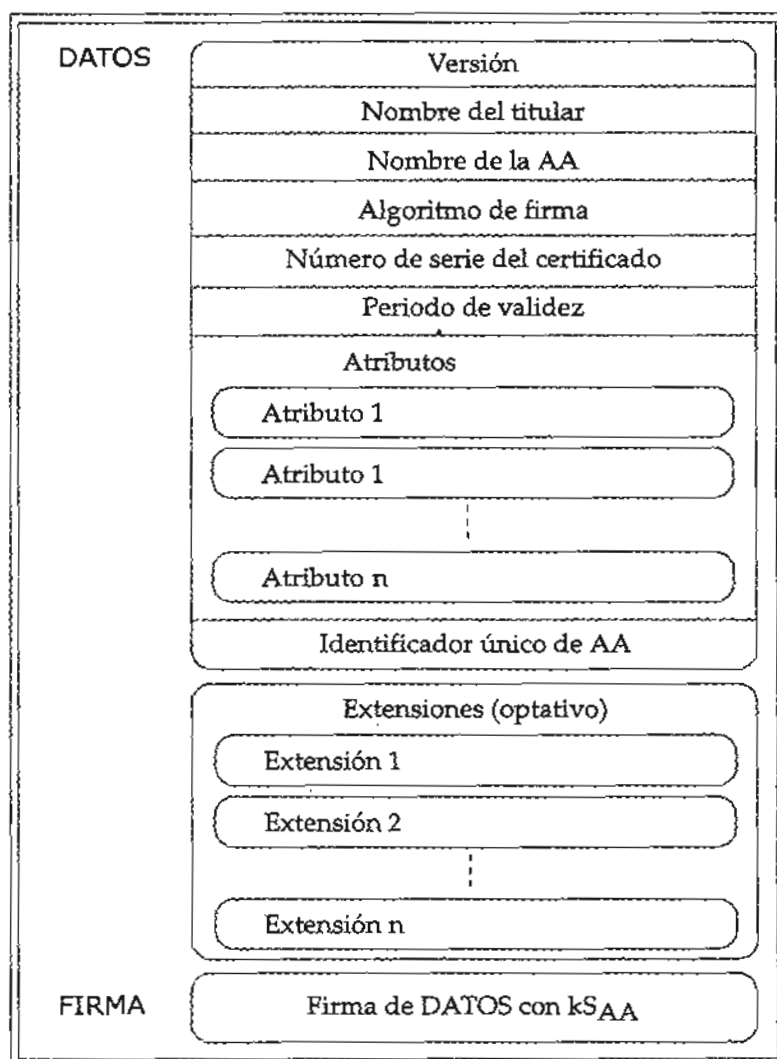


Figura 9.6. Estructura de un Certificado de Atributos.

issuer	AttCertIssuer,
signature	AlgorithmIdentifier,
serialNumber	CertificateSerialNumber,
attrCertValidityPeriod	AttCertValidityPeriod,
attributes	SEQUENCE OF Attribute,
issuerUniqueId	UniqueIdIdentifier OPTIONAL,
extensions	Extensions OPTIONAL

}

Como puede comprobarse, en primer lugar nos presenta el certificado de atributo como una secuencia de tres elementos: contenido del certificado, el algoritmo de firma y la firma de dicho contenido por parte de la Autoridad de Atributos.

Pasemos a comentar los distintos campos del tipo **AttributeCertificateInfo**:

El campo **versión** es del tipo **AttCertVersion**, que es a su vez del tipo **INTEGER**, correspondiendo un 0 para la versión 1 (v1) y un 1 para la versión 2 (v2). Como se especifica en el perfil definido en la RFC 3281, se impone que el certificado sea v2 porque la v1 se corresponde con la X.509 de 1997, que es muy insuficiente para las necesidades posteriormente detectadas e incompatible con la especificación aquí presentada (que sí es compatible con la X.509 del 2000). En el certificado de la clave pública, la compatibilidad entre versiones es muy importante porque el certificado debe tener un periodo de vigencia bastante grande (algunas legislaciones así lo imponen), pero en el certificado de atributos esta circunstancia tiene poca relevancia, ya que se supone que estos certificados son expedidos para regular el control de acceso a servidores y servicios dependiendo de situaciones muy coyunturales y, por tanto, con periodos de validez no muy largos.

El campo **holder** sirve para identificar a la entidad a cuyo nombre se ha expedido el certificado. Se define como una secuencia de tres elementos optativos:

```
Holder ::= SEQUENCE {
    baseCertificateID          [0] IssuerSerial OPTIONAL,
        -- nombre de la CA emisora y numero de serie del
        -- Certificado de la Clave Publica del titular
    entityName                 [1] GeneralNames OPTIONAL,
        -- nombre de usuario o nombre de rol
    objectDigestInfo           [2] ObjectDigestInfo OPTIONAL
        -- usado para autenticar directamente al titular (holder),
        -- por ejemplo, un ejecutable
}
```

Este campo tiene bastante interés porque representa la ligazón con la identidad del titular, lo que conlleva una posible relación con los certificados de la clave pública (certificados de identidad) y, por tanto, con las infraestructuras PKI. Aunque en la RFC 3281 los comentarios de la especificación están en inglés, en la definición de este y otros campos que aquí presentamos se ha optado, como en otras ocasiones, por incluir los comentarios en español (sin eñes ni acentos), quedando bajo nuestra responsabilidad lo acertado o no de dicha adaptación. Tal y como está definido (no es un **CHOICE**), podría darse el caso de que apareciesen más de una de las opciones, lo que podría conllevar cierta confusión acerca de cuál de ellas es la predominante. Por este motivo (aunque la presencia de más de uno de estos identificadores aporta versatilidad y potencia a la definición) se recomienda usar solamente una de las opciones posibles. La RFC 3281 impone que cualquier protocolo que siga el perfil ahí definido debe

especificar claramente cuál es la opción que utiliza y cómo ello se relaciona con el esquema de autenticación que dicho protocolo utilice.

En la opción *baseCertificateID*, el tipo *IssuerSerial* está constituido a su vez por una secuencia de elementos que identifican, cada uno de ellos, un certificado de la clave pública (certificado X.509) emitido a nombre del titular (*holder*) de este certificado de atributos. Esto es:

```
IssuerSerial ::= SEQUENCE {
    issuer          GeneralNames,
    serial          CertificateSerialNumber,
    issuerUID       UniqueIdentifier OPTIONAL
}
```

Los dos primeros campos identifican a la CA emisora de ese certificado de la clave pública y al número de serie correspondiente. El tercer campo, que es optativo, se añade si ese certificado de la clave pública contiene un campo de identificación única (*issuerUniqueID*), en cuyo caso los valores de los campos primero y tercero deben ser coincidentes.

Siguiendo con la definición del campo **holder**, la opción *entityName* representa un nombre identificativo del usuario titular o del rol al que se le expide un certificado de atributos, que puede ser cualquiera de las posibilidades que ofrece el tipo *GeneralNames*. En este caso, si los mecanismos de autenticación que se vayan a usar en el control de acceso están basados en el uso de certificados de la clave pública, el nombre que aparece en *entityName* debe coincidir con uno de los nombres del titular de dicho certificado. (Recuérdese que, según dejamos constancia en el apartado «El campo de extensiones» dentro del epígrafe 6.6, si el titular de un certificado X.509 no tiene nombre distintivo X.500, o éste no aparece en el certificado, se exige que exista la extensión *subjectAltName*.) Al margen de que la entidad titular del certificado de atributos pueda tener más de un nombre, si se usa la opción *entityName* sólo puede usarse un único nombre para identificar al *holder*.

La tercera opción que la norma ofrece para el campo **holder** es mediante el campo *objectDigestInfo*. Esta opción ha sido diseñada para dar satisfacción a los requisitos presentes en algunos entornos, en los cuales no interesa ligar directamente el certificado de atributos ni a un nombre (*entityName*) ni a un certificado de clave pública (*baseCertificateID*). La idea consiste en ligar el certificado a un objeto mediante la colocación del valor *hash* de ese objeto en el campo **holder**. Su especificación es como sigue:

```
ObjectDigestInfo ::= SEQUENCE {
    digestedObjectType      ENUMERATED {
        publicKey           (0),
        publicKeyCert       (1),
        otherObjectTypes    (2) },
        -- en este perfil NO DEBE
        -- ser usado otherObjectTypes
    otherObjectTypeId      OBJECT IDENTIFIER OPTIONAL,
    digestAlgorithm         AlgorithmIdentifier,
    objectDigest            BIT STRING
}
```

Como puede comprobarse, esta opción permite generar certificados de atributos ligados a una clave pública. También permite, mediante el campo *otherObjectTypes*, ge-

nerar certificados que contengan, por ejemplo, los privilegios asociados con un objeto ejecutable tal que una *clase* de Java. En el perfil definido en la RFC 3281 se matiza en qué condiciones puede utilizarse el *hash* de una clave pública, aunque no se permite (por ahora) la inclusión de estos otros tipos de objetos. No obstante, estos otros objetos ofrecen un amplio abanico de posibilidades para recoger, a medida que los planteamientos vayan evolucionando, las necesidades que se detecten en los esquemas de autorización.

Como se deduce de los párrafos anteriores, en comparación con el campo *subject*, que sirve para especificar la identidad del titular de un certificado de clave pública, el campo **holder**, que cumple la misma función en un certificado de atributos, está definido de forma mucho más variada y versátil. Es en la especificación de este campo **holder** (además, claro está, del campo reservado a los atributos) donde mayor novedad y diferencia existe entre las definiciones de ambos tipos de certificados. En efecto, los restantes campos del tipo **AttributeCertificateInfo** son fáciles de entender para aquellas personas que ya conocen la definición formal del certificado de la clave pública.

Así, es bastante fácil de interpretar el campo **issuer**, que está definido por el tipo *AttCertIssuer* y sirve para indicar el nombre de la *Autoridad de Atributos* (AA) que ha generado el certificado. Está definido mediante un CHOICE para hacer compatible la v1 del certificado (más elemental) con la v2 (que ofrece algunas posibilidades más). Su especificación en el perfil definido en la RFC 3281 es:

```
AttCertIssuer ::= CHOICE {
    v1Form      GeneralNames,      -- NO DEBE ser usado en este perfil
    v2Form      [0] V2Form         -- solo presente en v2
}

V2Form ::= SEQUENCE {
    issuerName      GeneralNames OPTIONAL,
    baseCertificateID [0] IssuerSerial OPTIONAL,
    objectDigestInfo [1] ObjectDigestInfo OPTIONAL
    -- en este perfil DEBE estar presente issuerName
    -- en este perfil NO DEBEN estar presentes
    -- ni baseCertificateID ni objectDigestInfo
}
```

Es decir, mientras la v1 sólo incluye el muy versátil tipo *GeneralNames* (que ya describimos resumidamente en el apartado «*Los nombres de las entidades en la X.509 v3*» dentro del epígrafe 6.4), la v2 define este campo mediante una secuencia de tres elementos, siendo el primero de ellos coincidente con el de la v1 y optativos los dos últimos, que son del tipo *IssuerSerial* y *ObjectDigestInfo* respectivamente, tipos ambos presentes en la especificación del **holder** comentada más arriba. Además, el perfil definido en la RFC 3281 no permite, por ahora, el uso de estos parámetros complementarios e impone la restricción de que dentro de las múltiples opciones que ofrece el *GeneralNames* (véase su definición formal en el apartado «*El campo de extensiones*» del epígrafe 6.6) no esté vacío el reservado para *directoryName*, es decir, que todas las AAs generadoras de certificados de atributos deben tener un nombre distintivo X.500 para dejar más claramente asegurada su ubicación y facilitar la interacción con las infraestructuras que soportan los certificados de la clave pública.

Los campos **signature**, **serialNumber** y **attrCertValidityPeriod** no ofrecen ninguna novedad significativa en relación con sus equivalentes en el certificado de la clave pública. El primero de ellos identifica el algoritmo utilizado por la AA para firmar el certificado, el segundo es un entero que junto con la identidad de la AA

emisora sirve de identificador único del certificado, y el tercero marca el periodo de validez mediante el tipo *GeneralizedTime* ya comentado en los capítulos precedentes.

El campo **attributes** está especificado como una secuencia de elementos de tipo *Attribute* ya definido en el contexto del certificado X.500. Su especificación es como sigue:

```
Attribute ::= SEQUENCE {
    type           AttributeType,
    values         SET OF AttributeValue
    -- se requiere la presencia de al menos un valor
}
```

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY DEFINED BY AttributeType

Es decir, cada uno de los atributos presentes en la secuencia está definido a través de un identificador de objeto en el que se especifican además los valores que puede adquirir. En el siguiente apartado se comentan los atributos más significativos y la utilidad de su uso.

El campo optativo **issuerUniqueID** tiene la misma significación y el mismo poco interés que su homónimo del certificado X.500. En el perfil definido en la RFC 3281 se desaconseja su uso.

Por último, el campo **extensions** es exactamente el mismo que está definido en el certificado X.509. También dedicaremos otro nuevo apartado a comentar las extensiones que tienen mayor interés de cara a la provisión de los servicios de control de acceso.

TIPOS DE ATRIBUTOS

El campo de atributos aporta información de control de acceso asociada con el titular del certificado, que puede ser, en principio, cualquier entidad comunicante. Para la provisión del servicio de Control de Acceso, los certificados de atributos sirven como mecanismos de seguridad que permiten, a través del campo de atributos, asignar privilegios a las entidades participantes en un entorno de comunicación.

Como acabamos de ver en su definición, este campo del certificado puede contener una serie de atributos (se exige que, al menos, haya uno) cada uno de los cuales puede contener uno o varios valores, según esté definido. La especificación remite a un OBJECT IDENTIFIER para cada tipo de atributo que se defina, de forma que en esa definición se especificará la sintaxis concreta de ese tipo de atributo y el conjunto de valores que puede adquirir.

Como ya comentamos en el epígrafe 7.3 cuando se introdujo el concepto de atributo en la descripción de las entradas o inserciones que se definen en el árbol de información del Directorio (DIT), existirá un conjunto normalizado de atributos que serán los que entiendan los protocolos y sistemas diseñados conforme a la norma que los incluye. No obstante, en cada entorno de comunicación se podrán definir atributos específicos que sólo entenderán los sistemas de ese entorno. A continuación describiremos brevemente los atributos seleccionados en el perfil definido en la RFC 3281 que estamos utilizando como referencia (omitiendo los identificadores de objeto y otros detalles). Son los siguientes:

a) *Información para el Servicio de Autenticación (SvceAuthInfo)*

Este atributo identifica al titular del certificado ante un servicio o servidor explícitamente especificado mediante un nombre, pudiendo ir acompañado de un campo optativo que incluya información de autenticación específica para ese servidor. Lo más sencillo es que este atributo contenga una pareja constituida por un nombre y una contraseña que serán interpretados (una vez verificado el certificado) por la aplicación de control de acceso que resida en el recurso o servidor al que se pretende acceder.

La sintaxis definida para este atributo es:

```
SvceAuthInfo ::= SEQUENCE {
    service           GeneralName,
    ident            GeneralName,
    authInfo         OCTET STRING OPTIONAL
}
```

Como puede comprobarse, todos los tipos incluidos en la especificación son muy sencillos y fáciles de codificar. En este atributo se permite la aparición de varios de estos valores. Cuando se incluye el campo optativo que aporta información sensible, será conveniente cifrar los datos.

b) *Identidad de Acceso (accessIdentity)*

Este atributo identifica al titular del certificado ante un servidor o servicio. La sintaxis que usa es la *SvceAuthInfo* definida para el atributo antes descrito con la salvedad de que **no puede** usarse el campo optativo. Es decir, desde un punto de vista formal ambos atributos son muy parecidos, aunque su uso es diferente.

Así, mientras que el anterior sirve para resolver la autenticación del usuario ante el servidor (el dictamen será sí/no), este otro atributo está pensado para proporcionar información sobre la identidad del titular (que vendrá garantizada por la firma que la AA estampa en el certificado) para que pueda ser utilizada (una vez realizada la verificación) para determinar qué acciones u operaciones puede llevar a cabo ese usuario dentro del sistema que incluye a la entidad verificadora.

c) *Grupo (group)*

Mediante la información que aporta este atributo, la entidad titular del certificado puede identificarse como miembro de uno o varios grupos concretos (los que se especifican en el atributo).

Para la definición formal de este atributo se usa el tipo *IetfAttrSyntax*, definido de forma genérica para que pueda ser usado por varios de los atributos que se definan. La definición de esta sintaxis es la siguiente:

```
IetfAttrSyntax ::= SEQUENCE {
    policyAuthority  [0] GeneralNames OPTIONAL,
    values           SEQUENCE OF CHOICE {
                        octets      OCTET STRING,
                        oid         OBJECT IDENTIFIER,
                        string     UTF8String
    }
}
```

Esta definición genérica es de interés cuando se requiere separar claramente la *Autoridad de Política* de la *Autoridad de Atributo* que genera el certificado. En el primero

de los dos campos se identifica esa autoridad mediante un *GeneralName*. El atributo representado por esta sintaxis puede ser multievaluado, es decir, presentar varios valores para un mismo atributo, lo que se consigue mediante la SEQUENCE presente en el campo *values*. Para representar los valores se puede elegir entre tres tipos bastante sencillos, pero se impone la restricción de que todos los valores de la secuencia utilicen la misma elección. Es decir, en el caso del atributo *group* que ahora nos ocupa, la sintaxis permite que se identifiquen varios grupos distintos a los que pertenezca el titular del certificado, pero si para identificar uno de ellos se utiliza el tipo OCTET STRING, todos los demás grupos tienen que ser identificados usando ese mismo tipo.

d) *Identidad para tarificación (charginIdentity)*

Si está presente, este atributo sirve para identificar al titular del certificado a efectos de tarificación, es decir, identificar quién va a pagar los costes derivados del servicio. Por regla general, cuando el titular pertenece a un organismo público o privado será ese organismo el que se haga cargo de los gastos. Este atributo permite que eso se haga constar en el certificado firmado por la AA de acuerdo con la política de control de acceso definida en ese dominio de seguridad.

La definición formal de este atributo se apoya también en la sintaxis *IetfAttrSyntax* que acabamos de comentar. Es decir, se identificará quién es la *Autoridad de Política* y se permitirá que a la hora de decir quién paga se pueda señalar a más de una entidad.

e) *Rol (role)*

Como su nombre indica, este atributo permite informar acerca del rol asignado a la entidad titular del certificado de atributos.

La sintaxis utilizada para generar el valor en este atributo es:

```
RoleSyntax ::= SEQUENCE {
    roleAuthority           [0] GeneralNames OPTIONAL,
    roleName                [1] GeneralName
}
```

En cada pareja definida mediante esta sintaxis el primer elemento es optativo, es decir, puede omitirse la identificación de la Autoridad responsable de la definición del rol (que puede hacerlo emitiendo un certificado de especificación de rol). No obstante un mismo titular puede tener un mismo nombre de rol (por ejemplo «administrador») definido por más de una autoridad y en cada caso producirá efectos diferentes en cuanto a control de acceso se refiere. Este atributo permite varios de estos valores.

f) *Acreditación (clearance)*

Este atributo sirve para asignar información de control de acceso al titular del certificado en forma de niveles de seguridad para que pueda utilizarlo en los distintos esquemas y políticas de control de acceso en que pueda verse involucrado. La sintaxis en base a la cual se define este atributo es la siguiente:

```
Clearance ::= SEQUENCE {
    policyId                [0] OBJECT IDENTIFIER,
    classList              [1] DEFAULT {unclassified},
    securityCategories     [2] SET OF SecurityCategory OPTIONAL
}
```


Esta definición es mucho más abierta que lo que en principio pudiera parecer. El primer campo identifica la política de control de acceso bajo la que están definidas las acreditaciones que se recogen en este atributo. En esa política es donde deberá definirse el significado real de los otros dos campos de la especificación. El tipo en el que se apoya el segundo campo se define en la norma como:

```
ClassList ::= BIT STRING {
    unmarked      (0),
    unclassified  (1),
    restricted    (2),
    confidential  (3),
    secret        (4),
    topSecret     (5)
}
```

Si analizamos el significado ASN.1 de esta definición, vemos que lo único que normaliza es una ristra de bits con seis posiciones, de 0 a 5, a las que le asigna unos nombres de niveles de seguridad bastante clásicos. Pero ello no condiciona en absoluto a quienes sean responsables de definir la política de control de acceso en el dominio de seguridad, los cuales podrán definir los niveles presentes y su significado. El único condicionante que marca la definición del atributo es que el tipo del campo *classList* sea un BIT STRING.

Algo similar (sobre la libertad de los que definan la política de control de acceso) podemos decir de la definición del tercer campo, optativo, que sirve para incluir información adicional. En la norma se define de una forma tan abierta como un SET OF de elementos del siguiente tipo:

```
SecurityCategory ::= SEQUENCE {
    type      [0] IMPLICIT OBJECT IDENTIFIER,
    value     [1] ANY DEFINED BY type
}
```

Es decir, deja las manos libres para definir cualquier sintaxis por compleja y evolucionada que sea (obsérvese que desde un punto de vista formal esta definición es muy similar a la del tipo *Attribute* que vimos en el anterior apartado).

Hemos insistido tanto en dar detalles acerca de la definición de este atributo porque, observando a simple vista la especificación presentada en la norma, podría dar la impresión de que se trata de una definición cerrada y que todas las políticas que se pudiesen aplicar aquí están ya decididas. Por una parte, según venimos insistiendo machaconamente, es importantísimo que se traten de formalizar las políticas para que puedan ser analizadas por agentes telemáticos automáticos, pero también es muy importante dejar abierta la especificación para que pueda ser mejorada en un futuro, dejando libertad a los responsables de definir la política para que puedan adaptarla a sus necesidades reales.

EXTENSIONES EN LOS CERTIFICADOS DE ATRIBUTOS

Como ya se ha dicho, desde un punto de vista formal el campo de extensiones es en todo coincidente con el definido en el certificado X.509 de la clave pública, lo que equivale a decir que también es coincidente con las extensiones que pueden añadirse a las CRLs.

En la Recomendación X.509 y en otras normas se definen bastantes de estas extensiones. No obstante, por razones de oportunidad, al igual que hemos hecho con el campo de atributos, nos apoyaremos en las especificaciones de la RFC 3281, dado el punto de vista práctico y de adaptación a necesidades reales con el que ha sido desarrollada. En esta RFC se define un conjunto concreto de posibles extensiones. No obstante, los gestores de un determinado dominio de seguridad pueden decidir incluir cuantas extensiones crean convenientes a los certificados de atributos que se generen en el entorno que cae bajo su ámbito de competencia. Esas «nuevas» extensiones pueden pertenecer a los conjuntos definidos para los certificados X.509 o para las CRLs, o bien pueden ser definidas por ellos mismos con el propósito de que satisfagan algunas necesidades específicas que allí se hubiesen detectado. Otra cosa es que la RFC 3281 imponga cuáles han de ser las condiciones que han de cumplir esas nuevas extensiones si lo que se pretende es que los certificados definidos en ese entorno sean compatibles con el perfil ahí definido. La única prevención que impone esta norma para garantizar la compatibilidad es que las nuevas extensiones no estén calificadas como *críticas*. Esto es así debido a que si un servidor recibe un certificado de atributos con una extensión que no entiende y esa extensión está calificada como no crítica, el servidor simplemente la ignora y continúa analizando las restantes extensiones, pero si se encuentra con una extensión crítica que no es capaz de interpretar tiene que rechazar el certificado en cuestión.

Por lo general, el campo de extensiones aporta información acerca del certificado de atributos en lugar de hacerlo sobre el titular del certificado. A continuación se describen de forma resumida y simplificada las extensiones definidas en la RFC 3281. Son las siguientes:

a) *Identidad para Auditoría (auditIdentity)*

Consiste en definir una *identidad para auditoría* que pueda ser usada (por motivos legales de protección de datos personales) en algunos procesos en los que no se permita registrar la verdadera identidad de los usuarios. Puede tratarse de auditorías propiamente dichas o de registros de *logging*. En estos casos es conveniente la inclusión de una identidad que no se pueda relacionar con la identidad del titular (campo *holder*) sin la cooperación de la AA emisora del certificado. De esta manera, apoyándose en esta especie de alias, un determinado servicio puede trazar el comportamiento de un usuario sin por ello ser capaz de identificar al titular.

No obstante, si a posteriori se detecta un comportamiento indebido debe ser posible, con la cooperación de la AA emisora, localizar la verdadera identidad de la entidad propietaria del certificado bajo el que realizó determinada acción.

La sintaxis de esta extensión consiste, simplemente, en un OCTET STRING cuyo número de octetos se recomienda sea mayor que cero y menor de 20. Caso de estar incluida en un certificado, esta extensión debe ser *crítica*.

b) *Información sobre el Objetivo (targetInformation)*

Sirve para especificar la identidad de los servidores o recursos sobre los que se debe usar el certificado de atributos que contenga esta extensión. Un servidor (o la entidad verificadora en la que se apoye) que compruebe que su nombre no está entre los designados en esta extensión debería rechazar el uso de ese certificado. Obviamente, si esta extensión no está presente el certificado puede usarse sobre cualquier objetivo.

La sintaxis usada para representar esta información es la siguiente:

Targets ::= SEQUENCE OF Target

```

Target ::= CHOICE {
    targetName      [0] GeneralName,
    targetGroup     [1] GeneralName,
    targetCert      [2] TargetCert
}

```

Como puede observarse, se trata de un CHOICE, es decir, aparecerá una y sólo una de las distintas opciones. La primera de ellas sirve para especificar una lista (SEQUENCE) de nombres de servidores u objetivos para los que es de aplicación el certificado en cuestión. Si la que se usa es la segunda alternativa, un objetivo aceptará el certificado si él pertenece a uno de los grupos de objetivos (*targetGroup*) especificados. La tercera de las opciones se apoya en una secuencia de certificados cuya explicación y especificación formal obviarnos por considerarla menos significativa. En la norma no se especifica cómo se determinan los grupos ni los mecanismos que deben seguir los objetivos para saber si están entre los «destinatarios» del certificado de atributos, con lo cual la decisión sobre usar o no el certificado se deja bajo la responsabilidad de cada servidor u objetivo.

c) Delegación (Proxying)

En la especificación de la RFC 3281, el término *proxying* se usa para referirse a la situación en la que un servidor actúa en representación (por delegación) de un usuario. (Este tipo de delegación no coincide con el hecho de *delegación de privilegios* que una AA hace al emitir un certificado.)

Esta extensión sirve para que la AA que emite el certificado pueda especificar la identidad de los servidores que pueden estar involucrados en este tipo de delegación. La sintaxis usada es muy sencilla, ya que consiste en una secuencia del tipo que acabamos de ver para *targetInformation*:

ProxyInfo ::= SEQUENCE OF Targets

Para dar por válido que un servidor actúe en representación del titular de un certificado tanto la identidad de quien «envía» la delegación como de quien la recibe debe estar incluida dentro de las especificadas en el certificado y en la extensión, conforme a unas condiciones precisas establecidas en la norma. Puede pensarse también en una situación más compleja en la que un certificado es «delegado» más de una vez. Tanto en las situaciones más simples como en las más complejas, en los entornos donde estas delegaciones se consideren convenientes deberán establecerse condiciones rigurosas que eviten disfunciones e inconsistencias.

d) Identificador de la clave de la entidad emisora (authorityKeyIdentifier)

Esta extensión es exactamente la misma que está definida para el Certificado X.509 de la clave pública que resumimos en el apartado «Extensiones del certificado X.509 v3» dentro del epígrafe 6.4. La única diferencia es que en este caso *entidad emisora* es una AA en lugar de una CA. Esta extensión sirve para que la entidad que haga uso del certificado de atributos pueda identificar con garantías la clave pública de la AA con cuya clave privada fue firmado el certificado.

Debido a que el campo *issuer* en los certificados de atributos tiene más opciones que el correspondiente de los certificados de la clave pública, el certificado de atribu-

tos puede contener información que haga ociosa esta extensión, que si es recogida debe llevar el calificativo de no crítica.

e) *Acceso a la información de la autoridad emisora (authorityInfoAccess)*

Es también coincidente con la extensión de igual nombre que recogimos en el apartado de extensiones del epígrafe 6.4 (allí la ligábamos a la CA) y está definida formalmente en [HPFS02].

Sirve para ayudar a la entidad verificadora del certificado de atributos en la comprobación del estado de revocación de éste, para lo que se puede contar con la ayuda de un servicio OCSP. Realmente, si, como recomendamos cuando estudiamos las infraestructuras PMI en el Capítulo 7, no van a existir múltiples AAs organizadas jerárquicamente, ni cadenas de certificados de atributos, ni caminos de confianza, la localización del certificado no tiene por qué ser tan compleja como en las infraestructuras PKI. Si además los periodos de validez son reducidos será poco frecuente la existencia de certificados revocados, por lo que no merece mucho la pena tomar demasiadas medidas para comprobar el estado en que se encuentra el certificado. Esta extensión no es crítica.

f) *Puntos de distribución de CRLs (cRLDistributionPoints)*

En realidad esta extensión para lo que sirve es para localizar los puntos de distribución de ARLs (lista de certificados de atributos revocados), pero se ha aprovechado la definición existente dentro de las infraestructuras de certificación acerca de las CRLs. Por tanto, la especificación formal de esta extensión es exactamente la misma que presentamos en el apartado «Puntos de distribución de CRLs y CRLs indirectas» dentro del epígrafe 7.7.

En el perfil definido en la RFC 3281 se limitan mucho las opciones que ofrece la especificación de la extensión. Así, impone que solamente podrá señalarse un punto de distribución y que su nombre ha de ser un nombre distintivo X.500 o una URL (HTTP o LDAP). De estar presente en el certificado esta extensión no es crítica.

g) *No disponible información de revocación (noRevAvail)*

Sirve para que la AA emisora indique que no hay disponible información acerca de la revocación de ese certificado (*no revocation available*). Es decir, no estará incluido en ninguna ARL. Su sintaxis consiste simplemente en el valor NULL (se representa en DER mediante 0500₁₆). Si está presente, no es crítica.

Desde un punto de vista formal, la extensión lo único que dice es que no existe información de revocación para el certificado en el que está incluida, pero no que estén excluidas las revocaciones. No obstante, puede dársele un uso más específico.

En efecto, según hemos venido comentando, cabe esperar que los privilegios asignados mediante los certificados de atributos cambien con cierta frecuencia, por lo que los periodos de validez estipulados serían bastante reducidos. Ello conlleva que, bajo esas circunstancias, en la política de seguridad que determina el comportamiento de las AAs generadoras de certificados se opte por un *esquema de no revocación*, es decir, que en ese entorno de comunicación no está permitida la revocación de certificados. En este caso, debe utilizarse la presencia de esta extensión *noRevAvail* para indicar el uso de este esquema.

En sentido contrario, la ausencia de esta extensión sería indicación implícita de que sí es posible que el certificado esté revocado. Lo mejor sería que no fuese necesario revocar certificados. Pero si esto es inevitable, una buena política de control de

acceso podría consistir en imponer que todos los certificados de atributos que se generasen en ese entorno llevaran la extensión *cRLDistributionPoints* con la restricción que impone la RFC 3281, es decir, que se determinase un **único sitio** en el que comprobar el estado de revocación de cada certificado. De esta manera el siempre espinoso problema de las revocaciones se reduciría bastante y sería fácil de gestionar.

9.5. GENERALIDADES SOBRE EL SERVICIO DE NO REPUDIO

El *Servicio de No Repudio* (*Non-repudiation* en inglés) sirve para proteger a los usuarios de las redes telemáticas contra el hecho de que alguien niegue falsamente haber participado en algún evento u acción que haya tenido lugar.

En la versión en español de las recomendaciones de ITU-T, en lugar de *no repudio* se utiliza el término *no rechazo*, que viene a significar lo mismo pero no tiene las connotaciones negativas que acompañan al primero. Aquí hemos optado por mantener la denominación «clásica» de *No Repudio* por entender que es la más extendida y la que aparece con más frecuencia en los textos escritos en español.

En este y en el siguiente epígrafe se destacan los elementos principales que intervienen en su provisión y se plantea una clasificación de los distintos tipos de servicios de no repudio que pueden presentarse, tratando de destacar los aspectos más significativos de cada uno de ellos.

NECESIDAD DE LOS SERVICIOS DE NO REPUDIO

Aunque el espacio que aquí dedicamos a este servicio es bastante más reducido que el dedicado a otros (como puede ser el de autenticación o el de control de acceso), ello no significa que el servicio de no repudio tenga menos importancia o repercusión que aquéllos. Como veremos en los párrafos que siguen, este servicio es absolutamente imprescindible para conseguir que las comunicaciones que se desarrollen en el ámbito de la Sociedad de la Información puedan tener las mismas garantías (pero más robustas) que las que se llevan a cabo en las comunicaciones convencionales mediante papel.

Podemos destacar tres ámbitos de comunicación en los que la necesidad de este tipo de servicio se hace más patente:

1. En las **relaciones comerciales** y contractuales a través de redes telemáticas resulta imprescindible que se puedan establecer garantías sobre la autoría de un documento, o que se sepa con exactitud quién y cuándo ha enviado una oferta o un pedido a otra entidad comercial, o que se pueda disponer de algún elemento justificativo de que la otra parte ha recibido, por ejemplo, un requerimiento sobre un pago insatisfecho.

Estos son sólo unos cuantos ejemplos de las muchas situaciones que pueden darse en las relaciones comerciales (tanto entre empresas como en su relación con clientes y consumidores) en los que es necesario disponer de pruebas que eviten que quien ha tomado determinadas decisiones pueda negar falsamente esa realidad. Parece claro que si los servicios telemáticos que ofrecen esas garantías no se desarrollan e implantan, las relaciones comerciales sobre redes telemáticas seguirán teniendo las alas cortadas.

2. En las **relaciones de las administraciones públicas** con los ciudadanos, las prácticas consagradas por la costumbre y respaldadas por el ordenamiento jurídico dan una gran importancia, en multitud de actos administrativos, a la obtención de justificantes que avalen el haberlos llevado a cabo. Quizás los dos casos en los que esta situación se da de manera más notoria son:
 - a) Cuando alguien deposita un documento en un Registro de Entrada (que tiene derecho a exigir que quien atienda el Registro le devuelva una fotocopia sellada del documento entregado).
 - b) En la notificación de alguna resolución o trámite desde la Administración hacia los ciudadanos (que conlleva la obtención, por parte del organismo remitente, de un acuse de recibo firmado por el destinatario).

Situaciones similares se producen en otras muchas organizaciones públicas o privadas, como puede ser el caso de una Universidad, en las que la expedición de este tipo de justificantes está muy extendida y tiene una fuerte aceptación social.

Como más adelante se comenta, en las administraciones públicas existe un interés creciente por adaptar las normativas vigentes para que muchos trámites administrativos puedan llevarse a cabo a través de Internet, con la consiguiente comodidad y ahorro de tiempo para el ciudadano, pero se detecta que para que este tipo de prácticas se consoliden es imprescindible la implantación de servicios de No Repudio que proporcionen pruebas robustas.

3. En las **relaciones personales** a través de las redes telemáticas también pueden ser necesarias algunas de las garantías que estamos comentando, pero en menor medida y en un menor número de casos. En la mayoría de éstos es previsible que solamente sea necesario tener la seguridad de que quien escriba una carta no pueda luego repudiar su autoría, cosa que, según vimos en el capítulo anterior, está garantizada por la firma electrónica.

A pesar de lo perentorio que resulta la aplicación de los servicios de No Repudio, se detecta un cierto retraso en su implantación en relación con otros servicios de seguridad (o bien se dan por implantados pero bajo esquemas muy poco robustos). Una de las causas de este retraso puede entenderse, pero no justificarse, por el hecho de que para su adecuada provisión no sólo es necesario el concurso de TTPs genéricas como las presentes en infraestructuras PKIs o PMIs, sino que también se necesita de la implantación de TTPs especializadas en la provisión de este servicio.

FASES DEL SERVICIO DE NO REPUDIO

Podemos definir este servicio diciendo que consiste en la generación, verificación y almacenamiento de *evidencias* que puedan servir de *pruebas* (demostrables ante terceros) para garantizar que un determinado evento u acción ha sucedido. Una *evidencia* es una pieza de información segura en sí misma (o en conjunción con otra información) que puede utilizarse para resolver una disputa.

La *evidencia* deberá ser generada utilizando mecanismos criptográficos robustos para que pueda ser considerada como una prueba no falsificable. Dependiendo del tipo de servicio de no repudio, la evidencia deberá contener una información concreta que permita garantizar la ocurrencia del evento de que se trate. En cualquier caso, la evidencia deberá contener información que identifique de forma expresa la Política de No Repudio bajo cuyas reglas ha sido generada.

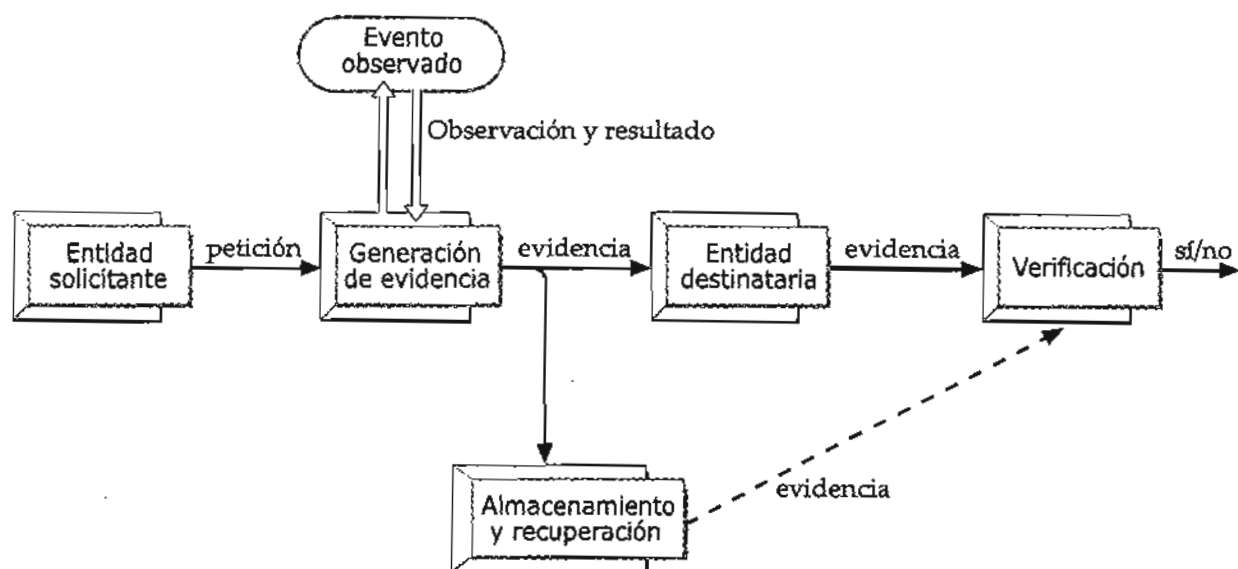


Figura 9.7. Diagrama de generación y verificación de evidencias.

De forma genérica, podemos considerar (véase Figura 9.7) las siguientes fases en la provisión de un servicio de no repudio:

- *Generación* de la evidencia bajo petición de la *entidad solicitante*.
- *Transferencia* de la evidencia.
 - *Envío* de la evidencia a la *entidad destinataria*.
 - *Almacenamiento y Recuperación* de la evidencia (opcional).
- *Verificación* de la evidencia.

A estas tres fases habría que añadirle otra más que cabe esperar que se produzca solamente de forma excepcional:

- *Resolución de disputas* a petición de parte.

Las dos primeras fases se llevan a cabo al mismo tiempo que está sucediendo el evento que se trata de constatar, mientras que la tercera puede llevarse a cabo inmediatamente después de que la entidad destinataria haya recibido la evidencia o cierto tiempo después. La *resolución de disputas*, caso de producirse, se hará transcurrido bastante tiempo desde que se produjo el evento.

Dado que la evidencia es una pieza de información segura en sí misma (o mediante su constatación con certificados u otras informaciones de seguridad generadas dentro de la infraestructura de clave pública en la que está siendo provisto el servicio), la entidad que la recibe puede almacenarla para su hipotética utilización posterior sin tener que utilizar para ello medidas de seguridad especiales. Opcionalmente, puede ser también almacenada al mismo tiempo que se envía a la entidad destinataria para poder ser usada como prueba tiempo después.

Precisamente, debido a que la evidencia es una pieza de información segura y no falsificable, podemos afirmar que es de esperar que la cuarta fase, la de *resolución de disputas*, solamente se produzca de forma excepcional.

En efecto, este tipo de conflicto sólo se producirá cuando alguien (la entidad demandante) acuse a otra parte de haber participado (o negar que lo haya hecho) en un determinado evento. La disputa será resuelta por las entidades especificadas en la Política de No Repudio para actuar como *árbitros* conforme a las reglas marcadas en esa misma política. La robustez de las pruebas debe dar lugar a que las resoluciones de los árbitros sean verificables y previsibles, por lo que quienes conozcan la existen-

cia de estas evidencias raramente presentarán demandas a las entidades participantes en el entorno de seguridad.

9.6. DIFERENTES TIPOS DE NO REPUDIO

En la norma «matriz» ISO/IEC 7498-2 *Security Architecture* se distinguen solamente dos tipos de servicio de no repudio (*de origen y de entrega*), clasificación que se mantiene, con ligeras variantes, en la ISO/IEC 10181-4 (ITU-T X.813)⁵. Nosotros, en el Capítulo 1, con la vista puesta en los servicios necesarios para el intercambio de mensajes, aumentábamos a tres el número de tipos distintos, incluyendo el *No Repudio de Envío*. Llegados a este punto, tratando de ampliar un poco más el ámbito de los entornos de comunicación en los que conviene disponer de estos servicios, vamos a distinguir entre siete tipos distintos de servicios de No Repudio. Son los siguientes:

1. *No Repudio de Origen (Non-Repudiation of Origin).*
2. *No Repudio de Envío (Non-Repudiation of Sending).*
3. *No Repudio de Depósito (Non-Repudiation of Submission).*
4. *No Repudio de Transporte (Non-Repudiation of Transport).*
5. *No Repudio de Entrega (Non-Repudiation of Delivery).*
6. *No Repudio de Creación (Non-Repudiation of Creation).*
7. *No Repudio de Conocimiento (Non-Repudiation of Knowledge).*

NO REPUDIO DE ORIGEN

En el caso de transferencia de información o de mensajes, este servicio sirve para que la entidad receptora tenga una prueba, demostrable ante terceros, del origen de los datos que ha recibido. De esta forma, la entidad emisora de un mensaje no podrá negar su participación en ese envío.

La forma más sencilla de proporcionar este servicio es que la entidad emisora proceda a firmar el mensaje antes de enviarlo. En principio puede tratarse simplemente de una firma digital normalizada realizada en base a su clave privada. Más robusto sería que la firma fuese una firma electrónica (del tipo de las estudiadas en el Capítulo 8) con el añadido de un sello de tiempo proporcionado por una TSA (*Time Stamping Authority*). Para posibilitar la generación de la firma y su posterior verificación será conveniente, tal como se recoge en la Figura 9.8, el concurso de distintas TTPs

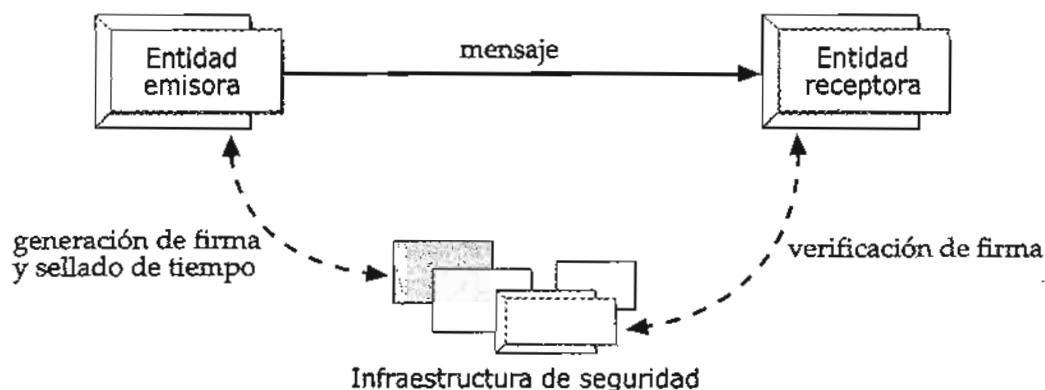


Figura 9.8. Servicio de No Repudio de Origen.

pertenecientes a la Infraestructura de Seguridad en que se apoyan las entidades que participan en la comunicación.

La entidad emisora del mensaje deberá firmar los datos con su clave privada, estén o no previamente firmados, e independientemente de quién sea el autor o autora de ellos o de quién los haya producido.

Por regla general, en una comunicación personal puede ser suficiente que se garantice la confidencialidad del envío y que los datos estén firmados de forma que se asegure la autenticación de su autor, pero en relaciones comerciales y, sobre todo, en las relaciones de los ciudadanos con la Administración, puede ser exigible conocer la identidad telemática de la entidad desde donde se envía el mensaje. Por ejemplo, en España, el Ministerio de Economía ha elaborado una normativa en la que, para determinado tipo de trámites legales, se exige que la dirección de correo que utilicen los ciudadanos no pueda ser cualquiera, sino que debe ser una dirección validada por el Ministerio, que mantendrá para ello una lista actualizada de proveedores que ofrezcan servicio de correo electrónico conforme a unas especificaciones previamente establecidas⁶.

Volviendo al esquema representado en la Figura 9.8, la entidad receptora podrá exigir, para aceptar el mensaje, que éste venga acompañado de la correspondiente firma (si no cumple estas condiciones puede decidir dar por no recibido ese mensaje). Con ayuda de la Infraestructura de Seguridad podrá validar la firma y, si lo tuviese, el sello de tiempo, obteniendo así una prueba que puede presentar ante terceros en caso de disputa o conflicto. Las TTPs que colaboran en esta tarea lo hacen *fuera de línea* (*off-line*), ya que si bien son fundamentales para la generación de la firma y el sellado de tiempo y la verificación, no participan de forma directa en la transferencia del mensaje.

NO REPUDIO DE DEPÓSITO Y NO REPUDIO DE ENVÍO

Ambos tipos de servicios de no repudio generan evidencias sobre un mismo evento: que la entidad emisora ha enviado el mensaje a una dirección concreta y en un momento concreto. La diferencia entre ellos consiste en la intención y en el destinatario de la evidencia generada en cada caso.

En la Figura 9.9 se representa la configuración requerida para la provisión de estos dos servicios. La novedad más significativa con respecto al servicio descrito en el anterior apartado consiste en la presencia de una TTP de No Repudio especializada trabajando *en línea* (*in-line*), es decir, de tal forma que los mensajes pasan a través de ella. Esta TTP consta, a su vez, de dos entidades: una de ellas para la recepción de los mensajes que le envíen las entidades emisoras y otra entidad de reenvío para hacer llegar el mensaje a la entidad destinataria. Además de esta TTP específica, en la figura aparece también el conjunto de TTPs que constituyen la Infraestructura de Seguridad.

Analizaremos en primer lugar el *No Repudio de Depósito*. Podemos definirlo diciendo que, merced a este servicio, la **entidad emisora** obtiene una prueba, demostrable ante terceros, de la fecha y hora en que el mensaje **fue enviado**.

Con un esquema como el planteado en el apartado anterior, con todas las TTPs operando *fuera de línea*, la entidad emisora no tiene posibilidad de obtener esa prueba. En muchos casos de la vida real la obtención de una evidencia de este tipo resulta muy necesaria. En efecto, supongamos que estamos en un escenario en el que, por ejemplo, con motivo de un acto administrativo o de una licitación comercial o de una participación en un congreso, un conjunto de entidades emisoras han de enviar un mensaje de correo a un destinatario prefijado *antes* de una fecha límite, de tal manera que si el mensaje se envía después de esa fecha, no procede tenerlo en consideración.

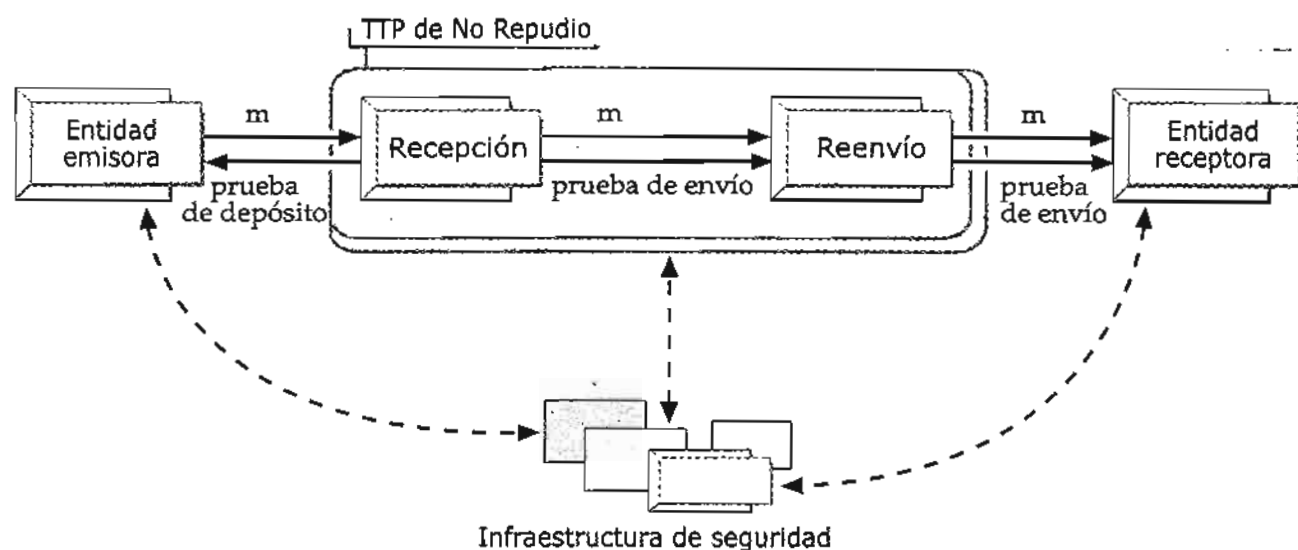


Figura 9.9. Servicios de No Repudio de Depósito y No Repudio de Envío.

En el Servicio Postal convencional este tipo de situaciones son fácilmente resolubles porque cuando se envía una carta certificada a través de una estafeta se obtiene un recibo o justificante que indica la fecha en que fue depositada en dicha estafeta. Si se trata de actos administrativos, se puede solicitar incluso que una fotocopia del documento enviado sea sellada con una marca de tiempo, de tal forma que, en algunos casos, puede servir como prueba de remisión. Lo que está ocurriendo realmente es que el Servicio Postal se está comportando como una entidad intermediaria de confianza para las dos partes involucradas en la comunicación.

Por ese mismo motivo, para la provisión de este servicio de seguridad en las redes telemáticas se necesita de una TTP de No Repudio especializada (Figura 9.9) que actúe como intermediaria en la transferencia del mensaje. La TTP en la que se *deposita* el mensaje genera una *evidencia* firmada con su clave privada y complementada con un sello de tiempo fiable que devolverá a la entidad emisora. Esta evidencia le servirá de prueba robusta para demostrar, en caso de conflicto o discrepancia, que el mensaje sí se envió y que se envió en la fecha y hora marcadas.

Situémonos ahora en el lado de la entidad receptora. Esta entidad obtiene una garantía acerca de las circunstancias en que se ha producido el envío del mensaje mediante el concurso del servicio de *No Repudio de Envío*. Podemos definirlo diciendo que, merced a este servicio, **la entidad receptora** obtiene una prueba, demostrable ante terceros, de la fecha y hora en que el mensaje **fue enviado**. En alguna literatura se asocia, erróneamente, este servicio al de No Repudio de Origen estudiado en el apartado anterior. Examinemos la diferencia:

En efecto, pensemos de nuevo en el escenario de comunicación antes descrito a modo de ejemplo. Si la entidad emisora envía su mensaje justo antes de finalizar el plazo, y si solamente dispusiésemos de un esquema como el planteado para la provisión del servicio de *No Repudio de Origen*, la entidad receptora sí tendría garantías de que el mensaje fue enviado y tendría pruebas acerca de la identidad telemática de la entidad emisora y de la fecha y hora **en la que firmó** el mensaje a enviar, pero no puede tener garantías de que el mensaje se enviase a esa hora y no cierto tiempo después. Es decir, no conoce detalles acerca del **evento de envío**. Si la entidad receptora quiere ser ecuánime y tratar por un igual a todos los hipotéticos enviantes, para poder aceptar el mensaje debe tener garantías de que se envió dentro del plazo marcado por las reglas estipuladas para esa comunicación.

La manera de que la entidad receptora adquiriera esa garantía es mediante la colaboración de la TTP especializada (Figura 9.9), que le enviará una *evidencia* firmada y con sello de tiempo que le servirá de prueba para demostrar, en caso de disputa, la hora exacta en la que el mensaje fue enviado por la entidad emisora.

Como decíamos al principio de este apartado, ambos servicios, el de *No Repudio de Depósito* y el de *No Repudio de Envío*, sirven para generar pruebas acerca de un mismo evento (la emisión del mensaje), pero se diferencian en el destinatario de la prueba y en el tipo de garantías que se obtienen con la recepción de esa prueba. Además de esa diferencia, se puede constatar que los problemas que resuelve el primero son mucho más importantes y acuciantes que los que resuelve el segundo.

NO REPUDIO DE ENTREGA Y NO REPUDIO DE TRANSPORTE

El servicio de *No Repudio de Entrega* (*Non-repudiation of Delivery*) es con mucho el más importante de todos, ya que trata de resolver la amenaza más difícil de contrarrestar: que la entidad receptora niegue falsamente haber recibido el mensaje o que se niegue a reconocer el contenido del mensaje recibido.

Consecuentemente, si se provee el servicio de *No Repudio de Entrega*, la entidad emisora del mensaje obtiene una prueba, demostrable ante terceros, de que los datos han sido entregados íntegramente a la entidad receptora adecuada.

En la Figura 9.10 se recoge el esquema de funcionamiento bajo el que opera este servicio. Como puede observarse, la TTP de No Repudio tiene la misma estructura que la representada en la Figura 9.10, sólo que la funcionalidad que aquí desempeña es distinta. Podría pensarse, por tanto, en una sola TTP capaz de generar las evidencias necesarias para los tres tipos distintos de no repudio. Además de esta TTP específica, en la figura se representa también el conjunto de TTPs que, operando *fuera de línea*, sirven de ayuda para conseguir la adecuada generación de las firmas, su sellado y la correspondiente verificación posterior.

La TTP intermediaria reenviará el mensaje m a la entidad receptora exigiéndole la devolución de una indicación de aceptación del mensaje recibido. (Dependiendo del grado de confianza que exista entre la TTP y la entidad receptora, ese mensaje de aceptación estará o no firmado con la clave privada de la entidad receptora y vendrá

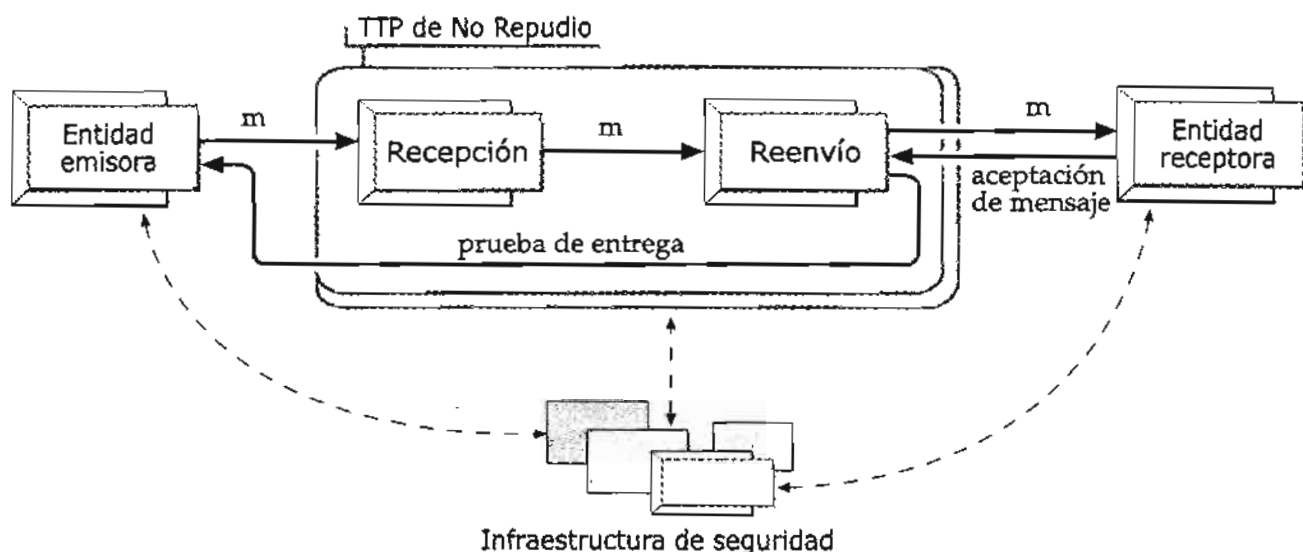


Figura 9.10. Servicio de No Repudio de Entrega.

acompañado de una marca fiable de tiempo.) A continuación, la TTP generará la evidencia que va a constituir la prueba de entrega cifrándola con su clave privada y añadiéndole un sello de tiempo facilitado por una TSA. En la evidencia debe constar la fecha y hora de entrega, la identidad de la entidad receptora y un resumen identificativo del mensaje enviado.

Siempre se puede argüir que si la entidad receptora no desea recibir un mensaje concreto puede dar la callada por respuesta y dejar establecida la duda sobre una posible disfunción en la transferencia de los datos. En primer lugar, cabe recordar que este esquema de comunicación está dentro de un dominio de seguridad en el que todas las entidades participantes aceptan las reglas que marca la política de seguridad allí vigente y que eso implica que la entidad receptora está **obligada** a respetar esas reglas y a responder a los mensajes que le envíe la TTP de No Repudio, con lo cual es de esperar que esa callada a que antes nos referíamos no suceda. Pero si se tiene la sospecha y se comprueba que esa entidad actúa maliciosamente no respondiendo (faltando a sus compromisos) siempre podrá ser «expulsada» de ese dominio, con lo cual **no puede** aparecer ante terceros como una entidad que facilita la provisión del servicio de No Repudio de Entrega.

Para asegurarse de que la entidad receptora sí está disponible y en condiciones de recibir el mensaje, se puede mejorar el esquema de la Figura 9.10 añadiendo un *mensaje sonda*, tal y como se recoge en la Figura 9.11. Este mensaje se lo envía la TTP a la entidad receptora indicándole que tiene para ella un mensaje con prueba de entrega y le informa de algunas particularidades de ese mensaje, como puede ser el tamaño y el tipo de información que contiene (texto, imágenes, formato de codificación o compresión de los datos, etc.) para que la entidad pueda decidir si está en condiciones de recibir un mensaje de esas características. En caso afirmativo respondería a la sonda con un mensaje de aceptación, lo que provocaría que la TTP le enviase el mensaje procedente de la entidad emisora. Caso de que la respuesta sea negativa, la TTP informaría a la entidad emisora indicándole que la entidad a la que pretendía hacer llegar los datos no está en condiciones de recibir un mensaje de ese tipo.

Tradicionalmente, lo que aquí hemos denominado como *No Repudio de Entrega* ha sido conocido como *acuse de recibo* y ha tenido su principal exponente en el Correo Postal convencional. Cuando se envía una carta certificada con acuse de recibo, el receptor de la misma debe firmar una cartulina indicando el día y la hora en el que ha recibido el envío. El Servicio de Correos devuelve, convenientemente cumplimentada, esa cartulina o tarjeta postal que sirve como prueba o evidencia de que la carta ha llegado a su destino. En muchísimas comunicaciones de las administraciones públicas con los ciudadanos se ha venido utilizando este procedimiento, que, aunque

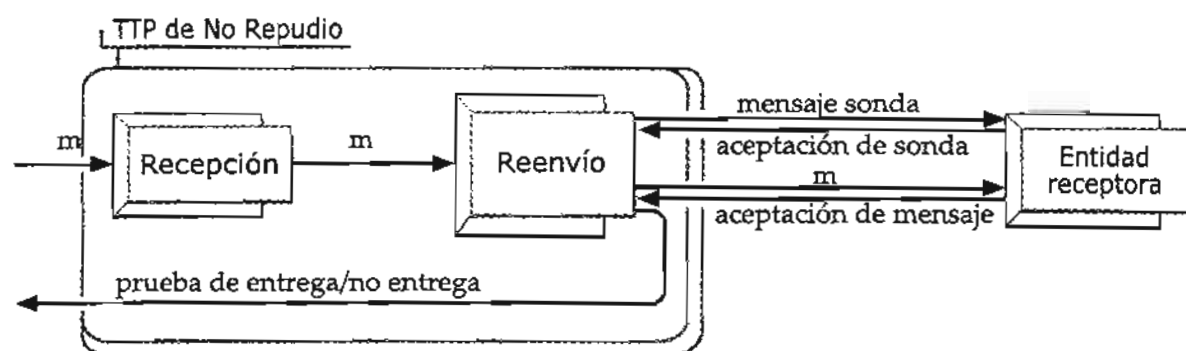


Figura 9.11. Mensaje sonda para comprobar la disponibilidad de la entidad receptora.

la seguridad puesta en juego es más que discutible, cuenta con suficiente aceptación social y con el necesario respaldo jurídico.

Esta presencia del *acuse de recibo* en las comunicaciones a través de correo postal ha dado lugar a que en la regulación jurídica de determinados actos administrativos se haya pensado trasladar a las comunicaciones a través de correo electrónico este tipo de exigencias⁷. Para ello hacen falta tanto soluciones técnicas como un marco jurídico adecuado que les dé cobertura. La confianza de los ciudadanos en este tipo de soluciones viene de la mano, como en el caso de la firma electrónica, de plantear servicios telemáticos que ofrezcan pruebas fiables y que tengan el necesario reconocimiento legal.

En este caso, dada la necesidad creciente de fomentar la presencia de tramitaciones administrativas a través de redes telemáticas, el soporte jurídico va por delante de las soluciones técnicas que se están poniendo en juego, las cuales adolecen de la necesaria seguridad: el correo electrónico, tal y como está implantado masivamente en la actualidad, no ofrece la generación de pruebas criptográficas robustas que sirvan para la aceptación por los ciudadanos de este tipo de prácticas.

El caso del correo electrónico presenta la particularidad de que en el extremo receptor del mensaje existen dos entidades diferenciadas: una que representa la estafeta final a la que llega el mensaje y otra que representa el buzón personal de la entidad destinataria. Sin pretender abordar ahora ni la estructura de las aplicaciones de correo electrónico ni los mecanismos de seguridad que éstas contemplan, conviene hacer una breve disquisición sobre la repercusión que, a efectos del no repudio, tiene la existencia de estas dos entidades receptoras.

En la Figura 9.12 se representa el modelo general del MHS (*Message Handling Service*), que es la propuesta conjunta de ISO e ITU-T sobre correo electrónico, más conocida como X.400, conforme a la numeración de la correspondiente Recomendación de este último organismo. El servicio de correo electrónico propuesto por la Internet Society (en la RFC 1891 y en la RFC 1894, entre otras muchas) guarda

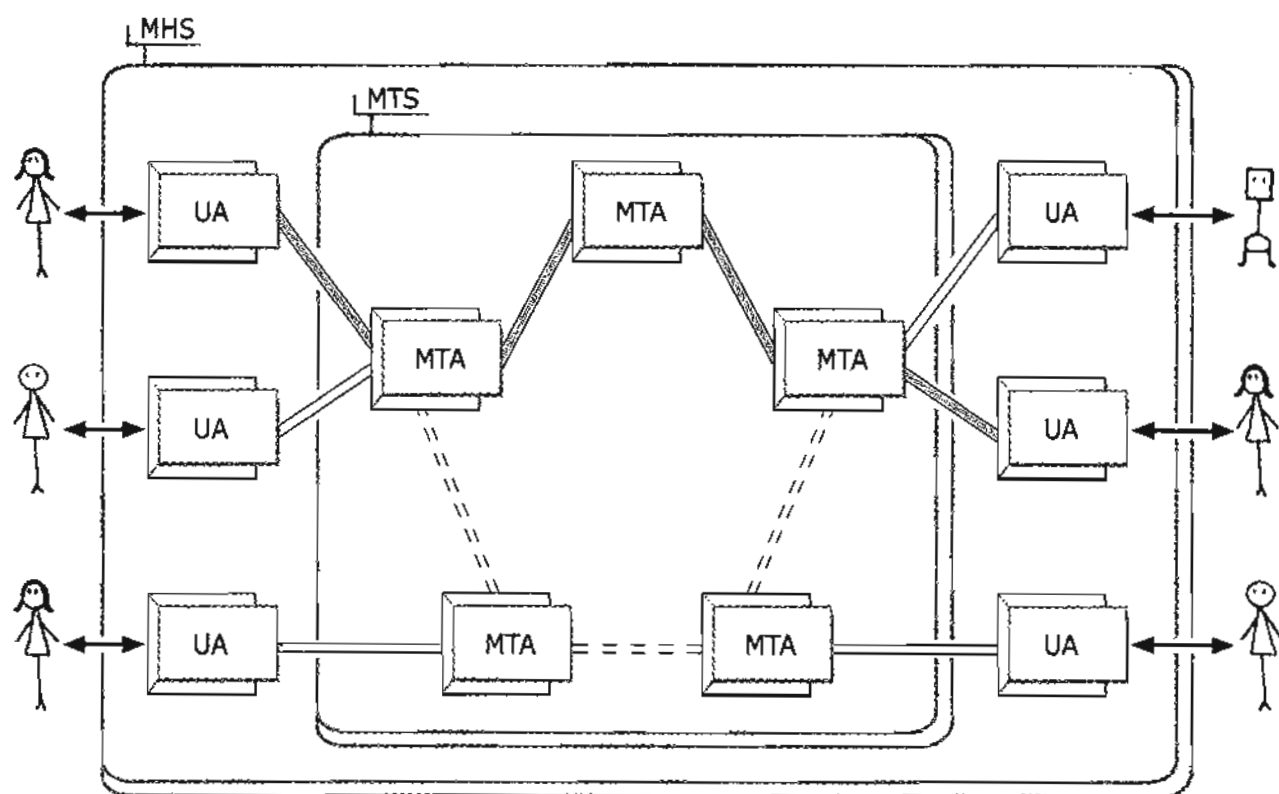


Figura 9.12. Modelo general del Correo X.400.

sustancialmente este mismo esquema (aunque planteado menos explícitamente), por lo que nos apoyaremos en él para los comentarios que siguen.

Como puede observarse, el usuario del servicio accede a una entidad telemática denominada *Agente de Usuario*, UA (*User Agent*), que depende de otra entidad que actúa como estafeta para un determinado conjunto de UAs. Esa otra entidad es el *Agente de Transferencia de Mensajes*, MTA (*Message Transfer Agent*), que colabora con otros MTAs para, mediante un procedimiento de almacenamiento y reenvío, llevar los mensajes de un punto a otro de la red hasta entregarlo en el MTA del que depende el UA de la persona destinataria del mensaje (el destinatario puede ser también un autómata).

Cuando llega un mensaje a un MTA, la operación de *entrega* (*submission*) se produce cuando el MTA coloca el mensaje en el buzón de la entidad destinataria (en el UA). En X.400 esa entrega está gobernada por un protocolo cuyo funcionamiento podría regularse para que solamente fuese efectiva cuando el usuario «abre» su correo, pero en la mayoría de las implementaciones existentes de correo (conformes con la RFC 1891 y la RFC 1894, o similares) esa entrega se realiza al margen de lo que haga la entidad propietaria del buzón.

Es decir, suponiendo que la confirmación de entrega que haga el MTA sea una pieza de información criptográficamente segura (cosa que no ocurre en la inmensa mayoría de los casos) de lo único que informa es de que el mensaje se ha depositado en el buzón del destinatario en determinada fecha y hora. Si éste, por cualquier motivo, no abre su correo hasta transcurrido un periodo largo de tiempo, se daría el caso de que la entidad emisora del mensaje obtiene una «prueba» robusta de algo por lo que no puede pedir responsabilidades a la entidad destinataria del mensaje. Este comportamiento tiene menos garantías que las que se ofrecen con los acuses de recibo tradicionales en las comunicaciones mediante papel. Y esto es algo inadmisibles. (Mucho menos fiable aún sería aceptar como prueba de entrega el mensaje de respuesta que, si le parece oportuno, puede enviar la persona destinataria en el momento de abrir el mensaje para proceder a su lectura.)

La implantación de aplicaciones de correo seguras pueden resolver algunos de estos problemas, aunque ello requeriría condiciones estrictas de seguridad en todos los agentes incluidos en la «cadena» de MTAs por la que atraviesa el mensaje, cosa difícil de imponer debido a la «liberalidad» bajo la que se ha venido extendiendo el servicio de correo electrónico. De hecho, el X.400 se definió en 1984 (aunque luego fueron apareciendo nuevas versiones) con la sana intención de ser un estándar único para el correo electrónico, de tal forma que el conjunto de MTAs constituye un sistema en sí mismo, denominado *Sistema de Transferencia de Mensajes*, MTS (*Message Transfer System*), que como tal sistema presenta hacia el exterior unas funcionalidades bien definidas. En la Figura 9.13 se representa el comportamiento del MTS en la transferencia de un mensaje.

En esta figura se aprecia cómo es el primer MTA de la cadena el que devuelve la *prueba de depósito* al UA que envió el mensaje, mientras la *prueba de entrega* es generada por el MTA al que está asignado el UA destinatario, aunque sea el primer MTA de la cadena el que se la entrega al UA que envió el mensaje. Está previsto, además, que estas pruebas sean criptográficamente seguras. La figura se ha representado deliberadamente para que se aprecie el gran parecido funcional con el esquema representado en las Figuras 9.9 y 9.10. Es decir, podemos afirmar que tal y como está concebido el X.400, el MTS se comporta, en su conjunto, como una «gran» TTP que sirve para proveer, entre otros, los servicios de no repudio (nada extraño, por cierto, si pensamos que el correo X.400 se diseñó con la mente puesta en emular lo que era en esa fecha el comportamiento del correo postal, donde era el Servicio de Correos

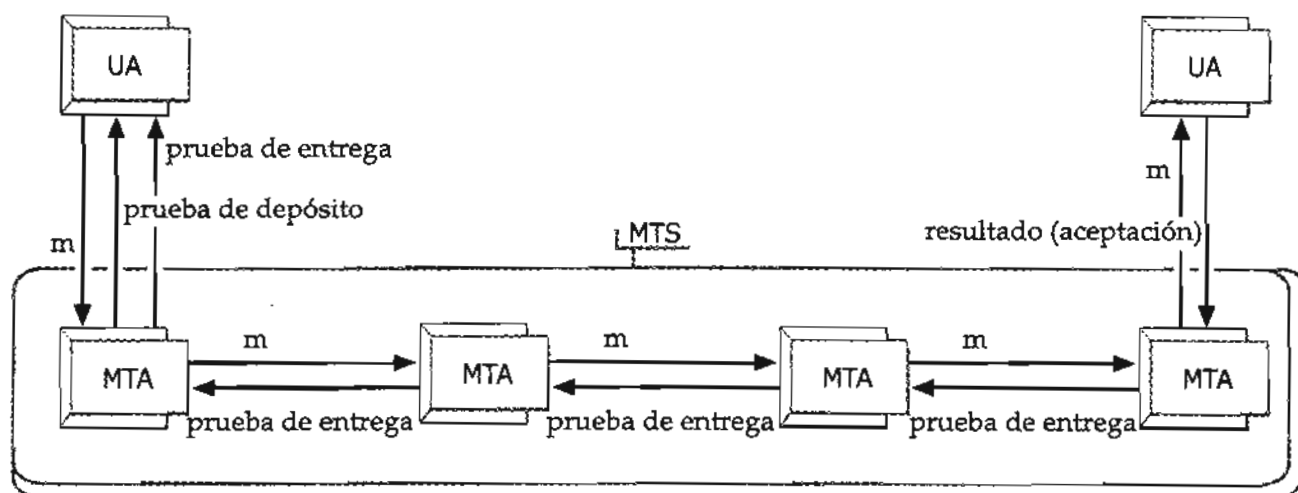


Figura 9.13. Cadena de MTAs en la transferencia de un mensaje.

—de titularidad estatal— el que hacía, en régimen de monopolio, de único intermediario entre quien enviaba una carta y quien la recibía).

Por todo ello, teniendo en cuenta la realidad actual, la pluralidad de proveedores y de aplicaciones de correo instaladas y la casi nula probabilidad de que se llegue a implantar una cooperación global entre todas ellas que emule, en cuanto a los servicios de No Repudio se refiere, un comportamiento global equivalente al representado en la Figura 9.13, teniendo en cuenta todo esto, decimos, lo más sensato es promover la implantación de **TTPs especializadas en No Repudio** con un funcionamiento similar al representado en las Figuras 9.9 y 9.10 que, mediante una política de seguridad adecuadamente definida, sirvan para dar satisfacción a unas necesidades tan acuciantes como las que hemos tratado de poner de manifiesto en los párrafos precedentes.

El servicio de *No Repudio de Transporte* (*Non-repudiation of Transport*) está definido para obtener garantías de seguridad en aquellas situaciones en las que, como es el caso representado en la Figura 9.13, entre la entidad emisora del mensaje y la entidad receptora existen un número indeterminado de entidades intermediarias que se encargan del adecuado «transporte» de éste hasta llegar a la entidad que se encarga propiamente de la entrega. Sería mejor decir «de transferencia» en lugar «de transporte» porque este último término se suele entender como referido al Nivel de Transporte, es decir, el Nivel 4 de la arquitectura OSI.

En cualquier caso, no merece mucho la pena polemizar acerca del nombre de este servicio porque debido a la funcionalidad que aporta, obtener pruebas fiables de que el mensaje ha llegado hasta una entidad que tiene acceso directo a la entidad receptora puede ser considerado como un caso particular del servicio de *No Repudio de Entrega*. En efecto, si nos fijamos en el escenario de comunicación reflejado en la Figura 9.11 apreciamos que cuando la TTP envía un *mensaje sonda* queda a la espera de recibir una confirmación proveniente de la entidad receptora. Si además tratamos de casar este escenario con el representado en la Figura 9.13, en el que la entidad que realiza la entrega es la última de una cadena, podríamos añadir a lo ya descrito que cuando la TTP «final» esté en condiciones de entregar el mensaje, al mismo tiempo que envía la sonda, puede generar una evidencia que mandaría a la entidad emisora del mensaje. En este caso, tenga o no éxito el sondeo, la entidad emisora obtiene una prueba de que el mensaje ha llegado a un punto en el que hay acceso directo a la entidad receptora, es decir, que se han salvado todas las dificultades que pudiesen estar relacionadas con la transferencia del mensaje a través de la red (aunque se puede

pensar que una funcionalidad similar se puede sacar también de un adecuado aprovechamiento del resultado de la sonda).

En cualquier caso, dada la importancia y trascendencia que tiene en sí el servicio de *No Repudio de Entrega*, pierden interés los matices y ventajas (dudosas) que puede aportar un servicio de *No Repudio de Transporte*.

NO REPUDIO DE CREACIÓN Y NO REPUDIO DE CONOCIMIENTO

El servicio de *No Repudio de Creación* (*Non-repudiation of Creation*) sirve para proteger a los usuarios de redes telemáticas contra el hecho de que una determinada entidad niegue falsamente haber creado un documento y ser responsable de su contenido.

Aunque guarda cierto parecido con el servicio de *No Repudio de Origen*, el ámbito y significado del servicio de *No Repudio de Creación* son bastante diferentes. En efecto, el *No Repudio de Origen*, tal y como aquí lo recogimos, se inscribe en el ámbito de la **transferencia de mensajes** y lo que asegura es la participación de una determinada entidad telemática en el envío de determinados datos, pero no ofrece pruebas acerca de quién sea el autor o autora del contenido de ese mensaje. El servicio de *No Repudio de Creación* sí ofrece pruebas, demostrables ante terceros, de este último extremo, al margen de que ese documento sea enviado o no en forma de mensaje.

El mecanismo por excelencia para proveer este servicio es la *firma digital* o, con mayor información complementaria, la *firma electrónica*. En realidad, de una forma o de otra, hemos estado haciendo referencia a este tipo de garantías y pruebas robustas a lo largo del capítulo anterior cuando hemos descrito la funcionalidad de la firma. Así, ya en el apartado «*Características de la firma electrónica*» dentro del epígrafe 8.6, recogíamos como una de sus particularidades que «El autor de la firma no puede repudiar su autoría» en el sentido de que «Quien accede al documento firmado adquiere una prueba, demostrable ante terceros, de que el mensaje ha sido firmado por quien dice ser el signatario». Como quedó demostrado, la firma electrónica cumple con todas estas expectativas al tiempo que cuenta con el respaldo jurídico necesario.

Mucho más sutil y menos significativo es el servicio de *No Repudio de Conocimiento* (*Non-repudiation of Knowledge*). Podemos definirlo diciendo que sirve para proteger a los usuarios de redes telemáticas contra el hecho de que una determinada entidad niegue falsamente haber tenido conocimiento de un mensaje al que ha accedido o le ha sido entregado. Es el equivalente al «recibí» que se añade en algunos intercambios de información y aunque no tiene por qué estar relacionado con el envío de mensajes (puede tratarse del acceso a una base de datos), este servicio puede proveerse usando las facilidades que ofrece una TTP especializada en la provisión del servicio de *No Repudio de Entrega*. Por tanto, aunque las garantías que proporciona tienen distinto objetivo, en la práctica, podemos considerarlo como un caso particular de este último servicio.

NOTAS AL CAPÍTULO 9

¹ La recomendación ITU-T X.812 *Marcos de seguridad para sistemas abiertos: Marco de Control de Acceso* (coincidente con la norma ISO/IEC 10181-3) presenta, como su título sugiere, una descripción general del servicio que nosotros utilizaremos tanto en lo que se refiere a las definiciones como a la nomenclatura que utiliza su versión en español. Otra

norma de estos organismos de la que hemos aprovechado algunos conceptos es la Recomendación ITU-T X.741 *Gestión de sistemas: objetos y atributos para el control de acceso* (coincidente con la norma ISO/IEC 10164-9). Dentro del ámbito del IETF existen varias RFCs que tocan algunos temas relacionados con el control de acceso; en el presente texto haremos especial referencia a las definiciones contenidas en la RFC 3281 *An Internet Attribute Certificate Profile for Authorization*, que a su vez se apoya en las definiciones de la recomendación ITU-T X.509 tan reiteradamente citada por nosotros.

² Al lector interesado en la especificación e implementación de controles de acceso sobre una Base de Datos compleja o una Base de Información distribuida, en las que las entradas puedan estar anotadas mediante diversos atributos o calificadores, se le recomienda estudiar detenidamente la especificación de seguridad del Directorio. Para ello, puede obtener una descripción resumida pero muy completa en [Ford94] o bien consultar directamente el conjunto de recomendaciones ITU-T X.500 o su equivalente en ISO.

³ Ya en 1985 el Departamento de Defensa hizo público su *Trusted Computer Security Evaluation Criteria* en el que dividía las políticas de control de acceso en DAC y MAC, a las que posteriormente se les añadió la RBAC. En las páginas web del NIST (organismo de normalización de los Estados Unidos) se puede acceder a <http://csrc.nist.gov/rbac/> donde se recogen los aspectos más significativos de esta técnica, se comentan los trabajos de normalización en curso y se incluye una lista de referencias para obtener información adicional, entre ellas el artículo original [FeKu92] que aquí hemos utilizado para la descripción de RBAC, y un libro [FKC03] publicado en 2003.

⁴ Una experiencia europea en la implantación de políticas RBAC utilizando certificados de atributos e infraestructuras PMI la constituye el proyecto PERMIS. En la web de dicho proyecto, en la University of Salford, se puede consultar, entre otros, el siguiente artículo divulgativo (probablemente redactado en 2002): *The PERMIS X.509 Role Based Privilege Management Infrastructure*, cuyos autores son D. W. Chadwick y O. Otenko.

⁵ Además de en la norma ISO/IEC 7498-2 (ITU-T X.800) *Security Architecture*, el tema del No Repudio también se aborda en la ISO/IEC 10181-4 (ITU-T X.813) *Marcos de seguridad para sistemas abiertos. Marco de No Rechazo*, y en la ISO/IEC FCD 13888-1 *Security techniques. Non-repudiation*, la cual hemos tenido en cuenta a la hora de seleccionar los distintos tipos de servicios de *No Repudio* que se describen en el presente texto, aunque la clasificación, definición y tipificación que aquí se recoge quedan bajo la responsabilidad (y la experiencia) del autor.

⁶ En [FeMa02] se comentan distintas leyes y disposiciones de la Administración Pública española en relación con el tema del no repudio. En concreto, se cita la Orden de 26 de diciembre de 2001 que establece criterios para la tramitación telemática de determinados procedimientos en el ámbito del Ministerio de Economía. En ella se crea un Registro Telemático para la presentación de escritos y solicitudes y se establecen condiciones para admitir la notificación telemática utilizando buzones de correo electrónico. Para ello se prevé la existencia de una lista actualizada «de proveedores que ofrezcan servicios de correo electrónico con las especificaciones que previamente se establezcan». De esa forma, la Administración puede enviar mensajes de respuesta con ciertas garantías acerca de su recepción por parte de quien inició el trámite administrativo.

⁷ En [FeMa02] se citan y comentan, entre otras, la *Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico*, que regula el acuse de recibo en la tramitación de contratos electrónicos; la omnipresente Ley 30/1992 sobre *Régimen jurídico de las Administraciones Públicas y de Procedimiento Administrativo Común* que admite la existencia de documentos en soporte electrónico y regula las condiciones que han de cumplir las notificaciones de la Administración; la Ley 24/2001 que modifica la ley anterior y añade detalles para el caso de que las notificaciones se hagan vía correo electrónico, y la Orden del Ministerio de Economía ya citada en la nota anterior. Puede apreciarse que las posibilidades que ofrecen las infraestructuras telemáticas en que se apoyan esas normativas son manifiestamente insuficientes para las necesidades allí requeridas. Para conseguir la aceptación por parte de los ciudadanos de esta nueva forma de llevar a cabo trámites administrativos es necesario no sólo cobertura jurídica, sino soluciones técnicas que ofrezcan la debida confianza.

CAPÍTULO

10

ESPECIFICACIÓN DE POLÍTICAS DE SEGURIDAD: CERTIFICACIÓN Y FIRMA

A lo largo de los capítulos anteriores, y de forma especial en los cuatro últimos, hemos venido haciendo reiterada referencia al tema de la *Política de Seguridad*. Quizás sea uno de los conceptos que, junto con los de *clave pública* y *certificado*, más frecuentemente han aparecido en estas páginas. No podía ser de otra manera porque son muchas y muy diversas las situaciones en las que la política de seguridad que se adopte condicionará grandemente el grado y calidad de la protección que se establezca sobre sistemas y protocolos.

En el presente capítulo no pretendemos hacer una recopilación ni una clasificación de todas esas situaciones porque sería tarea bastante tediosa y además algo estéril. Sería tediosa porque para entender adecuadamente cada una de las referencias habría que repetir el contexto en el que se hizo y sería estéril porque no añadiría nada nuevo. Precisamente la gracia del asunto está en entender que los distintos escenarios en los que se aplican servicios de seguridad que requieren soluciones específicas y adaptadas a las necesidades que en ellos se detecten.

Lo que sí es cierto es que en los temas relacionados con la certificación y la firma hemos venido posponiendo la matización de algunas cuestiones para abordarlas, de forma global, en este capítulo. Y eso haremos: dentro del ámbito de la certificación, se tratará de clarificar y establecer las diferencias entre *Política de Certificación* y *Declaración de Prácticas de Certificación (CPS)* que tanta relevancia tienen para un funcionamiento adecuado de las aplicaciones seguras, como puede ser el caso de la *Firma Electrónica*. Además, se abordará una especificación formal de este mecanismo de firma, así como lo relativo a políticas y prácticas de Sellado de Tiempo (tan necesarias para garantizar la validez de un documento firmado).

10.1. POLÍTICAS DE SEGURIDAD Y PROVEEDORES DE SERVICIOS

Supongamos la existencia de un dominio de seguridad en el que tenemos una aplicación telemática segura (o conjunto de ellas) prestando determinados servicios con el

concurso de una o varias CAs y alguna que otra TTP especializada (por ejemplo, una TSA de sellado de tiempo y una TTP de No Repudio).

Un primer escenario que podemos contemplar consiste en pensar que tanto las CAs como las TTPs han sido diseñadas y desarrolladas de forma expresa para este entorno de comunicación. Este es un escenario que, sin decirlo de forma expresa, hemos tenido en mente en la mayoría de ejemplos que hemos ido viendo a lo largo de los anteriores capítulos. En este caso, que se puede corresponder con situaciones reales, definiríamos:

- Una *política de seguridad* global para el desempeño de esas aplicaciones, como consecuencia de la cual
- definiríamos una *política de certificación*, de la cual se deduciría
- cuál debe ser el comportamiento de las CAs que hemos de desarrollar; y también
- definiríamos la *política de sellado de tiempo* y la correspondiente al *no repudio*, lo cual
- serviría para determinar el comportamiento tanto de la TSA como de la TTP de No Repudio.

Es decir, se trataría de un dominio «autosuficiente» en el que todo está bajo el control de una misma Autoridad de Seguridad que será responsable tanto de la decisión de *qué* debe hacerse como del diseño de *cómo* debe hacerse.

Pero este escenario no se corresponde con muchas otras situaciones en las que lo que se desea es implantar una aplicación segura, por ejemplo una transferencia de datos firmados electrónicamente, para lo que se necesita un servicio de certificación y un servicio de sellado de tiempo de los que no se dispone, por lo que, obviamente, resulta necesario buscar proveedores que los ofrezcan y, si lo que ofrecen casa con lo que se busca, *comprarles* esos servicios.

Existe una gran diferencia entre ambos escenarios. En este último caso, la Autoridad de Seguridad responsable del dominio tendrá que buscar qué es lo que ofrecen esos proveedores externos y elegir lo que mejor se adapte a sus necesidades y condicionantes. Para ello será necesario conocer con precisión:

- *qué* servicios y condiciones se compromete a cumplir cada proveedor (la *política* bajo la que se generan los certificados o los sellos de tiempo), y
- *cómo* lo cumple (lo que podemos denominar la *declaración de prácticas*).

Según comprobaremos más adelante, lo primero se puede deducir de lo segundo y puede documentarse de forma más genérica y menos detallada, aunque no por ello menos precisa.

Esto no significa que en un escenario de los que hemos denominado «autosuficientes» no sea posible realizar esta doble especificación del comportamiento de las CAs y TTPs, sino que en esas circunstancias esta diferenciación es menos necesaria.

En los siguientes epígrafes abordaremos esta doble perspectiva para estudiar en primer lugar (con cierto detalle y precisión) el importantísimo asunto de las *políticas de certificación* y la *declaración de prácticas de certificación* (CPS), y en segundo lugar (con menor atención) el correspondiente distingo que puede hacerse en lo relativo a los servicios de *sellado de tiempo* que tan necesarios resultan para la implantación de la *Firma Electrónica*. Aunque este mismo planteamiento se puede hacer sobre cualquier otro servicio que proporcione un proveedor, nos hemos limitado a lo que ya es una realidad tecnológica consolidada (las *prácticas de certifi-*

cación) y, en menor medida, a los planteamientos emergentes sobre requisitos exigibles a las TSAs.

10.2. CPS Y POLÍTICA DE CERTIFICACIÓN

Como acabamos de decir, existe una fuerte relación entre el concepto de *política de certificación* y el de *prácticas de certificación*. En los tres apartados siguientes trataremos de definir ambos conceptos con la mayor precisión posible y trataremos de ver la relación existente entre ambos.

Además, dentro de la estructura del certificado existen campos que permiten hacer referencia a la política y a las prácticas bajo las que ha sido generado, lo que sirve para que las entidades usuarias de un certificado concreto conozcan las posibilidades, las restricciones y las responsabilidades que su uso conlleva.

POLÍTICA DE CERTIFICACIÓN

Según hemos visto anteriormente, un *certificado de la clave pública* (al que denominaremos simplemente *certificado*) es emitido por una CA bajo unas determinadas condiciones, de forma que una entidad usuaria del certificado (*relying party*), fiándose de la firma de esa CA y de las condiciones bajo las que ha sido emitido, confía en la ligazón existente entre la clave pública certificada y entidad titular (*subject*) del certificado. Parte de esas condiciones se plasman en la *política de certificación*. También sabemos que un mismo certificado puede reconocer más de una política de certificación y que la entidad titular de un certificado puede ser otra CA que «hereda» en su totalidad (o solamente en parte) la potestad de emitir certificados bajo esas políticas.

Aunque, como también hemos visto, existen tres versiones del certificado, para los fines aquí perseguidos vamos a dar por supuesto que cuando nos refiramos al *certificado* estamos refiriéndonos a su *versión 3*, que admite la presencia de múltiples y diversas extensiones.

Apoyándonos en la definición recogida en la Recomendación X.509, podemos decir que una *política de certificación* consiste en un *conjunto de reglas*, previamente definidas e identificadas, que indican la *aplicabilidad de un certificado* a un colectivo particular y/o tipo de aplicación con requisitos de seguridad comunes. Por ejemplo, una determinada *política de certificación* podría indicar la aplicabilidad de un certificado en la provisión de servicios de autenticación en el intercambio de datos para la compraventa de bienes dentro de un rango prefijado de precios.

Es decir, el elemento distintivo son las *reglas* bajo las que se define.

También hemos visto con anterioridad (e insistiremos en ello en el siguiente epígrafe) que el documento donde se especifica una política de certificación concreta debe estar registrado mediante un OBJECT IDENTIFIER único y específico que permita:

1. Hacer referencia a él en algunas extensiones del certificado.
2. Que su contenido sea accesible a las entidades usuarias del certificado para que puedan conocer el significado exacto de sus reglas.

Esa especificación puede ser o bien textual (y por tanto inteligible para un ser humano), o bien formal (inteligible también para un agente automático), o ambas cosas a la vez.

DECLARACIÓN DE PRÁCTICAS DE CERTIFICACIÓN (CPS)

En la RFC 2527 *Internet X.500 Public Key Infrastructure Certificate Policy and Certification Practices Framework* [ChFo99] (apoyándose a su vez en los planteamientos de una asociación de abogados [ABA95]) se recoge que una *Certification Practices Statement* (CPS) es una declaración de las prácticas que una Autoridad de Certificación emplea en la generación de certificados.

Según esta RFC (que tiene solamente carácter informativo¹), una *Declaración de Prácticas de Certificación* (CPS) puede adoptar la forma de una proclamación por parte de los gestores de la CA acerca de los detalles de los sistemas robustos y fiables que utilizan y de las prácticas que emplean para la generación de certificados. Una CPS puede tener también el carácter de información técnica complementaria al contrato que pueda suscribir con la CA tanto quien pide que se le extienda un certificado como quien lo usa, y puede hacer referencia a otros múltiples documentos de carácter legal o contractual.

Por ejemplo, en la versión 1.2 de la CPS de la empresa *VeriSign* (que coincide, como se explica en nota aparte, con otras CPS de la española ACE), antes de iniciar la descripción propiamente dicha, incluye un resumen de derechos y obligaciones (que hacen referencia a diversos apartados de la declaración) cuyo formato es muy similar al de las cláusulas que se incluyen en un contrato de compraventa o de prestación de servicios.

Un detalle que puede resultar curioso, por la novedad que conlleva, es que una de esas advertencias iniciales al usuario consiste en prevenirle que para utilizar los certificados generados bajo esta política deben haber recibido previamente la formación adecuada sobre el uso de claves públicas, firma digital, certificados, PKIs y CPSs.

Esta exigencia lo que pone de manifiesto es que para utilizar de forma responsable servicios de seguridad avanzados es necesario tener un conocimiento técnico que esté en proporción con la importancia de la transacción o compromiso que se adquiera. Al igual que existe un aforismo que recuerda que *la ignorancia de la ley no exime de su cumplimiento*, en la resolución de un pleito o disputa no debe caber que una de las partes alegue *ignorancia de la tecnología* como eximente de las responsabilidades que ha contraído por el mero hecho de utilizarla².

Para que una declaración de prácticas de certificación pueda establecer compromisos y obligaciones a las entidades usuarias de los certificados deberán tomarse medidas para que las personas responsables de esas entidades usuarias tengan conocimiento o noticia acerca de la CPS bajo la que se ha generado el certificado.

Por ese motivo debe existir un sitio en la red con un **identificador único**, un URI, en donde se publique la CPS completa, al tiempo que debe existir un **aviso** o **nota** de redacción breve que avise a los usuarios acerca los elementos más significativos de las condiciones de aplicación de un certificado. De forma similar a lo dicho para el identificador de la política de certificación, la existencia de estos dos elementos de información permite:

1. Hacer referencia a ellos en algunas extensiones del certificado (en concreto, esto está ya previsto en el campo *policeQualifiers* dentro de la extensión *certificatePolicies* que analizaremos de nuevo en el epígrafe siguiente).
2. Que su contenido sea accesible a las entidades usuarias del certificado para que puedan tener conocimiento exacto de los compromisos que su uso conlleva.

RELACIÓN ENTRE CPS Y POLÍTICA DE CERTIFICACIÓN

Según lo que acabamos de ver en los anteriores apartados, es fácil apreciar que si bien la *Política de Certificación* y la *Declaración de Prácticas de Certificación* son cosas distintas, generadas con una intencionalidad distinta y provenientes de fuentes distintas (la primera de ellas de origen eminentemente técnico y la segunda de procedencia jurídica y contractual), a pesar de ser distintas, decimos, existen muchos puntos, regulaciones y restricciones que deberán aparecer tanto en una como en otra. Es decir, la intersección entre ambas no será el conjunto vacío.

Aunque nada hay claramente normalizado sobre el particular (la RFC 2527 que venimos aquí usando como referencia es un loabilísimo trabajo, pero no propone un perfil que pueda servir como condicionante de conformidad³), podemos decir que prácticamente todo lo que esté recogido en la política de certificación deberá formar parte de la CPS. Habrá algunos puntos de la CPS que no tendrán cabida en la definición de la política de certificación, como son: la gestión del ciclo de vida de los certificados, las cuestiones de custodia de las claves, la cualificación técnica del personal, las condiciones de seguridad física de los edificios que contengan a los equipos informáticos, etc. (cosas todas ellas que detallaremos en el epígrafe 10.4).

Otra diferencia importante entre ambas descripciones radica en la manera en que deben ser redactadas. La CPS debe ser un documento entendible y consultable por las personas que gestionan las entidades telemáticas que van a hacer uso de los certificados e, incluso en los temas que sean comunes con la política de certificación, esta descripción deberá hacerse con mayor detalle, justificando, dentro de lo posible, las decisiones que se adoptan. Por contra, la política de certificación debe ser clara y concisa y, a ser posible, definida formalmente (según hemos venido insistiendo) para que pueda ser interpretada de forma automática por los agentes telemáticos que han de manejar los certificados.

Podría darse el caso, ilustrado en la Figura 10.1, en el que una misma CA operando bajo una sola CPS pudiese emitir certificados bajo varias políticas de certificación distintas. Por ejemplo, la política que hemos denominado PC1 podría estar orientada a las necesidades de un conjunto de entidades finales (desde A1 a E1) implicadas en una determinada aplicación telemática, en tanto que las otras políticas estarían

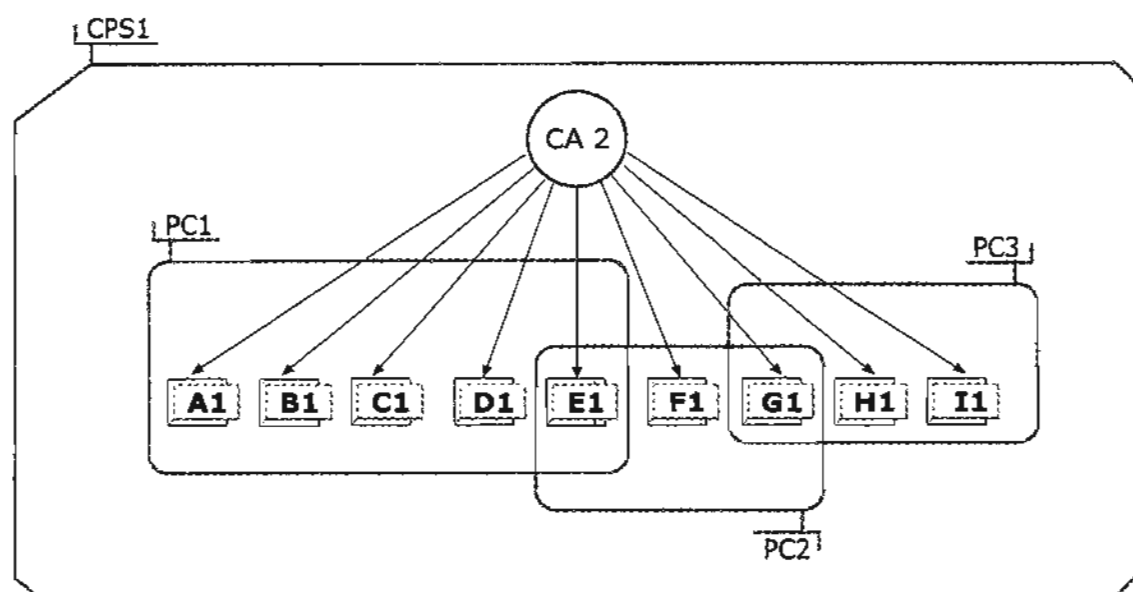


Figura 10.1. Tres políticas de certificación y una sola CPS.

orientadas a otros usuarios y otras aplicaciones. En la figura se ha reflejado el hecho de que haya entidades (caso de E1 y G1) involucradas en más de una de estas políticas. La situación puede ser aún más plural si pensamos que un mismo colectivo de entidades y una misma aplicación podrían operar, a su vez, bajo varias políticas distintas (dependiendo, por ejemplo, del valor monetario de una transacción).

De otro lado, podría darse el caso contrario: un conjunto constituido por múltiples CAs operando bajo la misma política de certificación pero cuyos certificados se han emitido bajo CPS distintas. La Figura 10.2 recoge esquemáticamente una situación como esta, en la que, para mayor sencillez, no se ha representado la posibilidad, bastante razonable, de que cada una de las CAs pueda soportar a su vez más de una política de certificación.

Un ejemplo sencillo [ChFo99] que ilustra una situación de este tipo podría ser un gobierno central que definiese una política general para la gestión de información confidencial sobre todo el personal funcionario. La definición de esta política estaría constituida por un sencillo conjunto de reglas que indicasen las personas para las que es de aplicación y las aplicaciones telemáticas concretas en las que debe usarse. Diferentes ministerios, departamentos o entes locales se verían afectados por esta política común, aunque podría tener cada uno de ellos sus propios certificados generados bajo CPSs distintas que, eso sí, soportarían, entre otras, la política común.

El esquema reflejado en la Figura 10.2 se corresponde bastante bien con situaciones que con frecuencia pueden presentarse en la vida real y que, además, son ilustrativas de otra característica diferenciadora entre CPSs y políticas de certificación. Veámoslo.

Un escenario de comunicación interesante puede ser aquel en el que una determinada política de certificación se defina de tal forma que sea aceptable y reconocible en ámbitos diversos pertenecientes a organizaciones diversas. Y ello no porque la defina una autoridad «superior» (como es el caso de un gobierno central en el ejemplo antes expuesto), sino porque determinada organización pueda reproducir y aceptar otra política previamente conocida que ha sido diseñada con objetivos compatibles con los detectados en esa organización. Sin embargo, ello no quita para que cada una de esas otras organizaciones pueda tener su propio ámbito de generación de certificados con su propia CPS.

Es decir, por regla general, la CPS está más ligada a cada organización concreta que genere certificados y a las garantías de protección que ofrece, tanto a las entidades

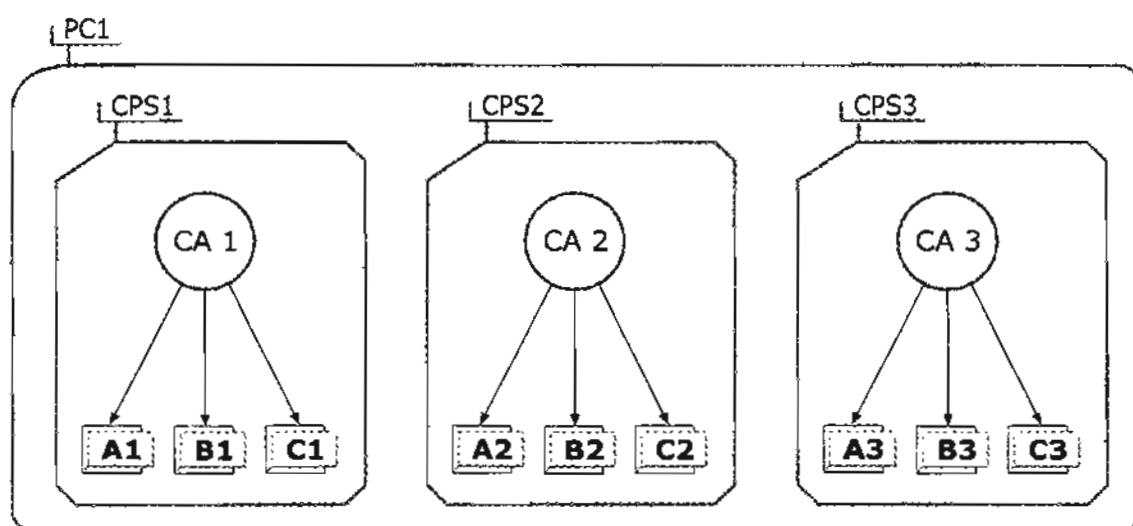


Figura 10.2. Tres CPSs y una sola política de certificación.

afiliadas que solicitan certificación de su clave pública como a las entidades usuarias de los certificados por ella emitidos. Por tanto, la CPS, que es una descripción muy extensa y detallada, no parece el medio más adecuado para buscar la *interoperabilidad* entre CAs y organizaciones.

La política de certificación, en cambio, sí que puede constituir una base firme para ese tipo de *interoperabilidad*: a medida que se vaya avanzando en propuestas de *normalización* y de *definición formal* de políticas será más fácil establecer *comparaciones* y definir *compatibilidades* aplicables de forma automática, facilitando, de esa manera, la comunicación entre organizaciones diferentes y diversas.

10.3. LIGAZÓN ENTRE EL CERTIFICADO, LA POLÍTICA DE CERTIFICACIÓN Y LA CPS

Multitud de detalles relacionados con la estructura del certificado dependen, como insistiremos más adelante, de la política de certificación y de la CPS bajo las que haya sido generado. Más concretamente, en la definición formal del certificado existen tres extensiones que se pueden usar para contener referencias y limitaciones acerca de estas políticas. Estas tres extensiones son:

- Correspondencias de políticas (*policyMappings*).
- Políticas de Certificación (*certificatePolicies*).
- Restricciones sobre la política (*policyConstraints*).

La primera de ellas la describimos y comentamos en el Capítulo 7 cuando se analizó su utilidad para la configuración de infraestructuras de certificación flexibles y adaptables a las necesidades de los usuarios. Ya allí recogimos, y aplicamos en un ejemplo, el interés que tiene esta extensión de cara a establecer compatibilidades entre políticas de certificación sobre la base de aceptar los identificadores de objeto de cada una de ellas. Nada más hay que añadir ahora a lo ya dicho.

La segunda también se analizó en ese apartado, aunque allí obviamos describir la definición del campo de calificadoros, directamente relacionado con la CPS. En el siguiente apartado se describe y comenta de forma más completa, ya que es, sin duda, la que mayor interés presenta de cara a los objetivos perseguidos en este epígrafe.

La extensión sobre restricciones sólo ha sido someramente citada con anterioridad. Está pensada para certificados emitidos a otra CA y, como su nombre indica, sirve para establecer limitaciones acerca de la validez de la política bajo la que esos certificados han sido generados. Le dedicaremos un apartado propio a su análisis.

POLÍTICAS DE CERTIFICACIÓN (*CERTIFICATEPOLICIES*)

La especificación formal de esta extensión, recogida en la RFC 3280 [HFPS02], es la siguiente:

certificatePolicies ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
 policyIdentifier CertPolicyId,
 policyQualifiers SEQUENCE SIZE (1..MAX) OF
 PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
 policyQualifierId **PolicyQualifierId,**
 qualifier **ANY DEFINED BY policyQualifierId }**

Como ya habíamos visto, esta extensión permite especificar una secuencia de una o más *PolicyInformation*, cada una de las cuales informa sobre una política concreta a través de dos campos distintos:

- Un identificador no ambiguo de la política de certificación, *CertPolicyId*, que es del tipo OBJECT IDENTIFIER y será por tanto una colección ordenada de enteros.
- Una secuencia de calificadores distintos cada uno definido por un identificador, *PolicyQualifierId*, que también es del tipo OBJECT IDENTIFIER y sirve para identificar la CPS bajo la cual la CA ha generado el certificado.

Aunque el campo de calificadores es optativo, el caso que aquí nos interesa es cuando sí están presentes y la extensión está marcada como crítica, lo que obligaría a la entidad usuaria del certificado a analizarlo en todos sus términos (incluidos los calificadores) si desea emplear la clave certificada para ciertas aplicaciones.

En el perfil definido en la RFC 3280 se especifican dos de estos posibles calificadores y se proporcionan sus correspondientes identificadores de objeto. Proponen que se usen de forma alternativa conforme a la siguiente definición:

Qualifier ::= CHOICE {
 cPSuri **CPSuri,**
 userNotice **UserNotice }**

CPSuri ::= IA5String

UserNotice ::= SEQUENCE {
 noticeRef **NoticeReference** **OPTIONAL,**
 explicitText **DisplayText** **OPTIONAL }**

NoticeReference ::= SEQUENCE {
 organization **DisplayText,**
 noticeNumbers **SEQUENCE OF INTEGER }**

DisplayText ::= CHOICE {
 ia5String **IA5String** **(SIZE (1..200)),**
 visibleString **VisibleString** **(SIZE (1..200)),**
 bmpString **BMPString** **(SIZE (1..200)),**
 utf8String **UTF8String** **(SIZE (1..200)) }**

El *CPSuri*, como su nombre sugiere, es un puntero, en forma de URI (*Uniform Resource Identifier*), que señala la ubicación de la CPS publicada por la CA. La especificación no matiza cómo han de emplear esta información las implementaciones locales de las entidades que hagan uso del certificado. Pero queda claro que en caso de litigio por un uso inadecuado del certificado, la CA queda a salvo de posibles responsabilidades distintas de las especificadas en su CPS.

De otra parte, el *UserNotice* contiene un pequeño texto pensado para ser mostrado al usuario del certificado (bien sea la propia entidad titular u otra entidad que esté haciendo uso de él) como advertencia previa a la utilización de la clave pública allí certificada. Este tipo está constituido, a su vez, por dos campos optativos:

- El primero de ellos, a través del tipo *NoticeReference*, identifica un mensaje de texto concreto mediante el nombre de una organización y un número de orden. Este mensaje no está incluido, por tanto, de forma explícita en el certificado sino que se identifica mediante esos dos parámetros. Así, en una implementación concreta, el programa que gobierne la entidad usuaria del certificado dispondrá de un fichero ordenado por organizaciones y números de orden conteniendo los mensajes que tiene que mostrar. Esta opción tiene la ventaja de que no incrementa el tamaño del certificado, además de permitir que los mensajes de advertencia estén redactados en varios idiomas y ser mostrados, dependiendo del entorno en el que se ejecute, en el idioma propio de la persona que ha de leerlo.
- La otra alternativa, representada por el campo *explicitText* consiste en la inclusión del mensaje de texto en el propio certificado. La RFC limita a 200 el número de caracteres, pero deja a discreción de los usuarios aceptar mensajes algo más largos.

En cualquier caso, lo que sí queda claro es la conveniencia de que esta extensión sea incluida en el certificado, y que sea marcada como crítica para así disponer de un mecanismo que permita dejar bien identificadas tanto la política de certificación como la CPS bajo las que se ha generado ese certificado.

RESTRICCIONES SOBRE LA POLÍTICA (*POLICYCONSTRAINTS*)

La extensión *policyConstraints* se aplica para establecer restricciones sobre la política en los certificados emitidos por CAs que están presentes en el camino de certificación. Estas restricciones pueden aplicarse en dos vías optativas: pueden ser usadas para prohibir compatibilidad de políticas o para exigir que todos los certificados pertenecientes al camino de certificación tengan un identificador de «política aceptable». Se define como:

```
PolicyConstraints ::= SEQUENCE {
    requireExplicitPolicy      [0] SkipCerts    OPTIONAL,
    inhibitPolicyMapping       [1] SkipCerts    OPTIONAL }
```

SkipCerts ::= INTEGER (0..MAX)

Si el campo *inhibitPolicyMapping* está presente, según se recoge en [HFPS02], el valor indica el número de certificados que pueden aparecer en el camino de certificación antes de que la correspondencia entre políticas deje de estar permitida. Por ejemplo, un valor 1 indica que la compatibilidad de políticas puede ser aplicada en certificados emitidos por la CA titular de *este* certificado, pero no en los siguientes certificados que se pudiesen generar sucesivamente.

Esta opción está relacionada con una salvedad a la que ya hemos aludido con anterioridad: la sucesiva emisión de certificados representa una sucesiva «transmisión» de confianza, pero cuanto más larga es la cadena, más dudas pueden aparecer acerca de la confianza. Cuando se trata de un dominio «cerrado» la cosa es poco grave porque todas las entidades operan bajo el mismo conjunto de políticas y todas las CAs

generan sucesivos certificados bajo ellas. Pero cuando una CA emite un certificado hacia *fuera* del dominio, puede ser interesante controlar los riesgos [ChFo99] de una «delegación transitiva» de la confianza. Es decir, el dominio A confía en el dominio B, el dominio B confía en el dominio C, pero el dominio A no quiere confiar en el dominio C. Para ello dispone de la opción *inhibitPolicyMapping* que limita el número de «saltos» en los que se mantiene la compatibilidad de «su» política.

Por otra parte, si el campo *requireExplicitPolicy* está presente, el número de saltos indica el número máximo de certificados sucesivos que pueden ser emitidos por una cadena de CAs antes de exigir que aparezca *de forma explícita* la política bajo la que está emitido. Las entidades usuarias de ese certificado «derivado» tendrán una lista de cuáles son las políticas aceptables y, conforme a ello, admitirán o rechazarán ese certificado. Con esta opción se trata de resolver el mismo problema que hemos descrito en el párrafo anterior, pero con distinto mecanismo. En este caso la solución consiste en que las CAs externas al dominio (y distantes cierto número de «saltos») estén obligadas a declarar su propia política cuando emitan certificados.

10.4. CONJUNTO DE ESTIPULACIONES PARA LA CERTIFICACIÓN

Una CPS debe servir para clarificar los derechos y obligaciones que contraen tanto las entidades que solicitan la emisión de un certificado como las que los generan o, simplemente, los utilizan. Es por esto por lo que *las estipulaciones* o cláusulas mediante las que se define una CPS están fuertemente orientadas al seguimiento del ciclo de vida del certificado desde que una entidad suscriptora solicita que le sea expedido hasta que termina su validez, de forma que una renovación del certificado supondría iniciar de nuevo el ciclo completo [FoBa97]. En la Figura 10.3 se representa gráficamente esta secuencia temporal por la que atraviesa el certificado en sus distintos estados.

Según la RFC 2527, un *Conjunto de Estipulaciones (Set of Provisions)* es una colección de prácticas y/o declaraciones de política que abarcan un conjunto de aspectos de interés tanto para la definición de las políticas de certificación como para la definición de la CPS⁴.

En los párrafos que siguen trataremos de resumir la formulación de estas estipulaciones comentando sus aspectos más significativos. Tanto la versión en español de los distintos términos como la interpretación de lo que es «más significativo» para nuestros objetivos, son responsabilidad del autor de este libro, recomendando al lector interesado en este tema consultar directamente la norma y otras CPSs a las que tenga acceso. Para la plasmación de estos comentarios nos apoyaremos tanto en la interpretación que hace la RFC acerca del significado de los distintos puntos como en lo deducido de nuestra propia experiencia y lo recogido en algunas CPSs reales⁵ (como las ya citadas de *VeriSign* y *ACE*).

Los componentes que constituyen estas estipulaciones son los siguientes:

1. Introducción: nombres, entidades y aplicabilidad.
2. Estipulaciones generales: cuestiones legales y prácticas.
3. Identificación y autenticación de entidades.
4. Requisitos de operación: gestión de claves y de certificados.
5. Controles de seguridad físicos, procedimentales y personales.
6. Controles técnicos de seguridad: generación y custodia de claves.



Figura 10.3. Ciclo de vida del certificado.

7. Perfiles del Certificado y de las CRLs.
8. Administración de la especificación.

Al análisis de cada uno de estos componentes dedicaremos uno de los siguientes apartados, excepto al último, que, como su nombre indica, está dedicado a asuntos de carácter administrativo, abordando los procedimientos a seguir para cambiar el contenido de la especificación y los procedimientos de aprobación y publicación.

Los objetivos que pretendemos cumplir con la descripción de estos componentes no son los de guiar al lector en la redacción de un conjunto de estipulaciones (como hemos dicho, para eso está la norma y las CPSs existentes), sino procurar que el lector estudioso adquiriera una visión de conjunto y se haga una idea de la multitud de detalles que es necesario tener en cuenta para definir adecuadamente una política de certificación y una CPS (por ejemplo, sólo el índice de la versión 2.2 de la CPS de *VeriSign* ocupa ocho páginas).

Por esa razón, aunque resulte a veces algo tedioso, abordaremos todos y cada uno de los apartados recogidos en la RFC 2527 acompañados de brevísimos comentarios que ayuden a entender el tipo de información que es necesario proveer en cada uno de ellos. También puede parecer que a veces existe redundancia entre algunos puntos de distintos apartados; ello es debido al hecho de que en las estipulaciones se tocan con frecuencia algunos temas (por ejemplo las claves) desde distintos ángulos de vista, lo que consecuentemente conlleva la necesidad de referencias cruzadas entre lo que se especifique en esos diferenciados apartados.

NOMBRES, ENTIDADES Y APLICABILIDAD

A modo de introducción a las estipulaciones debe aparecer un apartado en el que se introduzca el conjunto de estipulaciones y algunos detalles para identificar a la organización que las ha generado, así como a las entidades y aplicaciones que se contemplan. Podemos destacar los puntos siguientes:

a) *Identificación*

Se deberán incluir los nombres telemáticos específicos de los distintos agentes presentes: CAs, RAs, Repositorios, etc., así como los identificadores de objeto (ASN.1) que estuviesen definidos.

b) *Conjunto de entidades y aplicabilidad*

b1) Cuántas CAs y de qué tipo existen. Si hay más de una habrá que indicar cuál de ellas es la CA raíz y cómo están organizadas las restantes CAs, matizando el ámbito de competencia y organizativo de cada una de ellas.

b2) Cuántas RAs y de qué tipo existen. Si el entorno es geográficamente muy amplio puede que existan diversas RAs locales para facilitar la «comparecencia» de las personas que deseen recibir un certificado.

b3) Quiénes pueden ser las entidades finales que reciban un certificado. Podemos llamarlas también «titulares» de un certificado, o «suscriptoras». Si, por ejemplo, la CA de la Universidad X sólo emite determinado tipo de certificado para sus profesores, una entidad usuaria de esos certificados sabe que una persona, por el mero hecho de ser titular de uno de ellos, es profesor o profesora de esa universidad.

b4) Aplicabilidad. Es de suma importancia dejar constancia del tipo de aplicaciones en las que se puede utilizar cada certificado. Puede darse el caso de que en esa organización se emitan distintos tipos dependiendo de las garantías que se exijan para cada uno de ellos (por ejemplo, *VeriSign* ofrece certificados Clase 1, Clase 2 y Clase 3, dependiendo de las exigencias y responsabilidades que conlleven). En cada caso, será preciso especificar:

- i) Lista de aplicaciones para las que el certificado *se adapta*. Puede tratarse de correo electrónico, firma de contratos, compra de bienes a través de la red, etc.
- ii) Lista de aplicaciones para las que *está prohibido* el uso de los certificados.
- iii) Lista de aplicaciones para las que *está restringido* el uso de los certificados. Si se incluye esta lista, queda también implícitamente especificado qué otros usos están prohibidos, siendo innecesaria la existencia de la otra lista.

c) *Detalles de contacto*

Será necesario dejar reflejado el nombre y dirección de la persona responsable del registro, actualización e interpretación de la política que se esté especificando.

Asimismo, deberá figurar el nombre y dirección telemática de una persona de contacto a la que se puedan dirigir los suscriptores o usuarios de los certificados.

ESTIPULACIONES GENERALES: CUESTIONES LEGALES Y PRÁCTICAS

En este apartado se abordan una serie de cuestiones relacionadas con obligaciones, garantías, responsabilidades y compromisos legales. Es uno de los apartados de las estipulaciones que deberá ser cumplimentado con mayor detalle y precisión. Cabe destacar los siguientes aspectos:

*a) Obligaciones**a1) Obligaciones de la CA*

- i) Declarar el compromiso a mantener activa su funcionalidad conforme a los controles físicos y lógicos que se especifican en otros apartados.
- ii) Notificar la emisión de un certificado al suscriptor o suscriptora titular del certificado. Podría, en su caso, entregar el certificado mediante un testigo de seguridad (por ejemplo, una tarjeta inteligente). Si la generación del par de claves del titular la realiza la propia CA, éstas se entregarían también en el mismo testigo de seguridad.
- iii) Notificar al titular del certificado la revocación o suspensión de su certificado. Deberá explicitarse el medio mediante el cual realizará esta comunicación que debe contar, en cualquier caso, con una prueba de no repudio de entrega.
- iv) Explicitar el medio mediante el cual notificará la revocación o suspensión de certificados a los restantes miembros suscriptores y a otras entidades usuarias.
- v) Informar acerca de los procedimientos que se seguirán para cumplir con las regulaciones legales vigentes en materia de generación de certificados.

a2) Obligaciones de la RA

- i) Mantener operativa la estructura técnica y administrativa necesaria para el registro de suscriptores (y compromisos que adquieran) para que la CA correspondiente emita el certificado.
- ii) Realizar la identificación y autenticación de las personas o entidades candidatas a ser titulares de un certificado, conforme a reglas precisas en función del tipo de certificado que demanden. Si existen varios tipos de certificados, los requisitos en cuanto a identificación personal podrán variar desde una mera comunicación telemática a la exigencia de comparecencia física de la persona solicitante acompañada de documentos acreditativos de su identidad y del cargo o representación que dice ostentar.
- iii) Informar acerca de los procedimientos que se seguirán para cumplir con las regulaciones legales vigentes en materia de registro y protección de datos personales.

a3) Obligaciones de la entidad suscriptora

- i) Dar por buenas todas y cada una de las especificaciones contenidas en las estipulaciones, aceptando las restricciones en el uso de la clave certificada y asumiendo los compromisos económicos y las responsabilidades jurídicas que ello conlleva.
- ii) Proteger y custodiar su clave privada protegiendo el acceso al testigo de seguridad en el que se encuentre almacenada conforme se establezca en las estipulaciones, haciéndose responsable de las consecuencias que puede acarrear la negligencia en esa custodia. Siempre que las circunstancias lo permitan, el uso de tarjetas inteligentes con mecanismos de autenticación biométricos facilita mucho las cosas y minimiza los riesgos en la custodia de las claves.
- iii) Comprometerse a comunicar de forma inmediata a los gestores de la organización cualquier situación que comprometa la seguridad de sus claves (por ejemplo, la pérdida de la tarjeta inteligente o divulgación del PIN de acceso).

- a4) Obligaciones de las entidades usuarias de los certificados
 - i) Aceptación de los propósitos para los que ha sido generado el certificado que utilicen.
 - ii) Aceptación de responsabilidades en cuanto a la verificación de firmas y revisión del estado de revocación o suspensión de los certificados.
 - iii) Tener conocimiento de las responsabilidades jurídicas que el uso de los certificados conlleva.
- a5) Obligaciones del Repositorio
 - i) Mantener información actualizada acerca de los certificados emitidos y del estado en que se encuentran (activo, suspendido, revocado).
 - ii) Publicar periódicamente la información necesaria sobre revocación y suspensión de certificados.

b) Garantías, limitaciones y responsabilidades financieras

En las estipulaciones deberán aparecer con tanto detalle como requiera el caso todo lo relativo a seguros, compromisos y reparaciones por perjuicios producidos, tales que: tipos de daños cubiertos, negligencia y fraude, limitaciones de pérdidas, capacidad financiera de la CA, indemnizaciones por parte de la CA o RA, indemnizaciones por parte del usuario de los certificados, etc.

c) Interpretación y compromisos legales

Se detallarán todas y cada una de las disposiciones legales vigentes en el ámbito donde se distribuyan y utilicen los certificados y se definirá el procedimiento para la resolución de conflictos. Este es un tema espinoso si estamos pensando en entornos de comunicación amplios en los que participen ciudadanos de distintos países con normativas distintas, razón de más para desear que se armonicen las legislaciones y se disuelvan las fronteras (no sólo para el movimiento de capitales).

Asimismo, se deberán concretar algunas aclaraciones legales en lo que se refiere a los términos jurídicos en los que están redactadas las estipulaciones (divisibilidad, supervivencia, fusión, novación, notificaciones y otras zarandajas).

d) Costes y tarifas

Cuando se trate de una organización que ofrece sus servicios a cambio de una prestación económica deberá estar claramente especificado y actualizado el importe de cada uno de ellos.

e) Publicación y Repositorio

Será necesario incluir un conjunto de estipulaciones para remarcar las obligaciones que adquieren los gestores del entorno de certificación en lo que se refiere a la publicación de información. Algunos extremos a considerar pueden ser:

e1) Publicación de información de la CA. Obligaciones de las CAs en cuanto a publicar informes sobre sus prácticas, sus certificados y el estado actual de cada uno de ellos. Un mecanismo útil puede ser desarrollar un agente en el que se almacenen los *históricos* de todas las operaciones y estados por los que ha atravesado cada certificado.

e2) Frecuencia de publicación. Por ejemplo, la publicación de revocaciones puede estar sometida a una periodicidad distinta dependiendo de la importancia (Clase 1, Clase 2, etc.) del certificado y de los riesgos distintos que comporte.

e3) Controles de acceso a la información publicada (incluida política de certificación, CPS, certificados, status de certificados, CRLs, etc.) y requisitos de uso de los repositorios que estén operativos.

f) Auditoría

La especificación de auditores externos puede ser uno de los elementos que aporten más confianza a las personas que hipotéticamente pueden ser suscriptoras o usuarias de los certificados. Por eso en las estipulaciones deben aparecer lo más detalladamente posible asuntos tales que: identidad de los auditores, frecuencia de las auditorías, asuntos auditados, acciones resultantes sobre deficiencias encontradas y comunicación de resultados.

g) Confidencialidad

El hecho de que los certificados de la clave pública (también llamados *certificados de identidad*) contengan tanta información personal hace sumamente importante que se estipule (de conformidad con la legislación vigente en materia de protección de datos) todo lo relativo a qué datos se consideran confidenciales y cuáles no, así como los procedimientos de denuncia ante los poderes públicos por causa de incumplimiento.

h) Derechos de propiedad

Deberán quedar claramente establecidos los derechos de propiedad sobre las claves, los certificados, la CPS y otros elementos de información presentes en la infraestructura de certificación.

IDENTIFICACIÓN Y AUTENTICACIÓN DE ENTIDADES

La identificación de las entidades que han de participar en el escenario es de vital importancia porque, dado el carácter virtual y no presencial de las operaciones y transacciones telemáticas que pueden ser llevadas a cabo, la validez de los resultados depende de la adecuada autenticación de los participantes. Algunas cuestiones que hay que reseñar son:

a) Registro inicial

Deberán fijarse los tipos de nombres que se admiten (X.500, FRC 822, URL, etc.) y los componentes que deben estar presentes en su estructura para determinar su

unicidad y para que sean inteligibles e interpretables. Deberán definirse reglas para la resolución de disputas sobre conflictos de nombres y se especificarán los documentos acreditativos y pruebas de identidad que se han de exigir a las personas solicitantes de certificados.

b) Procedimientos para cambio de clave

Cambiar de clave significa cambiar el principal elemento identificativo y distintivo de los usuarios y de los agentes telemáticos dentro del entorno de certificación. La necesidad de cambiar de clave se puede producir por diversos motivos, el más común de ellos debido a revocación del certificado anterior. Los procedimientos de identificación deben ser rigurosos y muy similares a los empleados en el procedimiento inicial de registro.

c) Procedimientos de autenticación ante peticiones de revocación o suspensión

Estas peticiones representan una alteración importante en el funcionamiento global del sistema, por lo que tienen que estar muy bien reguladas tanto las causas que lo permiten (véase siguiente apartado) como la identidad y autenticación de las personas que las solicitan.

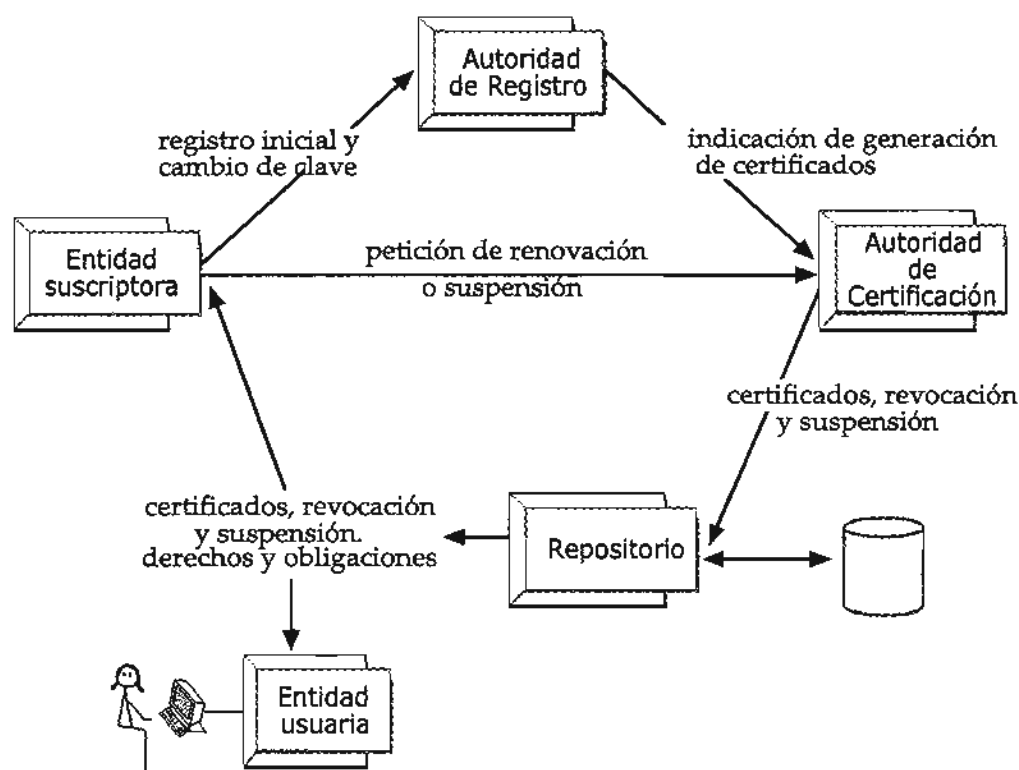


Figura 10.4. Relación entre entidades y agentes contemplados en la CPS.

REQUISITOS DE OPERACIÓN

Este es otro de los apartados de las estipulaciones que mayor repercusión tiene desde un punto de vista técnico, por lo que comentaremos con cierto detenimiento algunos de los puntos más significativos que abarca. Cabe resaltar:

a) Requisitos para la solicitud, generación y aceptación de certificados

Como su epigrafiado indica, se definirán con detalle (en función del nivel de seguridad del tipo de certificado) los requisitos exigidos a las entidades candidatas a obtener un certificado, así como los procedimientos para su generación, entrega y aceptación.

b) Suspensión y revocación de certificados

Como venimos insistiendo, este es un asunto bastante delicado que es necesario dejar bien clarificado. La suspensión es una medida de mucha menor trascendencia que la revocación porque, al final, el certificado puede acabar siendo revocado, pero también puede recuperar su estado de activo si desaparecen las circunstancias que provocaron la suspensión. En la RFC 2527 se recogen hasta 15 cuestiones distintas que conviene dejar reguladas. Las resumiremos en las siguientes:

b1) Circunstancias bajo las cuales un certificado puede ser revocado o suspendido (bajo petición de su titular o por decisión de los gestores de la CA que lo generó). Puede definirse, a su vez, más de un tipo de suspensión (cautelar, temporal, definitiva) dependiendo del grado de compromiso detectado.

b2) Procedimientos para la petición de revocación o suspensión.

b3) Límites del periodo de suspensión.

b4) Frecuencia y procedimiento de emisión de CRLs e informes sobre el estado de los certificados.

b5) Procedimientos y posibilidades para que las entidades suscriptoras y usuarias puedan consultar el estado de los certificados (incluido consultas en línea).

c) Procedimientos de auditoría de seguridad

c1) Tipos de eventos a registrar.

i) Peticiones del administrador de la RA o de los gestores de CAs para crear, revocar o suspender un certificado.

ii) Creación, emisión y revocación de certificados.

iii) Notificación de clave en peligro y notificaciones a las entidades suscriptoras.

iv) Emisión de CRLs y actualización de históricos en el Repositorio.

c2) Frecuencia de procesamiento, periodo de conservación y protección de los registros de auditoría.

c3) Valoración de vulnerabilidades.

d) Almacenamiento y archivo de registros

Una base imprescindible para responder ante auditorías y reclamaciones consiste en mantener un archivo amplio y completo sobre todo tipo de eventos que se produzcan en el dominio de certificación. Será necesario fijar en las estipulaciones los procedimientos de almacenaje y conservación de copias de seguridad con el añadido de sellos de tiempo para mayor garantía de verificabilidad.

e) Recuperación ante desastre y ante pérdida de seguridad

Como es bien sabido, la pérdida de seguridad en los elementos centrales de un dominio de certificación supone el derrumbe de todo el tinglado motado en torno a los certificados. Por eso es fundamental tener previsto la ocurrencia de este tipo de circunstancias y tener establecidas las medidas que se adoptarán en cada caso para restablecer la seguridad del entorno. Cuatro son las circunstancias que la RFC 2527 apunta que deben ser abordadas:

- e1)* Recuperación cuando los recursos físicos, el software o los datos están corrompidos.
- e2)* Recuperación cuando la clave pública de la CA es revocada.
- e3)* Recuperación cuando la clave privada de la CA está bajo sospecha o comprometida.
- e4)* Procedimientos de la CA para asegurar sus facilidades de seguridad después de un desastre natural o de otro tipo.

f) Cese de actividad de la CA

El que una CA deje de prestar servicios no significa que todos los certificados emitidos por ella dejen de tener validez de forma inmediata. Este tipo de situaciones deben estar previstas en la normativa jurídica que regule estas actividades. En concreto, el Real Decreto-Ley 14/1999 sobre firma electrónica indica que la CA deberá, antes de cesar en su actividad, comunicar a todo suscriptor de un certificado que no haya sido revocado o que no haya expirado, con una antelación de 90 días, el cese definitivo de su actividad y, en su caso, comunicar la posible sucesión por parte de otra entidad prestadora de servicios de certificación. En las estipulaciones deberá quedar reflejado el tipo de indemnizaciones conforme a las responsabilidades financieras de los gestores de la CA.

CONTROLES DE SEGURIDAD FÍSICOS, PROCEDIMENTALES Y PERSONALES

Se trata de controles no estrictamente técnicos desde un punto de vista de las tecnologías de la información y las telecomunicaciones (TICs), a los cuales la RFC 2527 dedica un apartado específico (el siguiente), pero que no por ello dejan de tener una gran importancia a la hora de redactar las estipulaciones de una CPS. Antes al contrario, gran parte de la confianza que los suscriptores y usuarios puedan depositar en los gestores del sistema (en relación con la adecuada custodia de las claves) descan-

sará en el convencimiento que tengan de que, desde un punto de vista físico y organizativo, se toman las medidas necesarias para conseguirlo. Esa es la razón por la que en las especificaciones de CPS existentes se aborde este conjunto de estipulaciones con tanto espacio y pormenor, aunque aquí nos limitemos simplemente a referirlas y comentarlas brevemente.

Estas estipulaciones deben especializarse para cada uno de los agentes presentes en el sistema: CAs, RAs, Repositorio, Bases de Datos, Archivos, etc. Se dividen en tres grupos:

a) Controles físicos

Se considera necesario que en las estipulaciones queden reflejadas con detalle las medidas de seguridad que se adoptan en cuanto a: localización y construcción del inmueble (ubicación del equipo), acceso físico, alimentación eléctrica y aire acondicionado, exposición ante inundaciones, prevención y protección contra incendios, y un largo etcétera.

b) Controles de procedimiento

Será necesario describir los distintos cargos de confianza y las tareas que llevan a cabo, el número de personas dedicadas a cada una de ellas, la identificación de los responsables, etc.

c) Controles de seguridad personales

Deberá especificarse de forma ordenada y selectiva, en función de los cargos que han de ocupar, los requisitos exigidos a las personas que realizan las distintas tareas. Estos controles consistirán en actuaciones tales como: comprobación de calificaciones, experiencias y acreditaciones; requisitos de aprendizaje y procedimientos de instrucción para cada cargo; frecuencia y procedimientos de reciclaje de conocimientos; frecuencia y secuencia de rotación entre cargos, etc.

CONTROLES TÉCNICOS DE SEGURIDAD

No es necesario insistir en la trascendencia que estos controles tienen de cara a la confianza que pueden depositar los usuarios en los certificados emitidos. Estos controles deben definirse y matizarse para todos y cada uno de los agentes telemáticos y sistemas presentes en el entorno de certificación. Entre los más significativos podemos reseñar los siguientes:

a) Generación e instalación del par de claves

Este es un asunto de capital importancia tanto en la definición de la política de certificación como de la CPS bajo las que se regirá el dominio. Las cuestiones con ello relacionadas, algunas de las cuales se resumen a continuación, deberán ser abordadas para todas y cada una de las entidades presentes, incluidas las entidades finales.

a1) *Quién genera las claves de cada entidad y cómo se proporciona la clave privada a cada entidad.* En algunos entornos se decide que la generación del par de claves (privada y pública) de las entidades suscriptoras se realice en las instalaciones centrales. En este caso, las claves son introducidas en un testigo de seguridad que será entregado a la persona indicada para su uso y custodia. En ese momento, el centro generador de claves debe borrar la clave privada de la entidad usuaria para que sea ella, y sólo ella, quien posea esta clave, responsabilizándose (incluso jurídicamente) de todas las operaciones de cifrado que con esa clave se realicen. Esta forma de actuar tiene algunas ventajas organizativas y una gran desventaja: a la entidad final siempre puede caberle la duda de que ese borrado se haya producido realmente.

Si las entidades finales están facultadas para generar por sí mismas su propio par de claves, podrán almacenar por sus propios medios la clave privada (por ejemplo en una tarjeta inteligente), remitiendo, después, la clave pública a la CA junto con la solicitud de un certificado. De esta manera, es insoslayable la responsabilidad que el usuario contrae en cuanto al uso y custodia de sus claves, no existiendo resquicio alguno que le permita atribuir a terceros una posible firma que haya sido realizada usando su propia clave privada. Si en las comunicaciones convencionales mediante papel es la persona signataria quien se inventa su propia firma caligráfica, parece razonable pensar que sea ella también la responsable de la generación de la clave privada que le servirá para realizar firmas electrónicas o para descifrar mensajes a ella dirigidos.

a2) *Cómo se proporciona de forma segura la clave pública de una entidad a la CA que ha de generar el certificado.* Habrá que decidir qué medio es el más adecuado para ello. En principio, cuando se trata de entornos geográficos muy extensos, lo más apetecible es emplear un medio telemático seguro para ello, pero existe el problema de que *antes* de que se establezca la certificación no existen, por lo general, otras claves públicas en las que confiar. Con todo, lo más seguro es emplear un medio presencial y no telemático para esta *primera* toma de contacto.

a3) *Entrega de la clave pública de una CA a los usuarios.* Si se trata de la CA de mayor jerarquía habrá que recurrir a varias vías en paralelo para ponerla en conocimiento de las entidades usuarias, algunas de las cuales conviene que sean vías extratelemáticas. Si se trata de CAs dependientes de otra CA, bastará con el certificado que le haya sido expedido.

a4) *Tamaño de las claves y parámetros de calidad.* Dependiendo de cada caso y de las responsabilidades que haya detrás del uso de un certificado, habrá que matizar el tamaño exigido para la clave y la calidad de los parámetros empleados en su generación.

a5) *Soporte utilizado en la generación de claves.* Habrá que definir cuáles son los sistemas y algoritmos utilizados en cada caso, matizando si la generación se hace mediante hardware o mediante software, y si se emplean para ello equipos homologados (cuya existencia reclama la Directiva europea sobre Firma Electrónica).

a6) *Propósito de uso de las claves.* En certificados X.509, estos propósitos se corresponden con los que se pueden reflejar en la extensión del mismo nombre. Para un mismo par de claves se puede expedir más de un certificado, cada uno bajo una política distinta. Asimismo, puede ser conveniente especializar los pares de clave para usos distintos: puede ser interesante usar una clave privada para firmar documentos y otra clave privada distinta para descifrar mensajes confidenciales cifrados con la clave pública correspondiente.

b) *Protección de la clave privada*

La protección de la clave privada es vital para conservar la ligazón que esa clave tiene con la identidad de las personas que gestionan las entidades (y las responsabilidades que de ello se derivan). En las estipulaciones deberán tenerse en cuenta aspectos tales que:

b1) *Estándares requeridos para el módulo criptográfico.* Es conveniente indicar con cuáles de los estándares de evaluación (y con qué niveles dentro de ellos) cumplen los distintos módulos criptográficos y los sistemas implantados. Para ello se pueden apoyar en el FIPS 140-1 (impuesto por el gobierno de Estados Unidos), en el europeo ITSEC o en otros criterios de evaluación de la seguridad.

b2) *¿La clave privada es un secreto compartido?* Para garantizar la integridad de la clave privada (principalmente de las CAs y de los sistemas más sensibles) se puede utilizar un procedimiento de secreto compartido, consistente en dividir el dato entre m entidades de forma que es necesario la concurrencia de n de ellas para regenerar ese secreto.

b3) *Almacenamiento, archivo y depósito de la clave privada.* De una parte, es necesario describir los procedimientos definidos para el almacenamiento y archivo de respaldo de la clave privada para que pueda ser recuperada en caso de destrucción o corrupción. Por tanto, es necesario definir si se proporciona, y en qué condiciones, un servicio de recuperación de claves (key escrow) como garantía en caso de pérdida o destrucción (que nada tiene que ver con el tipo de «recuperación» preconizado con la iniciativa *Clipper Chip* que comentamos, peyorativamente, en el Capítulo 6).

A la hora de definir el almacenaje será necesario distinguir si se trata de la clave privada con la que una CA firma los certificados o se trata de las claves privadas de las entidades finales. En el primer caso deberá estar claramente definido el procedimiento riguroso de almacenaje y custodia en dispositivos seguros a los que sólo tengan acceso las personas autorizadas. En el caso de entidades finales, el uso de tarjetas inteligentes aporta un elemento de confianza para este tipo de almacenamientos.

Dentro de las entidades finales, otra distinción importante se deriva del uso previsto para la clave que se ha de archivar o recuperar. Si se trata de una clave privada usada para firmar digitalmente no deben establecerse almacenamientos de respaldo porque **en ningún caso esa clave debe abandonar el dispositivo en el que está guardada** de forma protegida (por ejemplo, una tarjeta inteligente resistente a manipulaciones). Es la única forma de garantizar el No Repudio de la firma (y así lo exigen algunos estándares, por ejemplo, ANSI X.9.57). En este caso, si la clave se pierde accidentalmente no es necesario recuperarla de ningún sitio, sino que lo conveniente es generar nuevamente otro par de claves y solicitar la generación de un nuevo certificado y la revocación del antiguo. Si las firmas se realizan con sellado de tiempo, ninguna clase de conflicto podrá producirse que no sea fácilmente resoluble mediante la comprobación de fechas.

Caso de que la clave privada se utilice para descifrar mensajes cifrados confidencialmente, puede ser conveniente que la organización a la que pertenezca la entidad final tenga establecido algún procedimiento de almacenamiento y recuperación. Esto es así porque si se pierde o destruye accidentalmente esa clave privada, es razonable que se puedan recuperar determinados documentos cifrados con la clave pública pareja de esa clave privada. Algunos de ellos pueden ser antiguos e incluso estar cifrados con claves que en la actualidad ya no están vigentes, pero que es necesario conservar de

forma segura. En estos casos es necesario no destruir esas claves privadas cuando finalice su ciclo de vida y mantener un archivo de respaldo para poderlas recuperar cuando las circunstancias lo impongan (siempre, claro está, bajo unos procedimientos de control rigurosos).

Las estipulaciones que se definan para decidir estas cuestiones han de estar en consonancia con las que se definan para el uso de las claves [ítem a6) comentado más arriba].

b4) *Introducción, activación, desactivación y destrucción de la clave privada.* Es necesario definir los métodos y los procedimientos que se tienen previstos para realizar estas funciones que tanta repercusión pueden tener en el funcionamiento global del sistema.

c) *Otros aspectos de la gestión del par de claves*

Además de todo lo ya dicho para la clave privada puede ser necesario tomar decisiones sobre el periodo de uso de las claves públicas y privadas, así como sobre la forma de almacenar y archivar las claves públicas de los principales agentes telemáticos del sistema.

d) *Datos de activación*

Además de las claves privadas puede haber otra información de seguridad que necesita ser adecuadamente protegida. Tal es el caso de los PINs, contraseñas y otros datos de control de acceso, que, caso de existir, deberán ser adecuadamente generados y protegidos, siguiendo un esquema similar al planteado para la clave privada.

e) *Controles de seguridad en los computadores*

e1) *Requisitos técnicos específicos de seguridad en los computadores*, tales que controles de acceso optativos y obligatorios, etiquetas de control de acceso, mecanismos de identificación y autenticación, auditoría, y otros elementos que avalen la fiabilidad de los sistemas utilizados.

e2) *Criterios de evaluación de seguridad utilizados* (basados, por ejemplo, en TCSEC, ITSEC, etc.) y requisitos establecidos para análisis de sistemas, pruebas, perfiles, certificación y/o acreditación de productos.

f) *Controles de seguridad del ciclo de vida*

f1) *Controles en el desarrollo de los sistemas*: metodología utilizada, modularidad, pruebas, etc.

f2) *Controles de gestión de seguridad*: herramientas y procedimientos utilizados para probar la seguridad de los programas y del hardware utilizado.

f3) *Criterios de evaluación del ciclo de vida* basados en estándares reconocidos para este tipo de tareas.

PERFILES DEL CERTIFICADO Y DE LAS CRLs

Se trata de decidir acerca de la estructura de los certificados y CRLs que van a ser generados en el entorno de certificación al que correspondan las estipulaciones que se estén definiendo. Estas decisiones se refieren a los distintos campos que constituyen el certificado X.509 tal y como lo hemos descrito en los anteriores capítulos.

En concreto, para el *certificado X.509* la RFC 2527 apunta que deben señalarse los siguientes puntos: versiones soportadas, extensiones empleadas y su criticidad, identificadores de objeto de los algoritmos empleados, formato de nombres usados para identificar a las distintas entidades, restricciones de nombre, identificador de objeto de la Política de Certificación, uso de la extensión *restricciones a la política*, sintaxis y semántica de los calificadores de política (*policy qualifiers*) utilizados, y semántica del proceso seguido en el tratamiento de las extensiones críticas.

En lo que se refiere al perfil de la CRL soportado, la norma que estamos analizando recoge que debe puntualizarse: la versión de CRL utilizada (v1 o v2) y, en su caso, el tipo de extensiones específicas empleadas, así como su criticidad.

Es conveniente resaltar que cuando se defina una CPS o una Política de Certificación no es necesario que se tengan que tomar «obligatoriamente» decisiones sobre todos los puntos que hemos descrito en los anteriores apartados. Por ejemplo, en la detalladísima versión 2.1 de la CPS de *VeriSign*, nada se estipula acerca de algunos de estos aspectos. Lo que sí es cierto es que se «estipula que no se estipula», es decir, se incluye el título del ítem y se indica que nada concreto se especifica acerca de él. De esta forma, sabemos que sí los han tenido en cuenta pero que han decidido no pronunciarse sobre ellos. Esta es una de las grandes ventajas que tiene el seguir una norma y un epigrafiado normalizado: ayuda al lector a entender y comparar con mayor facilidad distintos *conjuntos de estipulaciones*.

Sírvanos el anterior comentario a modo de cierre para justificar el que hayamos hecho, a través de los siete últimos apartados, un comentario tan sistemático (aunque muy resumido) de los distintos componentes del *Conjunto de Estipulaciones* propuesto en la RFC 2527, aunque a veces pueda resultar algo aburrido su pormenorizado (y necesario) seguimiento.

10.5. POLÍTICA Y PRÁCTICAS DE SELLADO DE TIEMPO

La utilización de certificados de la clave pública es la viga maestra de cualquier infraestructura de seguridad. Por eso, la política de generación y distribución de certificados invade siempre el funcionamiento de todos los sistemas y servicios que operan bajo una infraestructura de este tipo. No obstante, existen otros servicios de seguridad y otras TTPs que proveen estos servicios. Será necesario, por tanto, especificar para todos y cada uno de ellos las condiciones en que se prestan, consiguiendo con ello que las entidades usuarias tengan un conocimiento ajustado de las garantías y responsabilidades que su uso representa.

Por esta razón, a medida que se vaya extendiendo y popularizando la utilización de otros servicios telemáticos seguros irán apareciendo normas que regulen las condiciones de prestación de esos servicios. Como hemos podido comprobar en los capítulos precedentes, uno de los temas que despiertan mayor interés (desde los puntos de vista jurídico, económico y tecnológico) es el relacionado con la firma digital o con

la firma digital arbitrada (a la que hemos dado en denominar *firma electrónica*, conforme a la nomenclatura impuesta por la correspondiente Directiva de la Unión Europea). Un servicio imprescindible para generar de forma fiable estas firmas electrónicas, como también hemos analizado en el Capítulo 8, es el de generación de sellos de tiempo para ser incluidos en los documentos que se firmen. [Las *Autoridades de Sellado de Tiempo* (TSAs) y los protocolos en que se apoyan los estudiamos ya en el Capítulo 7 al abordar las infraestructuras de seguridad.]

Siguiendo la pauta marcada por la definición de políticas y prácticas de certificación, una manera interesante de abordar este asunto es (según vimos al principio del presente capítulo) mediante la especificación del binomio *política de sellado de tiempo y prácticas de sellado de tiempo*.

Para abordar este asunto nos apoyaremos en un documento surgido en el contexto de los trabajos de ETSI tendentes a la complementación de la firma electrónica. Se trata de un *Internet Draft*⁶ sobre requisitos de seguridad para *Autoridades de Sellado de Tiempo* [PPR03], del que extraeremos parte de las consideraciones que plantea, sobre las cuales realizaremos, de forma muy resumida, algunos comentarios que nos puedan servir para entender el tema de forma adecuada. Aunque la demanda inicial de estos trabajos fuese la de servir de soporte en la provisión de la firma electrónica, los planteamientos que vamos a comentar pueden ser de utilidad para cualquier otra aplicación que necesite probar que determinados datos existían antes de una fecha y hora específica.

DISTINCIÓN ENTRE POLÍTICA Y PRÁCTICAS

Partimos de la base de un servicio de sellado de tiempo proporcionado por una *Autoridad de Sellado de Tiempo* (TSA) en la que confían tanto las entidades que solicitan el sellado de documentos (*subscribers*) como las entidades usuarias (*relying parties*), las cuales, habiendo recibido un documento sellado, dan por válida la fecha y hora allí reflejadas. La TSA es responsable de la generación de los *sellos de tiempo* (*time-stamp tokens*, *TSTs*) y de la adecuada gestión y operación de los sistemas en que se apoya para dar el servicio. Asimismo, la TSA es responsable de la operación de una o varias *Unidades de Sellado de Tiempo* (TSUs) que pueden actuar en representación de la TSA.

Así las cosas, cabe distinguir:

- a) *Política de sellado de tiempo* (*time-stamp policy*).
- b) *Declaración de prácticas de la TSA* (*TSA practice statement*).

En cuanto a la primera, podemos definirla, apoyándonos en los mismos términos con que definimos la *política de certificación* en el epígrafe 10.2, diciendo que la *política de sellado de tiempo* consiste en un *conjunto de reglas*, previamente definidas e identificadas, que indican *la aplicabilidad de un sello de tiempo* a un colectivo particular y/o tipo de aplicación con requisitos de seguridad comunes⁷.

De otra parte, la *Declaración de prácticas de la TSA* consistirá en un conjunto de estipulaciones o cláusulas en las que se recojan los derechos y obligaciones que contraen las entidades que solicitan la emisión de un sellado de tiempo, las TSAs que los generan y las entidades usuarias que dan por válido el sello de tiempo insertado en un documento. Asimismo, estas estipulaciones deberán recoger detalladamente la estructura organizativa, las prácticas operativas y los procedimientos seguidos para implantar la *Política de sellado de tiempo* definida en esa TSA.

La *Declaración de prácticas de la TSA* será un documento que defina cómo se conseguirán los requisitos técnicos, organizacionales y procedimentales identificados en una *Política de sellado de tiempo*⁸. En cierta manera, cabe decir que los usuarios de los servicios de sellado de tiempo pueden definir la *Política de sellado de tiempo* que debe seguir una TSA para adaptarse a sus necesidades, mientras que la *Declaración de prácticas* correspondiente será siempre un documento definido por la organización proveedora de ese servicio.

POLÍTICAS DE SELLADO DE TIEMPO

La Política de sellado de tiempo estará identificada mediante un OBJECT IDENTIFIER que será dado a conocer por la TSA a todas las entidades suscriptoras o entidades usuarias de los sellos de tiempo generados. De esa manera, las entidades implicadas tendrán conocimiento preciso de las condiciones en las que la TSA va a generar los sellos.

La identificación de la política deberá formar parte también del sello de tiempo insertado en el documento sellado. En el apartado «*Protocolo de sellado de tiempo: formato de los mensajes*» del epígrafe 7.9 se recoge la estructura de la información sobre el sello de tiempo, *TSSInfo*. Como se puede comprobar, en esa especificación aparece el campo:

reqPolicy

TSAPolicyId

El tipo *TSAPolicyId* se define como un identificador de objeto donde se describe la política de sellado bajo la que opera la TSA. Es decir, en la definición del sello debe aparecer obligatoriamente esta información acerca del OBJECT IDENTIFIER de la política. Además, también puede comprobarse que este campo aparece de forma optativa en la PDUs que conforma la petición de sellado de tiempo.

Se preconiza que la *precisión* de la fecha y hora en la que el sello fue generado sea de un segundo o mejor. La TSA deberá hacer pública también esta información para que tengan conocimiento de ella tanto las entidades suscriptoras como las entidades usuarias. En principio, se puede partir de la base de que una TSA debe proporcionar una precisión de un segundo. Si la precisión que proporcionan todas las TSUs (*Unidades de Sellado de Tiempo*) dependientes de esa TSA es mejor que un segundo, la TSA puede dar a conocer esta circunstancia en las estipulaciones que publique. Caso contrario, tiene la posibilidad de recoger ese detalle en cada uno de los sellos de tiempo que genere.

Para ello, en la especificación del sello de tiempo a que hacíamos referencia más arriba podemos encontrar los campos siguientes:

**genTime
accuracy**

**GeneralizedTime,
Accuracy**

OPTIONAL

El campo *accuracy* representa la *precisión* del instante de tiempo recogido, es decir, la desviación o la exactitud del campo *genTime* en comparación con UTC, ya que el campo *genTime* representa la fecha y hora a la que el sello de tiempo fue creado y se expresa como UTC (*coordinated universal time*) o «Zulu Time» (según se comentó en el epígrafe 7.9). El campo *accuracy* es optativo porque si en las estipulaciones de la Política de sellado de tiempo está especificado que todos los sellos están generados con la misma precisión, o se especifica una precisión por defecto, no sería necesario incluir ese dato en todos y cada uno de los sellos generados.

Como ya indicábamos, la utilización y fortaleza de los certificados de la clave pública condiciona y sirve de soporte en la generación de sellos de tiempo por parte de la TSA. Así, las entidades que confían en los sellos emitidos por una TSA condicionan su confianza en la validez de esas marcas a la vigencia del certificado X.509 que garantiza la validez de la clave pública pareja de la clave privada con la que esos sellos están firmados. Debido a ello, estas evidencias tienen sólo validez cuando se han generado dentro del periodo de validez del certificado y cuando existen garantías de que el certificado no estaba revocado en ese momento.

La Política definida debe contemplar una serie de obligaciones y responsabilidades de todas las entidades y agentes involucrados en la solicitud, generación y utilización de sellos de tiempo. Además, deberá garantizar que se han implementado los controles necesarios para llevarlas a cabo.

PRÁCTICAS DE SELLADO DE TIEMPO

En [PPR03] se recoge una propuesta acerca de los requisitos que debe satisfacer una Declaración de prácticas de las TSAs que contiene un minucioso y detallado conjunto de elementos sobre los que es necesario que se pronuncien los gestores de una TSA que ofrezca un servicio de sellado de tiempo a un entorno amplio de entidades finales. Estos requisitos los agrupa en cinco secciones, acerca de las cuales haremos a continuación un comentario muy resumido, destacando sólo aquellos asuntos que consideramos más relevantes.

La TSA deberá realizar y publicar una Declaración de prácticas mediante la cual deberá demostrar que ofrece fiabilidad suficiente para proveer de forma segura los servicios de sellado de tiempo y dar a conocer a todas las entidades suscriptoras y potenciales usuarios los términos y condiciones bajo los que se ofrecen sus servicios. Cabe destacar los siguientes aspectos:

a) *Gestión del ciclo de vida de las claves*

- *Generación del par de claves de la TSA.* Es fundamental asegurar la fiabilidad de la clave privada con la que la TSA firma los sellos de tiempo. Por tanto, la generación del par de claves deberá ser llevada a cabo en entornos físicos muy seguros, empleando para ello algoritmos y tamaños de clave reconocidos, y utilizando, además, módulos criptográficos que cumplan con las rigurosas condiciones impuestas en determinadas especificaciones públicas sobre criterios de evaluación de seguridad⁹.
- *Protección de la clave privada de la TSA.* Esta clave deberá estar contenida en módulos seguros protegidos ante manipulaciones (conformes con los criterios de seguridad antes citados) que permitan realizar el cifrado de los sellos sin que la clave salga de su perímetro protegido. Caso de que se realicen copias de respaldo, deberán definirse rigurosísimas medidas de seguridad, tanto técnicas como procedimentales, para garantizar su almacenamiento y evitar su manipulación.
- *Distribución de la clave pública.* La manera de garantizar la autenticidad y la integridad de la clave pública de la TSA es mediante la generación y distribución de un certificado X.509. La CA que emita estos certificados deberá operar bajo una política de certificación que proporcione niveles de seguridad iguales o superiores a los de la Política de sellado de tiempo que se está definiendo.

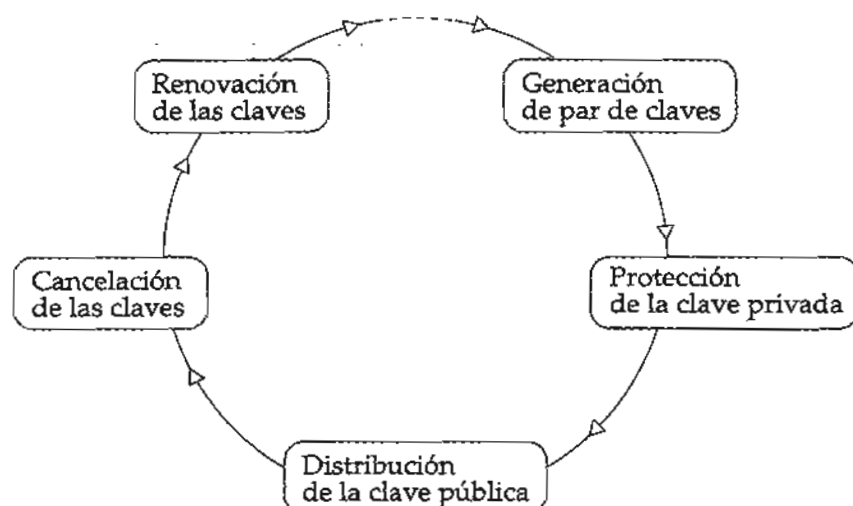


Figura 10.5. Ciclo de vida de las claves de la TSA.

- *Cancelación de las claves.* Dependiendo de los algoritmos utilizados y del tamaño de las claves se determinará el periodo de validez del certificado y, por tanto, del par de claves de la TSA. Además, por motivos relacionados con la pérdida de seguridad de la clave privada puede ser necesario que se dé por concluida su validez. Una vez que finalice el periodo de validez de la clave privada, la política definida garantizará que la TSA no usará esa clave para firmar ningún sello de tiempo. Asimismo, se establecerán procedimientos para destruir esa clave privada y todas las copias que de ella existan.
- *Renovación de las claves.* Deberán definirse procedimientos técnicos y organizativos para garantizar la generación de un nuevo par de claves cuyo ciclo de vida será semejante al seguido por las claves que han dejado de ser efectivas. No obstante, dependiendo del periodo de validez del certificado X.509, en la política que se defina se establecerá un periodo de tiempo (del orden de un año), a partir de la expiración de la validez de la clave de firmado, durante el cual deberán conservarse los registros relacionados con los servicios de sellado de tiempo.

b) Sellos de tiempo

Obviamente, la TSA garantizará que los sellos de tiempo que emita tendrán una estructura conforme a los estándares que comentamos en el Capítulo 7. Además la política que se está definiendo deberá garantizar, conforme a unos procedimientos establecidos, que el reloj del que obtiene la fecha y hora está sincronizado con una fuente UTC autorizada y con una precisión igual o mejor a la comprometida.

c) Operación y gestión de la TSA

Los procesos administrativos y de gestión que la TSA debe seguir constituyen la principal garantía de que el servicio ofrecido cumpla satisfactoriamente con los requisitos técnicos que se le exigen. Para especificar adecuadamente estos procesos es necesario tener en cuenta un abanico muy amplio y muy variado de temas, casi todos ellos similares a los que ya comentamos al describir el *Conjunto de Estipulaciones*

definido en la RFC 2527. Entre los puntos más significativos podemos reseñar los siguientes: a) Gestión de la seguridad; b) Protección de los activos y de la información; c) Roles y cualificación profesional del personal encargado de las tareas técnicas y administrativas; d) Seguridad física del entorno donde opera la TSA; e) Determinación del control de acceso a los equipos y a la información; f) Robustez de los procedimientos seguidos para el desarrollo y mantenimiento de los sistemas que soportan el servicio; g) Compromisos adquiridos para el caso de que la TSA tenga que dejar de prestar servicios; y h) Conformidad del servicio prestado con los requisitos legales existentes en los distintos países y regiones (por ejemplo los marcados por la Directiva sobre Firma Electrónica de la Unión Europea).

Como puede comprobarse, una cosa es definir técnicamente un servicio y unos protocolos (como hicimos en el Capítulo 7 en lo relativo a las TSAs) y otra muy distinta es ofrecer ese servicio hacia entidades finales (a cambio, en muchos casos, de la correspondiente compensación monetaria). Para esto último es necesario definir con detalle una política de seguridad y unas prácticas de comportamientos que traten de llevar al ánimo de los posibles suscriptores y usuarios un adecuado nivel de confianza.

10.6. UNA ESPECIFICACIÓN FORMAL DE LA POLÍTICA DE FIRMA

En el Capítulo 8, dedicado a los escenarios de firma y autenticación, abordamos la propuesta de firma electrónica desarrollada por ETSI con objeto de adaptarse a las especificaciones genéricas planteadas en la Directiva europea 1999/93/CE. Como hemos podido comprobar, la Política de Firma aparece a lo largo de toda la especificación como un elemento de capital importancia. Ya en la definición inicial se dice que la Firma Electrónica es una *«evidencia en formato digital que puede ser procesada para obtener confianza de que determinado compromiso ha sido explícitamente sostenido bajo una Política de Firma, en una fecha y hora concreta, por un signatario bajo un identificador (por ejemplo un nombre o un seudónimo), y opcionalmente un rol»*.

Es razonable que la Política de Firma tenga tanta relevancia porque el firmante siempre adquiere algún tipo de compromiso (personal, contractual o administrativo) cuando firma algún documento, por lo que será necesario que quede definido explícitamente el marco en el que adquiere ese compromiso.

Por otra parte, para que la verificación de la firma dé siempre el mismo resultado, independientemente de la entidad que la evalúe, las condiciones bajo las que el documento fue firmado deben estar perfectamente explicitadas. Cuanto más formal sea la forma de definir esa política, tanto menos ambigüedad se producirá en su interpretación.

ATRIBUTO IDENTIFICADOR DE LA POLÍTICA DE FIRMA

Aunque en la especificación del tipo *SignedData* recogida en la RFC 3369 se refleja que tanto los atributos firmados como los no firmados son opcionales, la especificación de ETSI considera, con buen criterio, que algunos atributos deben estar presentes y firmados, entre ellos el *Identificador de la Política de Firma* (*SignaturePolicyIdentifier*).

En el Capítulo 8 (epígrafes 8.8 y 8.9) ya se comentó la importancia que este atributo tiene en la especificación de la Firma Electrónica. En la Figura 10.6 se representa de nuevo la Figura 8.20 remarcando la posición que ocupa este atributo en dicha

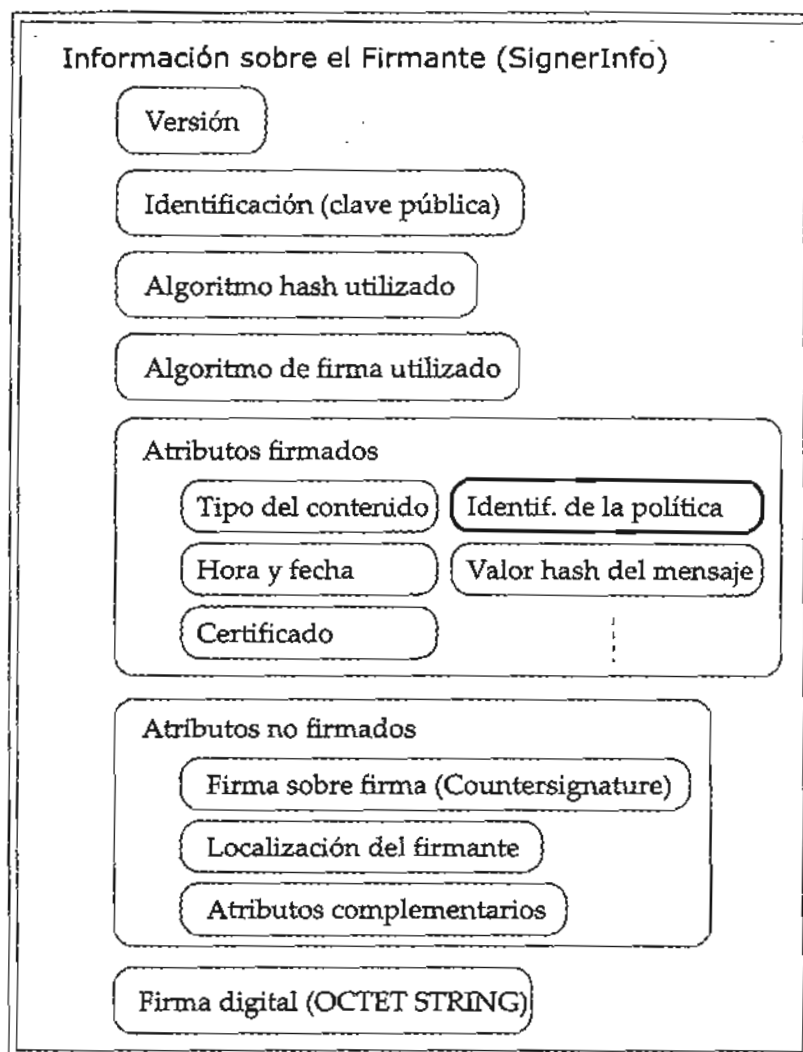


Figura 10.6. Especificación de la Información sobre el Firmante presente en un documento firmado.

especificación. Veremos ahora su definición formal y ampliaremos detalles acerca del significado de los campos que lo constituyen. Su especificación ASN.1 es la siguiente:

```

SignaturePolicyIdentifier ::= CHOICE{
    signaturePolicyId
    signaturePolicyImplied
SignaturePolicyId, SignaturePolicyImplied}

SignaturePolicyId ::= SEQUENCE {
    sigPolicyId
    sigPolicyHashSig
    sigPolicyQualifiers
    sigPolicyQualifierInfo
SigPolicyId, PolicyHash, SEQUENCE SIZE (1..MAX) OF OPTIONAL}

SignaturePolicyImplied ::= NULL
    
```

Como puede comprobarse, la definición deja la puerta abierta para una escapatoria: si en el CHOICE inicial se opta por el campo *SignaturePolicyImplied*, significa que la política de firma queda determinada de forma *implícita* por la semántica de los datos firmados y otras informaciones externas, con lo cual nada se detalla aquí (es de tipo *NULL*).

Aceptando esta posibilidad para los casos en que los compromisos adquiridos no sean muy firmes, pasaremos a comentar la otra opción, que es la que nos interesa para los fines aquí perseguidos. En este caso, el tipo *SignaturePolicyId* está constituido por tres campos:

El primero de ellos, *sigPolicyId*, identifica la especificación concreta de la política de firma bajo la que se opera. Para que esta identificación sea única y concreta, el tipo asociado se define mediante un identificador de objeto:

SigPolicyId ::= OBJECT IDENTIFIER

El segundo campo, *sigPolicyHashSig*, contiene el valor hash de la especificación de esa política y la identificación del algoritmo hash con que se ha realizado ese resumen. De esta forma, desaparece cualquier duda o ambigüedad acerca de cómo estaba definida esa política en el momento de realizarse la firma.

El tercero, *sigPolicyQualifierInfo*, es un campo optativo pensado para recoger información complementaria sobre la política que pueda servir de ayuda a las entidades que necesiten evaluar el documento firmado. Está definido como una secuencia de elementos calificadores con el siguiente formato:

**SigPolicyQualifierInfo ::= SEQUENCE {
 sigPolicyQualifierId SigPolicyQualifierId,
 sigQualifier ANY DEFINED BY sigPolicyQualifierId }**

Es una definición muy abierta, ya que cada uno de los calificadores que se definan están identificados mediante el tipo *SigPolicyQualifierId*, que es un OBJECT IDENTIFIER, y la sintaxis concreta de cada uno dependerá de la especificación asociada a ese identificador de objeto. Por el momento, la especificación de ETSI define sólo dos de ellos, pero deja la puerta abierta a nuevas incorporaciones que en un futuro se consideren convenientes. Estos dos calificadores ya definidos son:

- *spuri*: URI o URL de la web donde se encuentre la definición de la política.
- *spUserNotice*: Un texto o aviso, de unos 200 caracteres, que deberá ser presentado siempre que la firma vaya a ser validada.

(Puede comprobarse el gran parecido formal entre estos campos de la especificación y los correspondientes de la extensión *certificatePolicies* que reproducimos en el epígrafe 10.3.)

Así pues, toda esta información que contiene el atributo formará parte, junto con el documento o mensaje objeto de la firma y los restantes campos que describimos en el epígrafe 8.7, de la información que se presenta a la entidad que reciba un documento firmado (con la firma electrónica), garantizando de esta manera que en el proceso de aceptación o validación de esta firma, la entidad receptora tendrá cumplida noticia acerca de cómo y dónde está definida la Política de Firma bajo la que el signatario firmó el documento.

Esta definición podrá consistir en un documento redactado en lenguaje natural en el que se recojan todas las reglas y condiciones de la política, con lo cual sólo podrá ser entendido cumplidamente por un usuario humano. Otra alternativa más interesante de cara a un futuro consiste en que la Política de Firma esté especificada formalmente usando ASN.1, lo que facilitará que pueda ser interpretada por agentes telemáticos automáticos. En el siguiente apartado se describe de forma resumida la especificación que en este sentido se realiza en la propuesta de ETSI sobre la firma electrónica.

ESTRUCTURA GLOBAL DE LA ESPECIFICACIÓN SOBRE POLÍTICA DE FIRMA

Lo que vamos a presentar a continuación aquí no es la especificación completa de la Política de Firma desarrollada por ETSI, sino algunos de los elementos principales que la constituyen, comentando el alcance e interés de cada uno de ellos. Esta selección y estos comentarios tienen tan sólo un objetivo didáctico (el de ayudar a entender este tipo de planteamientos formales), por lo que el lector interesado en el tema deberá recurrir, como ya hemos apuntado tantas veces, a la definición completa recogida en la norma.

Por otra parte, la especificación formal que presentamos no es, claro está, una definición concreta de una política concreta que pueda ser implantada en un dominio de seguridad concreto. Antes bien, se trata de una especie de «plantilla» que deberán completar y cumplimentar quienes deseen definir, para su propio entorno de comunicación, una política de firma conforme a esta estructura.

Conviene aclarar que en la descripción que sigue la extensión y el nivel de detalle que dedicamos a explicar cada uno de estos elementos no será homogéneo ni «proporcional» al que ocupan en la especificación completa. Antes bien, en cada caso la explicación será más o menos detallada, optando por dedicarle un nuevo apartado a alguno de estos elementos cuando, conforme a los objetivos didácticos que nos hemos marcado, consideremos que ello es de interés. Aclarado esto, digamos que la definición inicial es la siguiente:

```

SignaturePolicy ::= SEQUENCE {
    signPolicyHashAlg           AlgorithmIdentifier,
    signPolicyInfo             SignPolicyInfo,
    signPolicyHash             SignPolicyHash      OPTIONAL }

SignPolicyHash ::= OCTET STRING

SignPolicyInfo ::= SEQUENCE {
    signPolicyIdentifier       SignPolicyId,
    dateOfIssue               GeneralizedTime,
    policyIssuerName          PolicyIssuerName,
    fieldOfApplication        FieldOfApplication,
    signatureValidationPolicy SignatureValidationPolicy,
    signPolExtensions         SignPolExtensions    OPTIONAL
}
    
```

Es decir, la Política de Firma consta de tres campos:

- El identificador del algoritmo *hash* con el que va a ser resumida (recuérdese la formulación del atributo estudiado en el apartado anterior).
- El valor de ese resumen, *signPolicyHash*, que una línea más abajo lo declara del tipo OCTET STRING.
- La información sobre la Política de Firma propiamente dicha, *signPolicyInfo*, cuyo tipo está definido mediante la secuencia de seis campos mostrada más arriba. El significado de estos **seis campos** se describe a continuación:

El primero de los campos que constituyen el tipo *SignPolicyInfo* corresponde al **identificador único** de la política que se esté especificando y es del tipo *SignPolicyId*, cuya gran importancia ya resaltamos en el apartado anterior, donde dejamos constancia de que está definido como un OBJECT IDENTIFIER.

El segundo y tercer campo, *dateOfIssue* y *policyIssuerName*, reservados para reflejar la **fecha de emisión** de la política y la identidad de la **Autoridad emisora** de la Política, son fácilmente inteligibles y los tipos correspondientes que les asigna la especificación (*GeneralizedTime* y *PolicyIssuerName*, respectivamente) los hemos analizado ya en anteriores definiciones.

El cuarto campo, *fieldOfApplication*, está definido como una cadena de caracteres que sirven para definir el **campo de aplicación** de esa política (por ejemplo, el marco legal o contractual), los propósitos específicos en los que se espera que se aplique la firma, etc.

El quinto campo, *signatureValidationPolicy*, conlleva un conjunto de reglas que la política definida impone. Lo analizaremos más pormenorizadamente en el siguiente apartado.

El sexto campo deja sitio para incluir cuantas **extensiones a la Política de Firma** se considere conveniente para añadir nuevas reglas o información complementaria que se considere de interés. La definición del tipo *SignPolExtensions* en que se apoya es:

SignPolExtensions ::= SEQUENCE OF SignPolExtn

SignPolExtn ::= SEQUENCE {
 extnID **OBJECT IDENTIFIER,**
 extnValue **OCTET STRING }**

La definición coincide *casi* por completo con la de las extensiones del certificado recogida en la RFC 3280 *Certificate and CRL Profile* [HFPS02] (compatible con la formulada en la Recomendación X.509) a la que hemos venido haciendo referencia reiteradamente a lo largo de los capítulos anteriores. La única diferencia consiste en que las extensiones a la Política de Firma carecen del campo *critical* que allí sí aparece (por defecto, este campo toma el valor FALSE, con lo cual, si se consideraba que este campo no hacía falta aquí, se podría haber asumido la definición genérica indicando que *nunca* debería hacerse referencia al campo *critical*).

En la especificación que estamos comentando, este campo de extensiones es una puerta abierta a posibles informaciones complementarias, no aportándose ninguna propuesta concreta de estas nuevas extensiones. El campo de extensiones aparece no sólo aquí, en la estructura inicial, sino en muchos otros lugares de la especificación.

REGLAS QUE MARCA LA POLÍTICA

El quinto campo de la estructura inicial analizada en el apartado anterior, *signatureValidationPolicy*, está definido mediante el siguiente tipo:

SignatureValidationPolicy ::= SEQUENCE {
 signingPeriod **SigningPeriod,**
 commonRules **CommonRules,**
 commitmentRules **CommitmentRules,**
 signPolExtensions **SignPolExtensions** **OPTIONAL**
}

Como puede observarse, en esta especificación es muy frecuente usar casi el mismo nombre para el campo que para el tipo (excepto la primera letra que debe ser mayúscula o minúscula para adaptarse a las normas de ASN.1), lo que obliga, en todos

los casos, a «buscar más abajo» para averiguar de qué tipo se trata, no utilizando lo que es costumbre generalizada: cuando se trate de tipos estándar hacerlo constar lo antes posible.

El primer campo, *signingPeriod*, indica la fecha y hora a partir de la cual la política definida puede comenzar a usarse para generar firmas y la fecha y hora a partir de la cual no debe hacerse tal cosa. Se define mediante dos campos del tipo *GeneralizedTime* que ya hemos visto en otras ocasiones.

El cuarto campo está reservado para incluir, optativamente, un conjunto de extensiones a la política tal y como comentamos en el apartado anterior.

Los campos segundo y tercero están dedicados, como su nombre sugiere, a definir un conjunto de reglas, tanto comunes como específicas, que trataremos de analizar en el presente apartado. En la Figura 10.7 se presenta un cuadro-resumen de la estructura de la especificación en un primer nivel de detalle (para facilitar su representación y poder desglosar algunos campos se ha optado por alterar, en algunos casos, el orden con el que estos campos aparecen en la definición).

El campo *commonRules* detalla reglas que son comunes a todos los tipos de compromisos que se adquieren aplicando esta política. El tipo que lo soporta está definido en la especificación de la siguiente manera:

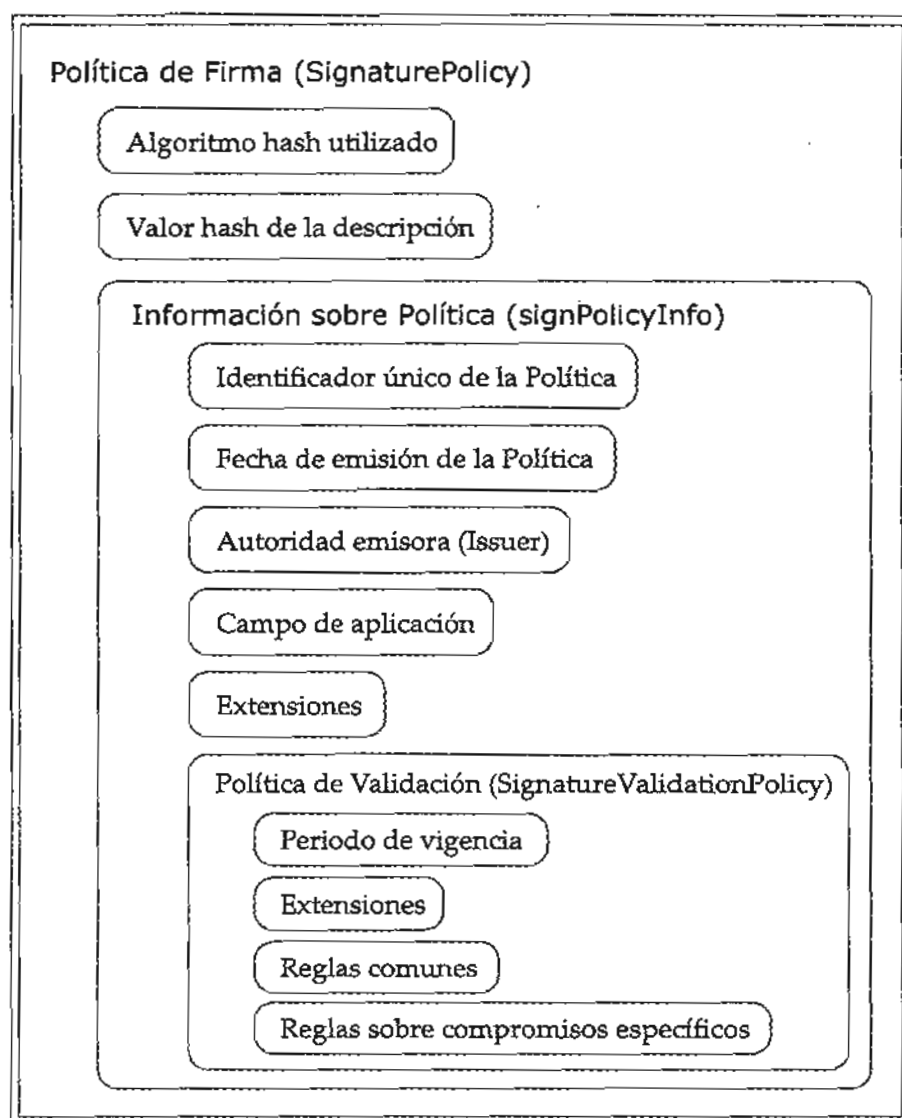


Figura 10.7. Estructura de la especificación sobre Política de Firma.


```

CommonRules ::= SEQUENCE {
signerAndVerifierRules           [0] SignerAndVerifierRules           OPTIONAL,
signingCertTrustCondition       [1] SigningCertTrustCondition       OPTIONAL,
timeStampTrustCondition        [2] TimeStampTrustCondition        OPTIONAL,
attributeTrustCondition        [3] AttributeTrustCondition        OPTIONAL,
algorithmConstraintSet         [4] AlgorithmConstraintSet         OPTIONAL,
signPolExtensions             [5] SignPolExtensions             OPTIONAL
}

```

Es decir, además del posible añadido del campo de extensiones ya comentado, abarca cinco grupos de reglas especializadas sobre distintos temas como son condiciones impuestas a los certificados, los sellos de tiempo, los atributos y los algoritmos usados en la generación y verificación de la firma electrónica.

En cuanto al campo *commitmentRules*, está destinado a facilitar la inclusión de una SEQUENCE de otro tipo, denominado *CommitmentRule* (en singular), con una estructura casi idéntica a la del tipo más arriba mostrado, salvo que posee un campo adicional destinado a identificar el tipo concreto de *compromiso*. Este campo específico se apoya en un tipo definido como:

```

CommitmentType ::= SEQUENCE {
identifier                       CommitmentTypeIdentifier,
fieldOfApplication             [0] FieldOfApplication OPTIONAL,
semantics                      [1] DirectoryString OPTIONAL }

```

Que, como puede observarse, está constituido por tres campos:

- El identificador del compromiso concreto.
- Una explicación textual sobre el campo de aplicación previsto para ese compromiso (por ejemplo, para transferencia de mensajes).
- Una descripción del significado de este compromiso dentro del campo general de aplicación de la presente política.

En resumen, en la especificación de la política aparece un paquete de reglas bajo la denominación de *CommonRules* y una secuencia de tantas *CommitmentRule* como se desee, especializadas cada una de ellas en un *compromiso* concreto e identificado. En la Figura 10.8 se recoge una representación simplificada de esta situación.

Volviendo a la definición del tipo *CommonRules*, las distintas reglas que ahí se detallan, que son las mismas que, como acabamos de decir, aparecen en la definición de cada uno de los *commitment*, tienen que ver con:

1. Reglas para la entidad firmante y la entidad verificadora.
2. Reglas sobre el uso de los certificados y su revocación.
3. Reglas sobre el sellado de tiempo.
4. Condiciones de confianza en los atributos utilizados.
5. Restricciones respecto a los algoritmos utilizados.

Aunque en la definición todos los campos aparecen como optativos, en esta especificación de Política de Firma se obliga a que si algunas de estas reglas (concretamente las tres primeras) no se recogen como *CommonRules* deben estar presentes en todos y cada uno de los grupos de reglas *commitmentRules* que se definan.

A continuación comentaremos brevemente cada una de ellas.

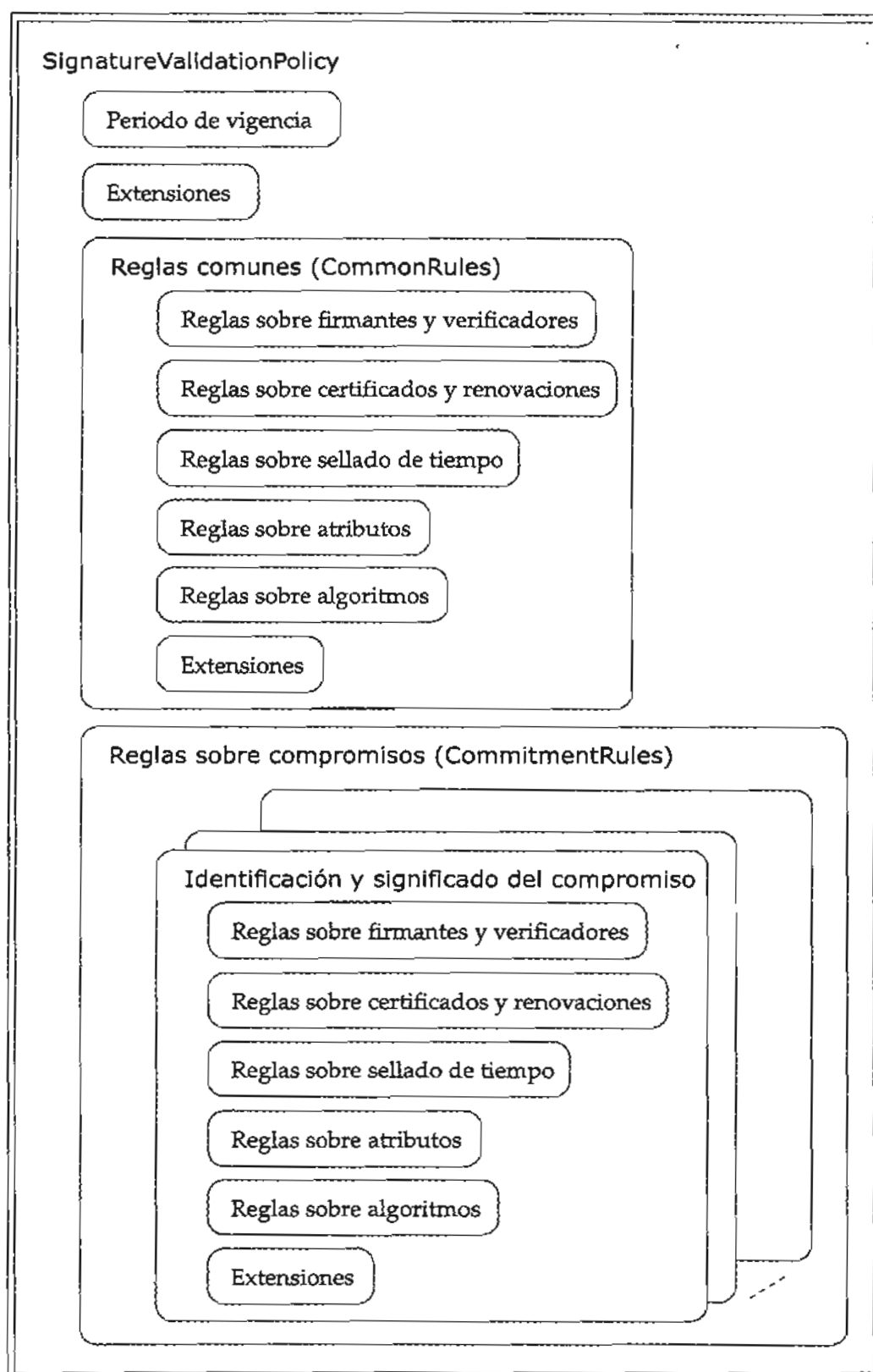


Figura 10.8. Reglas comunes y reglas para compromisos específicos.

REGLAS PARA LA ENTIDAD FIRMANTE Y LA ENTIDAD VERIFICADORA

Como desglose de este campo de la especificación (*signerAndVerifierRules*) se señalan, por separado, las reglas específicas para una y otra de estas entidades. La espe-

cificación de cada una de ellas, en la que hemos suprimido el campo de comentarios para simplificar la explicación, es la siguiente:

```

SignerRules ::= SEQUENCE {
    externalSignedData          BOOLEAN          OPTIONAL,
    mandatedSignedAttr          CMSAttrs,
    mandatedUnsignedAttr        CMSAttrs,
    mandatedCertificateRef       [0] CertRefReq DEFAULT signerOnly,
    mandatedCertificateInfo      [1] CertInfoReq DEFAULT none,
    signPolExtensions            [2] SignPolExtensions      OPTIONAL
}

VerifierRules ::= SEQUENCE {
    mandatedUnsignedAttr        MandatedUnsignedAttr,
    signPolExtensions           SignPolExtensions          OPTIONAL
}

```

Jugando con las combinaciones que ofrece el tipo binario del primer campo de *SignerRules* y con su carácter optativo, se regula si los datos firmados bajo esta política siempre son externos a una estructura CMS (véase epígrafe 8.9), siempre han de formar parte de ella, o son posibles ambas opciones.

El segundo y tercer campo son importantísimos: sirven para determinar con rigor y precisión cuáles son los atributos que, conforme a la política que se está definiendo, deben ser incluidos, en el momento de la firma, como atributos firmados y cuáles como atributos no firmados (véase Figura 10.6). El tipo en que se apoyan estos dos campos, *CMSAttrs*, se define como una secuencia de **OBJECT IDENTIFIER**.

El cuarto campo, *mandatedCertificateRef*, tiene también capital importancia porque sirve para especificar (mediante el tipo que lo define, *CertRefReq*) si la entidad firmante debe incluir solamente su propio certificado (*signerOnly*) o, por el contrario, debe incluir la cadena completa de certificados hasta un punto de confianza. De significado similar es el quinto campo, pero menos significativo porque está definido para incluir una referencia a los certificados, solamente a título informativo, presentes en otro campo de la especificación del mensaje firmado.

En cuanto a la entidad verificadora sólo se define un campo (aparte del destinado a la inclusión de posibles extensiones que aparece por todas partes) que sirve para indicar cuáles son los **OBJECT IDENTIFIER** de los atributos (no firmados, obviamente) que si no los ha incluido la entidad firmante en el momento de generar la firma, deberán ser añadidos por la entidad verificadora a la hora de completarla.

REGLAS SOBRE LOS CERTIFICADOS Y SU REVOCACIÓN

Las reglas de la Política de Firma sobre certificados (*SigningCertTrustCondition*) se concretan en torno a dos grupos, uno sobre los árboles de certificados y otro sobre revocaciones, conforme a la siguiente definición:

```

SigningCertTrustCondition ::= SEQUENCE {
    signerTrustTrees             CertificateTrustTrees,
    signerRevReq                 CertRevReq
}

```



```

        minimum    [0] BaseDistance    DEFAULT 0,
        maximum    [1] BaseDistance    OPTIONAL
    }
    BaseDistance ::= INTEGER (0..MAX)

```

Este campo, que, recordémoslo, puede aparecer tanto en las reglas comunes como en las específicas de uno o varios *compromisos* (*commitments*), es particularmente interesante cuando en la Política de Firma que se está definiendo se identifiquen más de un punto de confianza mediante el campo *trustpoint*. En este escenario, las restricciones de nombre impuestas pueden ser distintas para cada caso, permitiendo mayor rigor en las situaciones más sensibles. Así, como ya comentamos con cierto detalle en el epígrafe 7.6, a partir de la ubicación del punto de confianza se pueden definir concretamente para cada caso tanto los subárboles permitidos como los excluidos, y, dentro de cada uno de ellos, la «profundidad» del espacio de nombres.

La especificación del campo *policyConstraints* se corresponde exactamente con la extensión del mismo nombre que describimos en el epígrafe 10.3. Consecuentemente, también su utilidad y posibilidades son las mismas que las allí comentadas.

En cuanto a las condiciones impuestas en relación con las revocaciones de certificados, el tipo que las delimita, *CertRevReq*, se define como:

```

CertRevReq ::= SEQUENCE {
    endCertRevReq    RevReq,
    caCerts          [0] RevReq }

RevReq ::= SEQUENCE {
    enuRevReq        EnumRevReq,
    exRevReq          SignPolExtensions    OPTIONAL
}

```

donde *endCertRevReq* se refiere a restricciones sobre certificados de entidades finales y *caCerts* a certificados de CAs. Ambos son del mismo tipo, *RevReq*, que es una secuencia de otros dos, el segundo de los cuales (correspondiente a *exRevReq*) es del tipo extensiones, y el primero de ellos es un tipo enumerado que se define como:

```

EnumRevReq ::= ENUMERATED {
    clrCheck        (0),
    ocspsCheck      (1),
    bothCheck       (2),
    eitherCheck     (3),
    noCheck         (4),
    other           (5)
}

```

cuyo significado es el siguiente: a) si el valor es 0, la comprobación de revocación se hará sobre las CRLs señaladas en la especificación de la firma; b) si es 1 se hará usando los servicios OCSP; c) si es 2, la comprobación se hará sobre ambos; d) si es 3 se hará optativamente sobre uno o sobre otro servicio; e) si es 4 significa que, conforme a la política que se está definiendo, no es necesario hacer comprobaciones sobre revocación de certificados; y f) si el valor es 5 significa que la comprobación se hará mediante otros mecanismos indicados mediante las extensiones que se definan en el campo optativo *exRevReq* antes citado.

REGLAS SOBRE EL SELLADO DE TIEMPO

Estas reglas o condiciones (*timeStampTrustCondition*) están relacionadas con la inclusión de sellos de tiempo en las firmas generadas y verificadas conforme a la Política de Firma que se concrete. En la especificación que estamos comentando se definen como:

```
TimestampTrustCondition ::= SEQUENCE {
    ttsCertificateTrustTrees    [0] CertificateTrustTrees    OPTIONAL,
    ttsRevReq                  [1] CertRevReq                OPTIONAL,
    ttsNameConstraints         [2] NameConstraints            OPTIONAL,
    cautionPeriod              [3] DeltaTime                  OPTIONAL,
    signatureTimestampDelay    [4] DeltaTime                  OPTIONAL
}
```

Como puede observarse, todos los campos son optativos. Los tres primeros sirven para poner condiciones específicas a las autoridades de sellado de tiempo, TSAs, que cooperan en la generación de la firma electrónica. Como también puede apreciarse, los tipos de estos tres campos coinciden con los ya analizados en el apartado anterior, por lo que, de no estar presentes, las condiciones que se estableciesen de forma genérica para los certificados, las revocaciones y las restricciones de nombre serían también de aplicación en el entorno del sellado de tiempo.

Los otros dos campos, *cautionPeriod* y *signatureTimestampDelay*, sí son privativos de estas reglas que ponen condiciones a la forma en la que se añade un testigo de tiempo a la firma electrónica. Ambos son del mismo tipo, una secuencia de enteros que permite marcar un periodo («delta») de tiempo en forma de días, horas, minutos e incluso segundos. La definición de este tipo es:

```
DeltaTime ::= SEQUENCE {
    deltaSeconds    INTEGER,
    deltaMinutes    INTEGER,
    deltaHours      INTEGER,
    deltaDays       INTEGER
}
```

El *periodo de cautela* (*cautionPeriod*), como su nombre sugiere, hace referencia al periodo de tiempo que tiene que transcurrir desde el instante que figura en el sello de tiempo antes de que la firma pueda verificarse y darse por válida. La Política de Firma puede imponer esta demora para dar tiempo a que, si se ha producido algún tipo de disfunción o revocación de la clave pública del firmante, esa anomalía pueda ser detectada y debidamente publicada.

Otra cosa bien distinta es la que marca la *demora en el sellado de tiempo*, marcada en la definición de la política mediante el campo *signatureTimestampDelay*, que especifica el máximo periodo de tiempo que puede transcurrir entre la fecha y hora recogida en la firma y el sello de tiempo estampado por la TSA. Es evidente que tiene que existir cierto lapso de tiempo entre el instante en que se crea la firma básica y el reflejado en el sello de tiempo que le devuelve la TSA a la entidad que está generándola. Pero también es razonable que la Política de Firma fije, dependiendo del tipo de compromiso o condicionante, el valor máximo que puede tener ese retraso, porque cuanto mayor sea menos precisión hay en cuanto a la veracidad de la fecha y hora declarada por el firmante, y mayor es el riesgo de que aparezcan revocaciones que invaliden lo firmado.

CONDICIONES PARA LOS ATRIBUTOS Y RESTRICCIONES PARA LOS ALGORITMOS

El campo optativo *attributeTrustCondition*, presente en la especificación de las reglas, tiene como objetivo dejar bien sentado el papel que juegan los atributos que la entidad firmante pueda incluir en la generación de la firma. Como ya se comentó en el Capítulo 8, unas veces esos atributos serán añadidos por la entidad firmante bajo su propia responsabilidad y credibilidad y otras lo hará adjuntando un certificado de atributos emitido por una Autoridad de Atributos, AA (el formato de estos certificados se ha analizado con cierto detalle en el capítulo anterior, aunque allí la atención se fijara en la aportación de información de control de acceso).

Para empezar, según esta especificación de Política de Firma, si este campo optativo no está presente, los posibles certificados de atributos que puedan aparecer no serán tomados en consideración a efectos de validación de la firma. Esta opción parece una medida sensata porque, si ya de por sí la verificación de la firma es algo bastante complejo, el añadirle verificaciones sobre atributos complica todavía más la cosa. Todo esto relativiza la trascendencia de los condicionantes regulados mediante este campo de la especificación.

El tipo en el que se apoya está definido como:

```
AttributeTrustCondition ::= SEQUENCE {
    attributeMandated          BOOLEAN,
    howCertAttribute           HowCertAttribute,
    attrCertificateTrustTrees [0] CertificateTrustTrees  OPTIONAL,
    attrRevReq                 [1] CertRevReq             OPTIONAL,
    attributeConstraints        [2] AttributeConstraints  OPTIONAL
}
```

Si el primer campo, *attributeMandated*, tiene valor TRUE, entonces deberá estar presente al menos un atributo, certificado dentro de las condiciones marcadas mediante los restantes campos. Caso de que el valor sea FALSE, la firma puede ser perfectamente válida aunque no lleve atributos certificados.

El campo *howCertAttribute* sirve para especificar si los atributos incluidos por la entidad firmante son atributos de los que ella se reclama bajo su responsabilidad, o son atributos certificados, o los hay de ambas clases.

Los campos *attrCertificateTrustTrees* y *attrRevReq* son de tipos ya estudiados en apartados anteriores pero referidos a Certificados de Atributos por si existiesen infraestructuras PMI complejas. Ambos son optativos, por lo que pueden estar ausentes en la Política que se esté definiendo.

El último campo de esta serie, *attributeConstraints*, sirve para introducir restricciones respecto a los tipos de atributo que se consideran válidos en la Política de Firma que se defina. Para ello, la especificación permite incluir una secuencia donde se reflejen los tipos permitidos.

Para terminar con la descripción de los cinco grupos de reglas (presentes tanto en las *commonRules* como en las *commitmentRules* específicas de un determinado compromiso) comentaremos brevemente el campo *algorithmConstraintSet*.

Este campo optativo sirve para que desde la Política de Firma se puedan imponer restricciones tanto al conjunto de algoritmos que puedan ser utilizados (de *hashing*, de criptografía de clave pública, de firma digital, etc.) como al tamaño de las claves puestas en juego. Las restricciones pueden imponerse mediante el tipo:

AlgorithmConstraints ::= SEQUENCE OF AlgAndLength

```
AlgAndLength ::= SEQUENCE {  
    algID          OBJECT IDENTIFIER,  
    minKeyLength   INTEGER           OPTIONAL,  
    other          SignPolExtensions OPTIONAL  
}
```

El significado de los distintos campos es fácilmente deducible: se trata de una secuencia de algoritmos, cada uno de ellos referido por un OBJECT IDENTIFIER, para el que se indica optativamente el tamaño mínimo, en bits, que debe tener la clave (además del omnipresente campo de extensiones).

La especificación permite que estas limitaciones puedan establecerse en: a) la entidad firmante; b) CAs emisoras de certificados de entidades finales; c) CAs emisoras de certificados para otras CAs; d) Autoridades de Atributos (AAs); y e) Autoridades de Sellado de Tiempo (TSAs).

A modo de resumen de estos apartados, conviene aclarar que esta especificación de Política de Firma lo que presenta es una estructura básica. Será quien defina, a partir de ella, una política concreta para un dominio de seguridad concreto y quien decidirá qué reglas se impondrán: para ello la especificación ofrece multitud de caminos alternativos.

Del mismo modo, la especificación deja la puerta abierta para que se definan entornos en los que estén presentes múltiples AAs organizadas jerárquicamente y múltiples TSAs. Como ya hemos tenido ocasión de comentar en anteriores capítulos, estas situaciones distan mucho de ser realistas y abordables en un plazo razonable. Antes bien, la existencia de un entorno con múltiples CAs organizadas entre sí representa un problema complejo y difícil de resolver, planteado ya a principios de los años noventa del pasado siglo y todavía pendiente de ser solucionado satisfactoriamente (no habiendo sido implantadas masivamente soluciones de este tipo). Por todo ello, los escenarios que cabe esperar que se implanten, por ahora, tendrán un número reducido y fácilmente gestionable tanto de TSAs como de AAs.

NOTAS AL CAPÍTULO 10

¹ Esta RFC ha sido desarrollada dentro de los trabajos relacionados con la *Internet X.509 Public Key Infrastructure* con la intención de servir de ayuda a las personas que tengan que especificar políticas de certificación o prácticas de certificación para CAs o infraestructuras de clave pública. No se trata, por tanto, de una norma que defina un *perfil* al que tengan que adaptarse todas aquellas implementaciones que pretendan ser *conformes* con dicho estándar. No obstante, esta RFC representa un trabajo muy meritorio que ha tenido una aceptación y un seguimiento muy importante en los ámbitos a los que está dirigida.

² Este tipo de advertencias o condiciones sirven para ser tenidas en cuenta en un debate necesario acerca de los requisitos que es necesario exigir a los ciudadanos de cara a la implantación de la Sociedad de la Información. No se trata tan sólo de instalar sistemas y ponerlos a disposición de la ciudadanía, sino que resulta también necesaria una alfabetización acerca de cómo usarlos y de las repercusiones sociales e incluso legales que su uso puede tener. Todo el mundo entiende que para poder conducir un automóvil los poderes públicos exijan un permiso de circulación, tanto para el vehículo como para el conductor, con objeto de evitar en lo posible que alguien resulte atropellado. De forma similar, para evitar atropellos, sería también necesario analizar tanto las repercusiones que tienen los sistemas telemáticos como las necesidades de formación de sus potenciales usuarios.

³ La empresa *VeriSign* (que es la organización con mayor experiencia e implantación en el mundo de la provisión de certificados) en la versión 2.1 de su CPS acepta la RFC 2527 como base para la redacción de sus estipulaciones y reconoce que el marco definido en esa RFC se ha convertido en un estándar de facto en la industria de las PKIs.

De hecho, la versión 2.1, por adaptarse a la RFC 2527, representa un cambio en relación con versiones anteriores. Así, la versión 1.2 (que data de 1996 y estuvo vigente al menos hasta el 2000) tiene una redacción totalmente distinta. La organización española ACE (Agencia de Certificación Electrónica) redactó también en su día un conjunto propio de estipulaciones para la definición de su CPS. Más recientemente, en lo tocante a prácticas de certificación, ofrece dos redacciones en español: una la CPS de Telefónica y otra la CPS del BBVA (ambas con un *copyright* de *VeriSign* 1996-2000) que son una adaptación al pie de la letra de esa versión 1.2 antes referida y, por tanto, muy parecidas entre sí (al menos en la situación en que se encontraban en septiembre de 2003).

⁴ En la RFC 2527 se deja muy abierta la interpretación de la correspondencia entre el conjunto de estipulaciones y la CPS o política de certificación. Por una parte, mantiene que una política de certificación puede expresarse como un conjunto de estipulaciones, y también afirma que una CPS puede expresarse como un conjunto de estipulaciones con cada uno de sus componentes abordando los requisitos de una o más políticas o, alternativamente, como una colección de conjuntos de estipulaciones, aunque mantiene que una CPS debe refinar la formulación de una política de certificación cumplimentando detalles no incluidos en ésta.

No obstante, *VeriSign*, que, como hemos dicho, acepta la RFC 2527 como un estándar, en su versión 2.1 define su *Certification Practice Statement* siguiendo punto por punto el epigrafiado marcado en esa RFC para el *Set of Provisions*. Además, simultáneamente, esta compañía ha emitido otro documento (presumiblemente de rango superior al de esta versión 2.1, por estar destinado a toda la red mundial de organizaciones afiliadas) que es la versión 1.1 de lo que llama *Certificate Policies*, cuyo epigrafiado también es coincidente con el marcado en la RFC para el *Set of Provisions* y cuya extensión y nivel de detalle es muy similar al de la versión 2.1 de la CPS.

⁵ Para la adaptación al español de la nomenclatura en inglés de los distintos componentes recogidos en la RFC 2527 se ha utilizado la formulada para la redacción (conforme con el epigrafiado de esta RFC) de la CPS de una CA desarrollada en el año 2001 por un equipo de trabajo dirigido por el autor de este texto. El entorno de certificación desarrollado en ese proyecto estaba centrado en un conjunto no muy amplio de usuarios, todos en posesión de una tarjeta inteligente específica, que potencialmente podían realizar transacciones de elevada cuantía en una lonja de mayoristas. No se ha podido contar con el criterio e interpretación de la versión en español de las CPSs publicadas por ACE por no seguir, al menos en su versión vigente hasta septiembre de 2003, el epigrafiado y los componentes marcados en la RFC 2527.

Dada la aceptación creciente de la RFC 2527 como norma de referencia en las infraestructuras de clave pública, sería de bastante utilidad para las poblaciones que se entienden en español disponer de una versión consensuada de esta norma adaptando a nuestra lengua tanto los términos como sus referencias jurídicas.

⁶ Se trata de un documento *Internet Draft* de carácter informativo, titulado *Policy Requirements for Time-Stamping Authorities* [PPR03] publicado en abril de 2003, de cuyo contenido se responsabilizan solamente durante un determinado periodo de tiempo (seis meses). Es técnicamente equivalente a ETSI TS 102 023 v 1.2.1 (2002-06). Aunque ni sobre la claridad ni sobre la precisión del contenido de esta propuesta podemos lanzar las mismas opiniones laudatorias que cabe hacer sobre la otra propuesta que aborda estos mismos extremos de «política» y «prácticas» (la RFC 2527), la hemos utilizado como guión y referencia porque consideramos que es una aportación muy notable y porque, al menos, la conceptualización, el índice de temas y el epigrafiado que sigue pueden ser adecuados para abordar este tipo de especificaciones. Sería muy interesante que se avanzase y se continuase en esta línea de trabajo para que existan documentos normativos de referencia cuya conceptualización e índice de temas sean seguidos por todas las especificaciones sobre *prácticas de sellado de tiempo* que se generen. Habida cuenta que muchas de las secciones de estos documentos deben ser entendidas por personas con una formación técnica muy poco especializada, sería de bastante

utilidad que (al igual que se recomienda en la nota anterior) existiese una versión consensuada de esas futuras normas en español.

⁷ La definición de *time-stamp policy* que se recoge en [PPR03] es: «a named set of rules that indicates the applicability of a time-stamp token to a particular community and/or class of application with common security requirements», que es una adaptación puntual de la no muy brillante definición formulada en la X.509 (y asumida por la RFC 2527 y, consecuentemente, por nosotros mismos) acerca de la *certificate policy* (*política de certificación*). En efecto, esta definición es: «a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements», por lo que la definición de *time-stamp policy* propuesta en [PPR03] lo que hace es quitar donde dice «a certificate» y poner «a time-stamp token».

⁸ La conceptualización, el significado y la relación entre políticas y prácticas de sellado de tiempo aquí formuladas están recogidas de [PPR03]. No obstante, en ese documento se plantea la siguiente definición: «TSA practice statement: statement of the practices that a TSA employs in issuing time-stamp tokens». Que no es mucho definir, ya que, como dice el refrán, para ese viaje no hacían falta alforjas. Dicho sea en descargo de los autores de [PPR03], dicha definición no es sino una adaptación de la definición de CPS formulada por ABA (una asociación de abogados de Estados Unidos) y asumida en la RFC 2527. En efecto, en dicha norma se recoge la siguiente flojísima definición: «Certification Practice Statement (CPS) – A statement of the practices which a certification authority employs in issuing certificates».

⁹ Los estándares de evaluación que se citan son: «FIPS PUB 140-1 (1994) Security Requirements for Cryptographic Modules», «CEN Workshop Agreement 14167-2 Cryptographic Module for CPS Signing Operations – Protection Profile (MCSO-PP)» e «ISO/IEC 15408 (1999) (parts 1 to 3) Information technology – Security Techniques and Evaluation criteria for IT security».

SERVICIOS DE ANONIMATO PARA LA SOCIEDAD DE LA INFORMACIÓN

En anteriores capítulos hemos ido tratando los distintos servicios de seguridad sin entrar a describir las aplicaciones concretas (correo, transferencia de ficheros, servidores de información, etc.) que hacen uso de ellos. El análisis detallado de cada aplicación concreta podrá abordarse apoyándose en el estudio que aquí hemos realizado sobre estos servicios básicos. De forma semejante, en este capítulo se acometerá de forma genérica el estudio de los elementos principales que conforman el servicio de anonimato, haciendo referencia al tipo de aplicaciones que hacen uso de las facilidades que este servicio proporciona.

Consecuentemente, abordaremos en este capítulo la aplicación del anonimato en los medios de pago, en el voto a través de redes telemáticas y, de forma más genérica, en los sistemas que sirven de soporte a la democracia digital. Las aplicaciones emergentes que los soportan están todavía lejos de convertirse en estándares de facto o de jure; por eso abordaremos aquí diversos esquemas genéricos que puedan servir de pauta y comparación para analizar y evaluar las distintas aplicaciones que con el tiempo vayan surgiendo.

Ya en el Capítulo 2 se presentaron las características principales de las tarjetas inteligentes y a lo largo de los distintos capítulos hemos estado haciendo reiterada referencia a sus múltiples aplicaciones y usos. No obstante esto, hemos considerado conveniente hacer una descripción algo más detallada de este tipo de dispositivos en el presente capítulo, ligándolas al estudio del anonimato, debido a que consideramos que las aplicaciones que servirán para el desarrollo de la democracia digital en sus múltiples facetas se apoyarán en gran medida en tarjetas inteligentes, bien sean del mismo tipo que las que están actualmente normalizadas, bien sean resultado del desarrollo de una nueva generación de tarjetas que ofrezcan más posibilidades en sus conexiones externas y mayor capacidad de procesamiento y de memoria.

11.1. CRIPTOGRAFÍA AL SERVICIO DEL ANONIMATO

Ya en el epígrafe 5.6, dentro del apartado «*Otros mecanismos criptográficos*», hicimos referencia a una serie de mecanismos criptográficos que pueden servir de apoyo para conseguir el anonimato en algunas transferencias de información. Solemos denominarlos *mecanismos criptográficos avanzados* porque no se requiere su empleo para la provisión de los servicios de seguridad básicos a los que hemos venido dedicando nuestra atención en los anteriores capítulos.

El hecho de que los utilicemos como soporte de los servicios de anonimato no quiere decir que estos mecanismos sean de uso exclusivo en aplicaciones que proveen este tipo de servicios, sino que pueden también ser utilizados para construir protocolos seguros en otros esquemas de comunicación especiales en donde puedan ser de interés para los fines allí pretendidos. De igual modo, tampoco pretendemos en estos apartados hacer un estudio exhaustivo de todos los mecanismos criptográficos que pueden ser utilizados en la provisión de servicios de anonimato (para lo cual se usan también los mecanismos convencionales antes estudiados), sino que analizaremos solamente aquellos que consideramos más significativos. Fijaremos, por tanto, nuestra atención en tres de ellos:

- *Firma opaca o firma a ciegas.*
- *Secreto dividido.*
- *Secreto compartido.*

FIRMA A CIEGAS SIN TERCERA PARTE

La *firma opaca* o *firma a ciegas* (en inglés, *blind signature*) se caracteriza porque la entidad firmante no puede adquirir conocimiento alguno sobre el documento que está firmando. Posteriormente, la firma obtenida podrá ser verificada como válida por el propio firmante o por cualquier entidad que disponga de la información pertinente para ello, pero el firmante no puede establecer ninguna relación con las circunstancias en que realizó la firma.

En este apartado estudiaremos el escenario más sencillo y al mismo tiempo más robusto de firma opaca: aquel en el que solamente intervienen la entidad *A* peticionaria de la firma y la entidad *B* firmante, sin intervención de tercera parte alguna que haga de árbitro o de juez en ese proceso.

Para adquirir una aproximación intuitiva de lo que representa este mecanismo podemos poner un ejemplo de comunicaciones mediante papel* que puede servirnos de ayuda. Podemos suponer que la relación entre ambas entidades es la siguiente:

1. *A* prepara un documento impreso en un papel con la intención de que *B* lo firme. *A* guarda ese documento junto con un trozo de papel carbón en un sobre que cierra adecuadamente y se lo entrega a *B*. De esta forma «oculta» el contenido del documento. En el sobre está impresa una cuadrícula indicando el sitio exacto en el que *B* debe estampar su firma caligráfica.
2. Si *B* acepta lo que se le propone, firma el sobre, por ejemplo con un bolígrafo, de forma que su firma se estampe, mediante el papel carbón copiativo, en el

* En [Schn96] se propone este esquema para explicar los procedimientos que pueden seguirse en la obtención de dinero anónimo.

documento preparado por A. B ha firmado sin adquirir ningún conocimiento acerca del contenido del documento. B devuelve el sobre firmado.

3. A abre el sobre, retira el trozo de papel carbón y se queda con el documento original firmado por B.

A partir de ese momento A puede presentar ese documento firmado ante cualquier otra persona exhibiendo la firma estampada por B. En realidad, este esquema sólo funcionaría si se aceptase un documento firmado a través de papel carbón (lo cual es mucho suponer), pero como ejemplo nos puede servir.

Para la plasmación de este mecanismo en el ámbito de los procedimientos criptográficos nos apoyaremos en el esquema inventado por Chaum [Cha83] utilizando el algoritmo de cifrado asimétrico RSA. Como hemos dicho, en este proceso intervienen sólo dos entidades:

- a) Entidad peticionaria A que dispone de un mensaje m que quiere que B firme a ciegas.
- b) Entidad firmante B que realizará la firma. La clave pública de B es conocida en el dominio de seguridad donde se lleva a cabo la comunicación:

$$kP_B = (e, n)$$

El procedimiento necesario para que A obtenga la firma opaca, por parte de B, de un mensaje m es el siguiente:

1. A genera un valor aleatorio (k) que denominaremos *factor de opacidad* y que debe estar comprendido entre 1 y n , tal que k y n sean primos entre sí. Debido a ello, este número k posee un inverso, k^{-1} , en el conjunto de números comprendidos entre 1 y n . A continuación, la Entidad peticionaria A oculta (opaca) el mensaje m calculando $x = mk^e \bmod n$ y se lo envía a B:

$$A \rightarrow B: x = mk^e \bmod n$$

2. B, recibiendo x , no es capaz de conocer el valor de m . B cifra el mensaje opacado x con su clave privada, $kS_B = (d, n)$, obteniendo la firma opaca o firma a ciegas:

$$x^d = (mk^e)^d \bmod n = m^d k^{ed} \bmod n = m^d k \bmod n$$

A continuación B le envía a la entidad peticionaria la firma opaca:

$$B \rightarrow A: x^d = m^d k \bmod n$$

3. A «desopaca» el mensaje opacado firmado por B, x^d , dividiendo éste por k (multiplicándolo por su inverso k^{-1}), obteniendo:

$$s = x^d \cdot k^{-1} \bmod n = m^d k k^{-1} \bmod n = m^d \bmod n$$

Es decir, A obtiene:

$$s = m^d \bmod n = B_S(m)$$

que es el mensaje m cifrado con la clave privada de B, es decir, lo que podemos llamar la «firma simple» de B.

A partir de ese momento, la entidad A podrá presentar ese mensaje «firmado» (m más s) ante cualquier instancia, pudiendo demostrar que la firma la ha realizado

la entidad B porque ella y sólo ella es poseedora de la clave privada $kS_B = (d, n)$. Del mismo modo, la entidad B tiene que reconocer su autoría aunque no será capaz de relacionar el documento firmado que se le presenta con ninguna circunstancia de tiempo ni de origen que le permita averiguar cuándo y a petición de quién realizó la firma.

Lo que acabamos de describir es el procedimiento que se seguiría caso de que B acepte firmar algo que a priori no puede conocer. En el escenario de comunicación en el que se esté proporcionando un servicio de anonimato que utilice este mecanismo de firma opaca será necesario establecer las garantías necesarias para que B adquiera la confianza suficiente en que lo que está firmando no perjudica ni su seguridad ni sus derechos. Estas circunstancias serán las que se evalúen cuando se presenten, en epígrafes posteriores, algunos esquemas de voto telemático y de anonimato en los medios de pago.

Conviene darse cuenta de que en este caso la «firma» que obtiene A está realizada sobre el mensaje m completo y no en base al resumen $h = H(m)$ como es el caso del *mecanismo de firma digital RSA o DSA*.

No obstante, si m es muy grande, la entidad A podría pasarle a la entidad B el resumen h del mensaje y obtendría como resultado de desopacar la firma a ciegas:

$$s = B_s(h)$$

que coincide con lo que en los capítulos 5 y 8 hemos denotado como firma f en el mecanismo de firma digital RSA. Es decir, en ese caso A podría presentar el documento firmado (m, f) como si se tratase de una firma sobre resumen bajo algoritmo de firma RSA.

Existen muchas otras propuestas de algoritmos para la obtención y verificación de firma a ciegas, alguno de los cuales comentaremos en el siguiente apartado. El propio Chaum ha desarrollado otros esquemas y ha registrado las correspondientes patentes. En la página web personal de este autor puede encontrarse información detallada sobre ese asunto.

Si nos ceñimos a un esquema de comportamiento equivalente al que hemos descrito antes, podemos generalizar el procedimiento y utilizar una nomenclatura simplificada.

Para indicar que una entidad ha opacado un documento m para B , podremos denotarlo como:

$$O_B(m)$$

Para indicar que una entidad B ha generado una firma a ciegas sobre un documento m opacado, podremos denotarlo como:

$$B_{bsig}[O_B(m)]$$

En realidad, esta nomenclatura que proponemos (el subíndice procede de *blind signature*) es algo redundante, porque si lo que se firma es algo opacado ya se sabe que se trata de una firma a ciegas. Para evitar esa redundancia, podríamos haber decidido reflejar la firma ciega simplemente como $B_{sig}[O_B(m)]$. Pero ello tendría el inconveniente de utilizar una nomenclatura confundente, ya que en anteriores capítulos, cuando representábamos $X_{sig}(m)$ nos estábamos refiriendo a la firma f obtenida por X mediante un *mecanismo de firma digital RSA o DSA* basado en el resumen, $H(m)$, del mensaje. Por todo ello, aun asumiendo la redundancia, emplearemos B_{bsig}

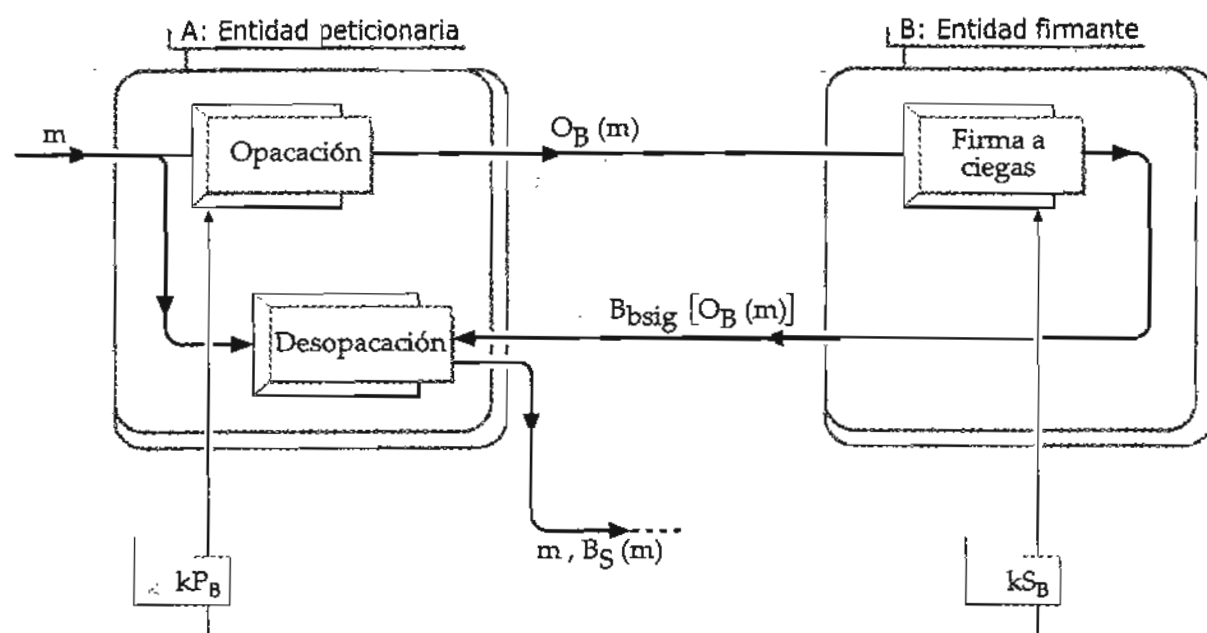


Figura 11.1. Esquema de firma a ciegas.

$[O_B(m)]$ para recalcar que lo que se firma es un documento opacado y que el algoritmo de firma será uno **específico** de firma a ciegas.

En la Figura 11.1 se representa este comportamiento generalizado que podemos describir abreviadamente de la manera siguiente:

1. A oculta (opaca) el mensaje m utilizando la clave pública de B (kP_B) calculando $O_B(m)$ y se lo envía a la entidad B:

$$A \rightarrow B: O_B(m)$$

2. B, con la información recibida, no es capaz de conocer el valor de m . B cifra el mensaje opacado utilizando su clave privada (kS_B) mediante un algoritmo específico de firma a ciegas:

$$B_{bsig} [O_B(m)]$$

A continuación B le envía a la entidad peticionaria la firma opaca:

$$B \rightarrow A: B_{bsig} [O_B(m)]$$

3. A «desopaca» el mensaje opacado firmado por B obteniendo **una firma** del documento m cuyo formato dependerá del algoritmo de firma a ciegas utilizado, pero que, en todo caso, servirá como prueba robusta ante cualquier otra entidad de que B ha firmado el documento m . Si se trata de un esquema equivalente al de Chaum antes estudiado, será $B_S(m)$ el valor de esa firma (situación que se recoge en la Figura 11.1).
4. La entidad A está en condiciones de enviar el mensaje firmado, m , más la firma de m realizada por B, a cualquier otra entidad perteneciente a ese dominio de seguridad (en la figura se ha representado este paso como un flujo de datos que se «sale» del marco representado).

Un procedimiento de firma tal que este podría ser vulnerado utilizando el método denominado «ataque mediante texto elegido» que ya comentamos en el Capítulo 5 al hablar de la fortaleza del RSA. Como consecuencia de ello el atacante podría conse-

guir falsificar la firma o incluso obtener la clave privada de B . Para contrarrestar este riesgo se han diseñado algunas mejoras y añadidos al algoritmo antes descrito, aunque, como ya hemos dicho, se puede usar este mismo procedimiento simplificado con tal de garantizar que en el escenario de comunicación en el que se esté proporcionando un servicio de anonimato se hagan las cosas de tal forma que se garantice que B sólo firme documentos que no perjudiquen su seguridad.

FIRMA A CIEGAS: ALGO MÁS QUE FIRMAR SIN VER

Conviene que fijemos nuestra atención en que la firma a ciegas se realiza por la entidad firmante *a petición de parte*, es decir, esta entidad firma algo que no es capaz de ver y que se le envía para que lo firme. El esquema de comportamiento es, por tanto, totalmente distinto al que se presenta cuando una entidad firma un documento «suyo» con la intención de garantizar su autoría en la generación o distribución de determinada información.

Si pensamos en el comportamiento de una Autoridad de Sellado de Tiempo (Figura 11.2) ante una solicitud que recibe, podemos observar que, según describimos en el epígrafe 7.9, la entidad peticionaria lo que le envía es, además de algunos datos complementarios, el *hash* del mensaje que pretende sellar. La TSA debe generar el sello de tiempo firmando parte de esos datos con el añadido de la fecha y hora en que realiza la operación. El motivo de no mandarle el mensaje m completo es evitar que la TSA pueda andar cotilleando acerca del contenido del mensaje, cosa que no es de su incumbencia, porque su único cometido es garantizar que dicho documento existía antes de determinada fecha y hora.

Es decir, en cierto modo, podemos considerar que una TSA firma «bastante a ciegas» la información que recibe, porque no es capaz de conocer el contenido de m . Sin embargo, la situación es totalmente diferente a la que se recoge en la Figura 11.1, ya que, a posteriori y en caso de conflicto, la TSA puede reconocer con toda claridad los sellos de tiempo por ella emitidos y relacionarlos con la entidad que realizó la correspondiente petición. Eso no tiene nada que ver con la firma a ciegas.

Veamos, para terminar, otra situación ficticia, representada en la Figura 11.3, en la que se imita el procedimiento de firma a ciegas recogido en la Figura 11.1. En este

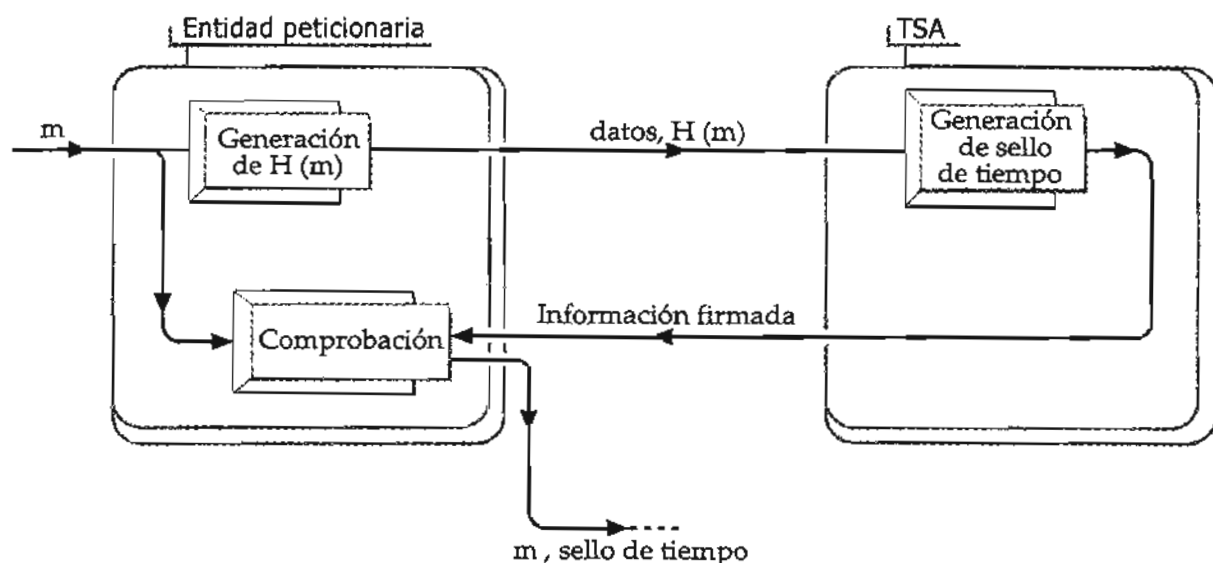


Figura 11.2. Firmado de un sello de tiempo.

caso, lo que la entidad peticionaria envía no es el mensaje opacado sino el resumen $H(m)$ del mensaje. Si la entidad B acepta cifrar $H(m)$ con su clave privada, la entidad peticionaria podría generar y distribuir el mensaje m firmado por B conforme al mecanismo de firma digital RSA. Una situación como esta puede que tenga algún sentido para implementar un esquema local de generación de firmas con dos módulos internos que cooperan entre sí, pero no se corresponde con las características de la firma a ciegas. Veamos por qué:

Si con este esquema estuviésemos tratando de sustituir a un verdadero esquema de firma a ciegas, apreciaríamos que, ciertamente, la entidad B firma $H(m)$ sin poder adquirir ningún conocimiento acerca del mensaje m , pero en otro momento posterior sí que tendría elementos de reconocimiento. En efecto, si en el esquema de la Figura 11.1 la entidad B estuviese firmando a ciegas distintos mensajes que le van enviado de tiempo en tiempo distintas entidades peticionarias, y si además pudiese conocer la identidad de esas entidades (cosa totalmente razonable de cara a que pueda adquirir confianza en la operación que está realizando), podría ir generando un registro donde anotase la entidad peticionaria, la hora y la pieza de información $O_B(m)$ que va firmando. Trabajo inútil, porque, posteriormente, cuando se encuentre los documentos firmados y desopacados, no tendrá pista alguna para relacionarlos con las anotaciones que en ese registro ha ido llevando. Por contra, si un registro similar lo llevase la entidad B en un esquema como el de la Figura 11.3, sí que podría relacionar sin género de duda los $H(m)$ que ha ido firmando con los que aparezcan después en los documentos firmados que A distribuya.

Es decir, para que exista firma opaca o firma a ciegas no basta con que la entidad firmante no sea capaz de ver lo que firma en el momento en que lo firma, sino que la entidad firmante no pueda establecer ninguna relación entre el documento firmado que se distribuya y las circunstancias en que se realizó la firma.

FIRMA A CIEGAS ARBITRADA

El esquema de firma a ciegas estudiado en los precedentes apartados es el que se utiliza con mayor frecuencia en las aplicaciones que, en mayor o menor medida,

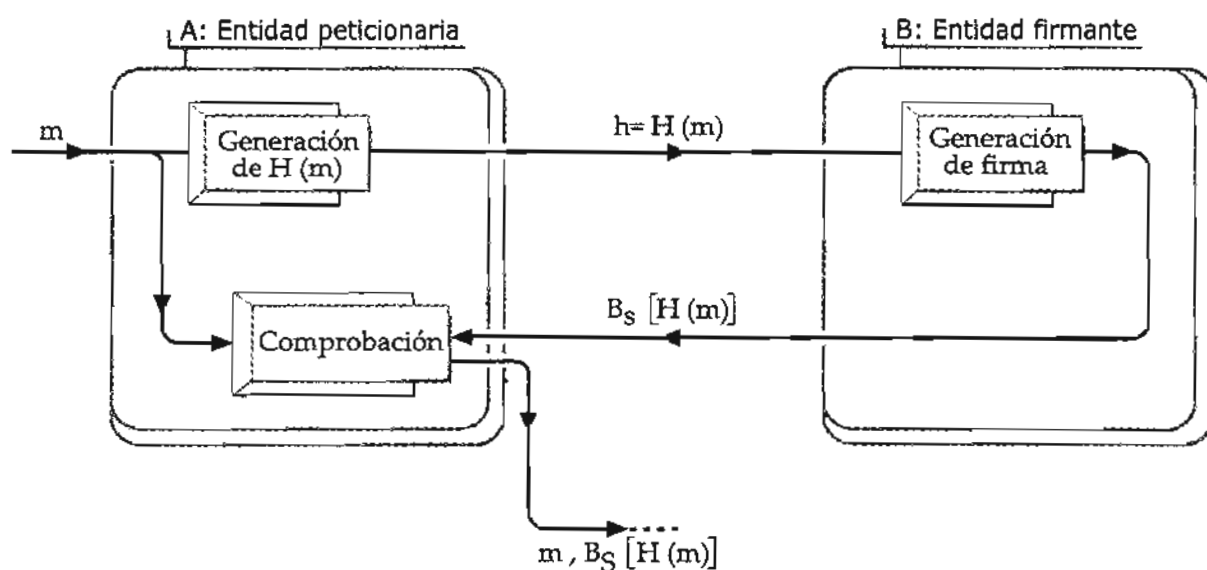


Figura 11.3. Falsa firma a ciegas: firmado del resumen del mensaje.

ofrecen algún servicio de anonimato. No obstante, existen otras propuestas que conviene conocer, siquiera sea de forma genérica, porque puede ser conveniente tenerlas en cuenta en algunos escenarios especiales.

Algunas de estas propuestas incluyen la presencia de una TTP que ayude a la consecución de los objetivos propuestos. En los párrafos que siguen destacaremos dos de estas configuraciones, cada una de las cuales utiliza la TTP con una intencionalidad distinta. El uso de la TTP como ayuda para configurar la firma opaca es una de ellas. La otra consiste en usar la TTP como una especie de juez o árbitro para «romper» lo que puede considerarse excesivo blindaje de la firma a ciegas basada en el esquema de Chaum.

Vayamos por partes y analicemos someramente el primero de esos dos esquemas. En este caso de lo que se trata es de conseguir una firma a ciegas por parte de la entidad firmante aunque para garantizar su validez sea necesario recurrir al dictamen de una TTP en la que confíen los hipotéticos receptores del mensaje firmado. En la Figura 11.4 se representa simplificada una posible plasmación de este esquema. El hecho de contar con la TTP, que comparte secretos tanto con la entidad peticionaria como con la entidad firmante, permite emplear algoritmos de generación de firmas menos complejos¹ que los puestos en juego en el esquema con solo dos entidades que analizamos en apartados anteriores.

La fortaleza de los algoritmos utilizados debe ir en consonancia, tanto en este como en otros casos, con el riesgo que represente la vulneración de la protección que ofrecen. No será lo mismo garantizar rotundamente que el voto sea secreto (que no se sepa quién ha emitido determinado voto) que, por ejemplo, mantener el anonimato de un ciudadano que accede a la web de una administración pública para manifestar, de forma anónima, su opinión o su queja sobre tal o cual asunto. En el primer caso, la ruptura del secreto supondría la invalidación total del proceso de votación que se esté realizando, mientras que en el segundo no parece razonablemente probable esperar

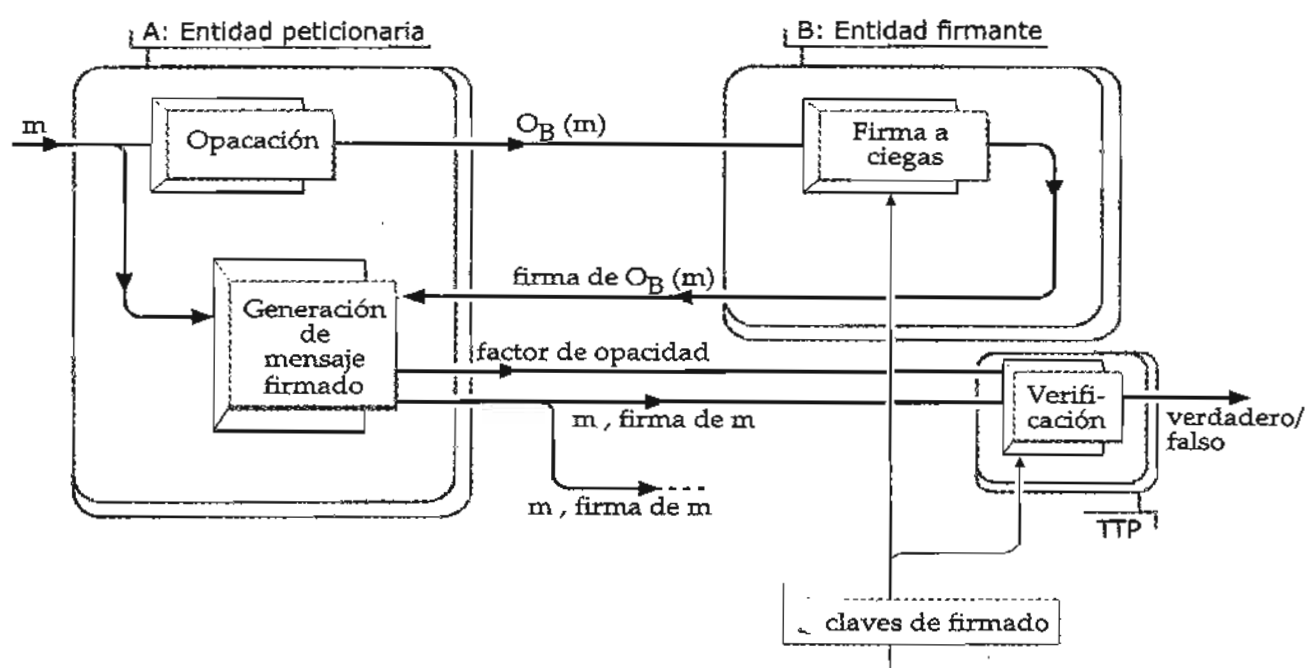


Figura 11.4. Generación y verificación de firma a ciegas con ayuda de una TTP.

que haya alguien interesado en poner en juego la suficiente energía como para conseguir romper el anonimato del opinante. Consecuentemente, si estuviésemos utilizando en ambos casos un mecanismo de firmas opacas, los algoritmos que sería necesario utilizar en cada uno de ellos no tendrían por qué ser de la misma fortaleza.

Volviendo al modelo propuesto en la Figura 11.4, vamos a suponer que en este esquema las claves secretas que usa la entidad B para generar la firma a ciegas son también conocidas por la TTP que va a realizar posteriormente la verificación de esa firma.

El proceso podemos describirlo como sigue:

- a) La entidad A peticionaria, utilizando un factor de opacidad, opaca el mensaje m y se lo envía a la entidad firmante B . En la figura se ha representado este mensaje opacado como $O_B(m)$ para cubrir el caso en que la opacación sea distinta en función de la entidad que se desea que firme a ciegas (como ocurre en el esquema de Chaum). Caso de que (como ocurre en otros algoritmos) la opacación no fuese específica de la identidad de la entidad firmante, el mensaje opacado se podría representar simplemente como $O(m)$.
- b) La entidad firmante B genera la firma opaca de $O_B(m)$ y se la envía a la entidad A .
- c) A realiza las comprobaciones que sea capaz de hacer (no está capacitada para verificar la firma que ha generado B) y:
 1. Envía a la TTP verificadora el factor de opacidad utilizado en el paso a).
 2. Genera el mensaje firmado: m más la firma de m realizada por B . Esta firma debe ser una pieza de información calculada a partir del valor de la firma que ha recibido de B (para evitar que esta entidad pueda posteriormente reconocerla).
- d) La entidad A envía ese mensaje firmado a cualquier otra entidad destinataria o, antes de distribuirlo, a la TTP para que verifique la corrección de la firma.
- e) Cualquier entidad que reciba el mensaje firmado (m más la firma a ciegas) si quiere verificar la validez de la firma tendrá que dirigirse a la TTP verificadora para conocer su veredicto sobre la validez o falsedad de la firma realizada por B , ya que B no puede realizar ese reconocimiento.

La gracia del asunto está en que la TTP conoce tanto el factor de opacidad generado por A como las claves utilizadas por B para la generación de la firma. Es evidente, por tanto, que la TTP debe ser de absoluta confianza para garantizar el funcionamiento de la firma a ciegas, ya que, si quiere, en cualquier momento puede desvelar los secretos bajo los que se sustenta.

Este esquema de funcionamiento puede extenderse a múltiples entidades peticionarias y varias entidades firmantes. En este caso, para garantizar la verificabilidad de la firma sería necesario incluir algunos identificadores en los intercambios de información descritos en los pasos anteriores.

El segundo de los dos esquemas que anunciábamos al principio de este apartado no se basa en incorporar una TTP para facilitar las operaciones sino en incluirla dentro del escenario para que actúe como juez o árbitro para deshacer la opacidad de la operación en caso de que se sospeche que alguna de las partes involucradas en el proceso actúa maliciosamente para obtener un provecho ilícito de la utilización del mensaje firmado a ciegas.

En la Figura 11.5 se representa un esquema en el que están presentes la entidad solicitante (se supone que pueden existir varias entidades de este tipo), la entidad

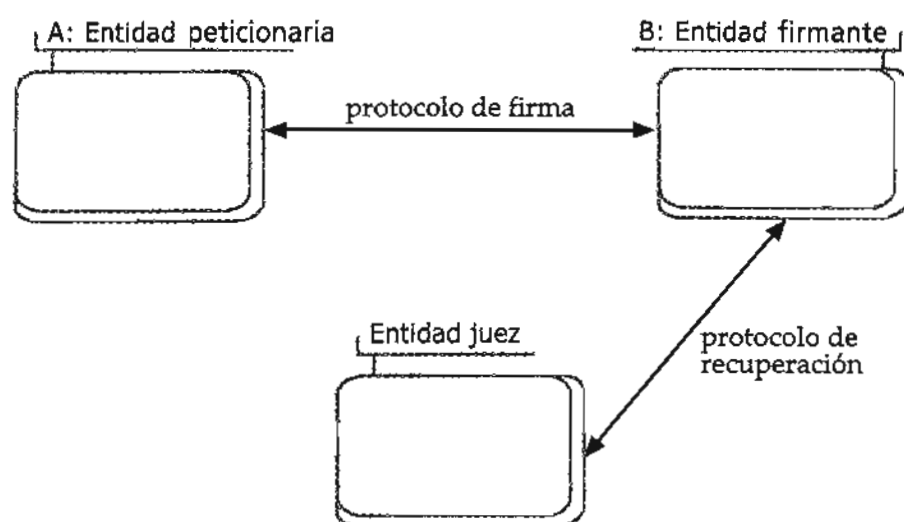


Figura 11.5. Esquema de firma a ciegas arbitrada por una entidad juez.

firmante y una entidad de confianza que actúa como juez. Entre estas entidades se establecen dos protocolos que pueden actuar de forma coordinada:

- Un protocolo *de firma* en el que sólo intervienen la entidad peticionaria y la entidad firmante.
- Un protocolo de *recuperación del vínculo* que relaciona a la entidad firmante con la entidad que actúa como juez.

Mediante el primero de ellos la entidad peticionaria obtiene la firma opaca robusta de un mensaje con las características que ya resaltamos al estudiar la firma a ciegas sin tercera parte. Es decir, de tal manera que la entidad firmante no puede relacionar la información que obtiene al ejecutar el protocolo con la pareja mensaje-firma que la entidad peticionaria distribuirá posteriormente.

Mediante la ejecución del protocolo de *recuperación del vínculo* la entidad firmante puede recibir información del juez que le permita relacionar la información que obtuvo al ejecutar el protocolo de firma con la pareja mensaje-firma producida por la entidad peticionaria. Para ello, durante la ejecución del protocolo de firma, la entidad peticionaria debe enviar cierta información que sólo el juez puede leer y la entidad firmante debe ir pasándole a la entidad juez la información suficiente para que ésta esté en condiciones posteriormente de ayudarle a recuperar el vínculo antes referido. Se supone que esta última revelación de datos sólo se pone en marcha cuando existen sospechas justificadas en relación con la utilización indebida de la opacación de la firma (en un entorno real debería ponerse en marcha mediante autorización judicial).

En [SPC95] se recoge un escenario de este tipo en el que se define un modelo genérico similar al representado en la Figura 11.5 que acabamos de comentar. En esta propuesta² se recogen, además de varias realizaciones concretas de estos protocolos, dos esquemas distintos para la obtención de la firma opaca que se distinguen entre sí dependiendo de la información que la entidad juez recibe de la entidad firmante durante la ejecución del protocolo de *recuperación del vínculo*.

Como puede apreciarse, la TTP que opera en el esquema representado en la Figura 11.4 participa de oficio en el procedimiento de verificación de la firma a ciegas, mientras que la TTP presente en el esquema mostrado en 11.5 no es de por sí imprescindible para la creación y verificación de la firma. La funcionalidad de esta TTP consiste en observar el proceso y adquirir información sobre él para, llegado el caso

de un comportamiento presumiblemente indebido, desvelar lo que la firma a ciegas oculta. Es evidente, no obstante, que la TTP del primer esquema también puede actuar, si las circunstancias lo requieren, bajo esta segunda funcionalidad.

SECRETO DIVIDIDO

A veces resulta interesante mantener una información secreta repartida entre varias entidades comunicantes con objeto de asegurar que la recomposición del secreto se realiza solamente si coinciden en la voluntad de reconstruirlo todas las partes entre las que aquél se había dividido.

Un ejemplo no telemático de una situación de este tipo podría ser el caso de la caja fuerte de un banco que para abrirla se requiriese la presencia de tres empleados, cada uno provisto de una llave distinta, que previamente le ha sido confiada para su custodia. Para poderla abrir hace falta la presencia de los tres y la voluntad colectiva de aportar el componente que cada uno posee. En este caso, lo que es definitivo no es la identidad de los empleados sino la posesión de la correspondiente llave.

Para emular lo que ocurre en las votaciones convencionales mediante papeleta, en un entorno de voto electrónico podríamos pensar en que para abrir la urna electrónica podría ser necesario que coincidiesen cinco personas, cada una de las cuales tuviese almacenada una información en un testigo de seguridad (por ejemplo una tarjeta inteligente), de tal manera que la urna estuviese programada para que sólo se abriese (mostrase su contenido) cuando esas cinco personas hubiesen aportado su información secreta (hubiesen introducido su tarjeta inteligente en un lector) y, como resultado de la adición de todas ellas, se reconstruyese un determinado secreto global.

El paso previo debería haber sido, por tanto, la división de un secreto (*secret splitting*) en tantas partes como individuos se quiere que participen en su reconstrucción. Cada uno de los fragmentos de por sí no tienen ningún valor: el secreto se reconstruye cuando se juntan todos ellos. Además el secreto global puede reconstruirse sin que ninguno exhiba ante los demás el fragmento que le corresponde. Conveniremos en denominar *secreto dividido* al caso en el que la partición del secreto se haga de forma que para reconstruirlo sean necesarias todas y cada una de las partes.

Un esquema sencillo [Schn96] para conseguirlo (ya planteado en los inicios de la criptografía moderna) consiste en aplicar un algoritmo basado en la utilización de la operación *OR exclusivo*, que tiene la propiedad de que dadas dos cadenas binarias M y N del mismo número de bits y una cadena 0 con todos sus bits a cero, se cumple que:

$$\begin{aligned}M \oplus M &= 0 \\M \oplus 0 &= M \\(M \oplus N) \oplus N &= M\end{aligned}$$

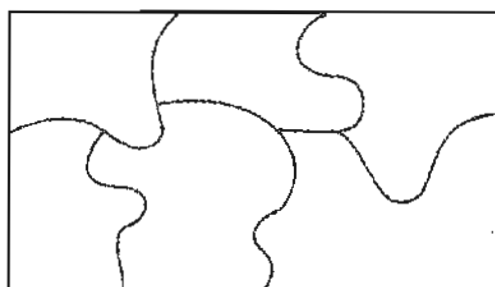


Figura 11.6. Secreto dividido. Para reconstruirlo son necesarias todas las partes.

Supongamos que existe un secreto S , por ejemplo la clave privada de la urna electrónica mediante la cual pueden leerse los datos contenidos en ella. Consideremos que el secreto S puede estar representado por una cadena de bits (1.024, por ejemplo). Si se tratase de dividirlo en solamente dos partes, un procedimiento sencillo para conseguirlo podría ser que la Autoridad de Seguridad que vaya a generar la partición del secreto S realice lo siguiente:

- a) La Autoridad de Seguridad genera de forma aleatoria una cadena de binaria, $S1$, del mismo número de bits que el secreto S .
- b) A continuación, la Autoridad de Seguridad, a partir de S y de $S1$ genera una nueva cadena aleatoria mediante la operación *OR exclusivo*:

$$S2 = S1 \oplus S$$

- c) La Autoridad de Seguridad reparte las cadenas $S1$ y $S2$ entre dos entidades distintas, A y B , que serán las que custodien la parte del secreto que se les ha confiado.

En cualquier momento posterior, si A y B se ponen de acuerdo y se realiza la operación *OR exclusivo* entre sus partes del secreto, se obtiene:

$$S1 \oplus S2 = S1 \oplus (S1 \oplus S) = S$$

A pesar de su sencillez, la operación *OR exclusivo* es muy robusta (ya vimos su empleo en los criptosistemas de secreto perfecto), de forma que, si las cadenas se han generado de forma bastante aleatoria, será bastante difícil romper un esquema de comportamiento como el antes descrito.

Conviene percatarse de que para recomponer el secreto cada entidad aporta su información al sistema donde se está realizando la operación de suma sin que ello implique que la otra entidad adquiera conocimiento acerca del valor de esa parte del secreto (ni del secreto global, por supuesto). Es decir, se puede reconstruir el secreto sin que ninguna de las entidades participantes averigüe la información secreta que posee la otra.

Este esquema de comportamiento puede extenderse sin ninguna dificultad a un mayor número de partes. Si, como en el ejemplo de la urna electrónica que vimos antes, son cinco las partes en las que hay que dividir el secreto, la forma de proceder sería la siguiente:

- a) La Autoridad de Seguridad genera de forma aleatoria **cuatro** cadenas de binarias ($S1$, $S2$, $S3$ y $S4$) del mismo número de bits que el secreto S .
- b) A continuación, la Autoridad de Seguridad, a partir de S y de las cuatro cadenas $S1$ a $S4$ genera una nueva cadena aleatoria mediante la operación *OR exclusivo*:

$$S5 = S1 \oplus S2 \oplus S3 \oplus S4 \oplus S$$

- c) La Autoridad de Seguridad reparte las cadenas $S1$ a $S5$ entre cinco entidades distintas.
- d) En cualquier momento posterior, si las cinco entidades depositarias de una parte del secreto se ponen de acuerdo y se realiza la operación *OR exclusivo* entre esas cinco cadenas, se obtiene:

$$S = S1 \oplus S2 \oplus S3 \oplus S4 \oplus S5$$

Todo depende de la Autoridad de Seguridad que es, en definitiva, la responsable del correcto funcionamiento del sistema. Las entidades depositarias del secreto dividido no tienen forma alguna de comprobar la corrección de la cadena que se les está entregando. Si la Autoridad de Seguridad, a la hora de generar y distribuir las cadenas, actúa maliciosamente, las entidades implicadas no pueden percatarse de ello hasta el momento en el que quieran reconstruir el secreto y no sean capaces. Y lo que es peor: pueden pensar que la culpa de ese fallo es de una de las entidades participantes que ha podido cambiar el valor de la cadena que le han confiado.

Es evidente, por tanto, que este tipo de solución sólo sirve en el caso de que todos los participantes confíen de forma absoluta en la Autoridad de Seguridad y que las partes del secreto estén almacenadas en un testigo de seguridad no manipulable.

El inconveniente grave de un esquema como el que acabamos de ver es que para reconstruir el secreto es necesaria la concurrencia de todas y cada una de las partes en las que se ha dividido. Si, en el ejemplo de la urna electrónica, una de las cinco personas necesarias para iniciar el proceso de apertura de la urna se ausenta del sitio en el que esa concurrencia debe tener lugar, bien porque las encuestas que se han realizado a pie de urna la hayan sumido en una fuerte desmoralización, bien porque la jornada de confraternización interpartidaria le haya resultado insufrible e insoponible, todo el proceso final de recuento se vería bloqueado. La solución que se plantea en el apartado siguiente trata de resolver esta dificultad.

SECRETO COMPARTIDO

Un planteamiento más general (del que el caso anterior sería un caso particular) consiste en compartir o repartir el secreto (*secret sharing*) entre varias partes de forma que para reconstruirlo no sea imprescindible reunir todas las porciones en que ha sido dividido sino solamente un número mínimo de ellas. Un esquema de este tipo de repartición del secreto es el que podemos denominar esquema umbral p sobre q , o esquema (p, q) , donde tanto p como q son enteros mayores que uno y p es menor o igual que q . En este esquema, a partir del secreto S se construyen q muestras o «sombras» (*shadows*) que se reparten entre distintas entidades, de tal suerte que se puede reconstruir el secreto reuniendo cualquier combinación de p de ellas.

En la Figura 11.7 se representa un ejemplo de esquema umbral $(4,7)$, en el que se han extraído siete sombras o muestras del secreto y para reconstruirlo solamente se necesitan cuatro cualesquiera de ellas. Es decir, tres muestras serían insuficientes para reconstruir el secreto, cualquier combinación de cuatro de ellas sería suficiente, y si se reúnen cinco o seis también puede reconstruirse, aunque sobradamente.

Existen distintas propuestas de algoritmos para plasmar un esquema de este tipo, casi todas ellas bastante antiguas y relativamente sencillas de entender. El algoritmo que aquí explicaremos es el propuesto por Shamir [Sham79] basado en la interpolación de polinomios definidos en un campo finito (recuérdese lo visto en el Capítulo 3). El esquema umbral (p, q) se definiría de la siguiente forma:

A partir del secreto S , se elige un número primo n (de tal suerte que tanto p y q como el secreto S sean números inferiores a él) y se construye un polinomio de grado $p-1$ tal que:

$$y = f(x) = (S + a_1 x + a_2 x^2 + \dots + a_{p-1} x^{p-1}) \bmod n$$

Los coeficientes $a_1, a_2, \dots, a_{p-1}$ se eligen de forma aleatoria de entre números también menores que n . El secreto S es el valor que adquiere y para $x = 0$ y las distintas

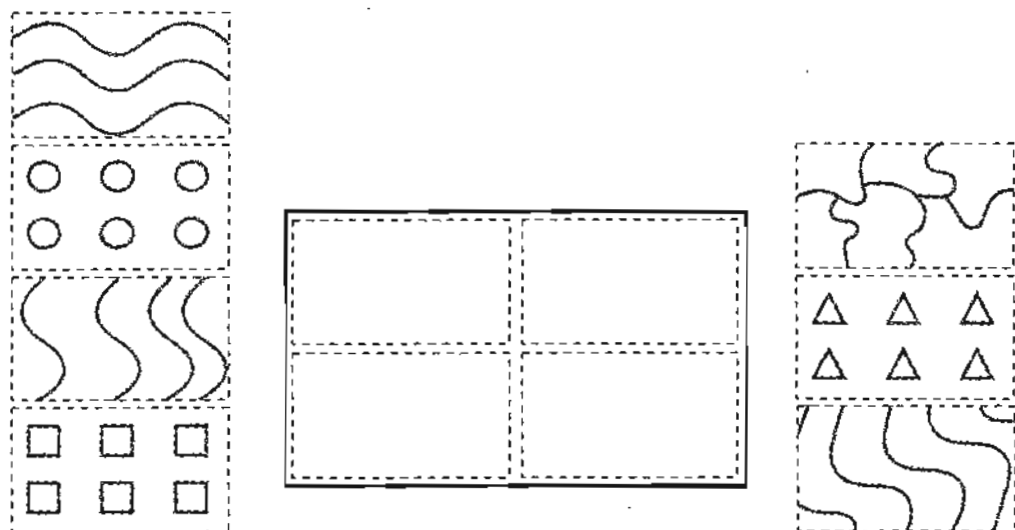


Figura 11.7. Secreto compartido: para reconstruirlo son necesarias cuatro de las siete muestras.

muestras o sombras se obtendrían dando valores arbitrarios, distintos de cero, a la variable x . Cada muestra estará constituida por uno de esos pares (x_p, y_p) que se hayan formado. Pueden construirse tantas q muestras como se deseen y es fácil demostrar que reuniendo solamente p de ellas puede calcularse el secreto S .

Para entenderlo fácilmente, veamos un ejemplo muy sencillo en el que p es igual a dos y el número n que acota el campo es muy pequeño. Supongamos que el polinomio es:

$$y = (9 + 4x) \bmod 11$$

Es decir, el secreto S que se quiere repartir es 9 y 4 es el coeficiente del término lineal que se ha elegido al azar. Si representamos $y = 9 + 4x$ en dos ejes de coordenadas, saldría una línea recta que quedaría perfectamente determinada por dos puntos. Por tanto, conociendo solamente dos puntos de esa recta se conocería el punto en el que corta al eje y , esto es, el valor del secreto S . Aunque la representación en el plano de la gráfica resultante teniendo en cuenta que los valores han de calcularse módulo 11 no sería una sola recta, las conclusiones respecto a la determinación del secreto son las mismas: como $p = 2$, son necesarias solamente dos muestras para reconstruir el secreto. El número de muestras, q , que pueden obtenerse sólo está limitado por el valor del módulo.

Por ejemplo, quien repartiese el secreto podría generar tres muestras que se podrían distribuir entre otras tantas personas:

- $(x = 1, y = 2)$, dado que $y = (9 + 4) \bmod 11 = 2$.
- $(x = 3, y = 10)$, dado que $y = (9 + 12) \bmod 11 = 10$.
- $(x = 4, y = 3)$, dado que $y = (9 + 16) \bmod 11 = 3$.

Es inmediato comprobar que a partir del polinomio $y = (S + a_1 x) \bmod 11$, con solamente dos de esas tres muestras o sombras se podría generar un sencillo sistema de dos ecuaciones con dos incógnitas del que se deduciría el valor $S = 9$.

Obviamente, para que sea muy difícil averiguar S sin contar con al menos dos muestras sería necesario disponer de números muchísimo más elevados que los que se han supuesto en el ejemplo, cuya única virtud es facilitar el seguimiento de las ope-

raciones mostradas. Por otra parte, el grado $p - 1$ del polinomio dependerá del número mínimo de muestras necesarias para reconstruir el secreto.

Volvamos al ejemplo de la urna electrónica planteado en el apartado anterior. La urna sólo se abre (muestra el contenido de las «papeletas» de votación) cuando los encargados de abrirla son capaces de reconstruir el secreto S , que en este caso está representado por los 1.024 bits de su clave privada. En el apartado anterior utilizábamos para ello el secreto S dividido en cinco partes, de tal manera que para ser capaces de reconstruirlo sería necesaria la presencia de todas y cada una de las partes. Y vimos también el problema que puede acarrear la exigencia de esta comparecencia unánime.

Utilizando el paradigma del secreto compartido podríamos desarrollar un esquema umbral (4,5), es decir, la Autoridad de Seguridad generaría cinco muestras o sombras y se las entregaría a cada una de las personas miembros de la Mesa Electoral encargada de abrir la urna: un Presidente o Presidenta, un Secretario o Secretaria y tres vocales. Ahora no es imprescindible la presencia de los cinco: con que sólo cuatro de ellos aporten su muestra del secreto, la urna podrá ser abierta.

El número mínimo de muestras necesarias, $p = 4$, determina el orden del polinomio, que en este caso será:

$$y = f(x) = (S + a_1 x + a_2 x^2 + a_3 x^3) \bmod n$$

donde S es el secreto y los coeficientes a_1 , a_2 y a_3 serán elegidos al azar entre números menores que n . A partir de aquí, para obtener las cinco muestras no queda sino dar cinco valores arbitrarios a x (distintos de cero, que se corresponde con el valor asignado a S) y obtener los correspondientes cinco pares (x_i, y_i) . En este caso, aunque uno de los miembros de la mesa se niegue a colaborar, la apertura se podrá realizar con todas las garantías.

Si bien, como hemos dicho, p determina el grado del polinomio, el número q de muestras a repartir es ampliable sin modificar el polinomio de partida. Si el esquema anterior lo quisiésemos convertir en un esquema umbral (4,7), como el representado en la Figura 11.7, sólo tendríamos que generar dos pares (x_i, y_i) más para tener las siete muestras necesarias. Llegados a este punto, la Autoridad de Seguridad podría decidir distribuir las muestras de forma que le entregase dos muestras a la persona que ostente la presidencia, otras dos muestras a quien lleve la secretaría y una muestra a cada uno de los vocales restantes.

Bajo una situación como esta, para reunir cuatro muestras y ser capaces de abrir la urna, podrían ser suficientes el presidente y dos vocales, o el secretario y dos vocales, o el presidente y el secretario sin presencia de vocales. Pero no podrían abrirla los tres vocales si no colabora o el presidente o el secretario.

Es decir, la repartición de un secreto mediante un esquema umbral (p, q) permite el establecimiento de reglas muy variadas y muy flexibles.

No obstante, estas soluciones de compartición de secreto siguen teniendo las mismas limitaciones que comentamos al describir el secreto dividido (al fin y al cabo son dos variaciones de una misma forma de proceder). Esta limitación estriba en que todo depende de la Autoridad de Seguridad porque las entidades depositarias de las muestras o sombras del secreto no tienen forma alguna de comprobar que lo que se les ha entregado es correcto. Si la Autoridad de Seguridad actúa maliciosamente las entidades involucradas no tienen forma de percatarse de ello hasta el momento en el que quieran reconstruir el secreto y no sean capaces.

Por todo ello, conviene insistir en que este tipo de soluciones son de interés sólo en el caso de que exista una confianza total en la autoridad que genera el reparto y, además, las entidades involucradas dispongan de dispositivos resistentes a manipula-

ción (*tamper resistant*) para almacenar su parte del secreto sin que ellas mismas sean capaces ni de leerlo ni de modificarlo.

11.2. TARJETAS INTELIGENTES AL SERVICIO DEL ANONIMATO

A lo largo de los capítulos precedentes, en multitud de ocasiones hemos venido haciendo reiteradamente referencia a la utilidad que podría representar el uso de tarjetas inteligentes en muy diversos escenarios.

En realidad, una tarjeta inteligente no es más que un dispositivo físico que permite almacenar información de seguridad de forma fiable que, además, debido a que tiene un microprocesador incorporado, puede gobernar tareas de entrada/salida y ejecutar algunos algoritmos criptográficos. Ello posibilita, entre otras muchas cosas, que pueda almacenar la clave privada de su titular y cifrar datos con ella. Pero muchos otros dispositivos podrían responder a esta breve descripción y, sin embargo, no han merecido tanta atención por nuestra parte. Luego debe haber algo más.

En efecto, hay, al menos, tres cosas más que hacen que las tarjetas inteligentes presenten para nosotros tanto interés. Su portabilidad (el aspecto externo es similar a una tarjeta de plástico convencional) y su bajo precio son dos de ellas. La otra es que están fabricadas de forma que los datos en ellas almacenados no se pueden leer ni alterar de forma fraudulenta.

Por todo ello, en las aplicaciones telemáticas seguras que interesan a los ciudadanos es frecuente encontrar que las tarjetas inteligentes son unos testigos de seguridad que ayudan grandemente a diseñar escenarios de comunicación complejos al tiempo que sirven para preservar la privacidad de los participantes.

Alguna limitación o inconveniente habrán de tener. Así es en efecto: debido a su reducido tamaño, la capacidad de proceso y, sobre todo, de almacenamiento es limitada. Pero aun así esta es una restricción que va siendo menor a medida que van evolucionando las tecnologías de fabricación. Por eso es acertado aventurar que las tarjetas inteligentes, u otros dispositivos de similares características que se desarrollen en un futuro, van a seguir jugando un papel muy relevante en el desarrollo de esas aplicaciones que protegen la privacidad de los ciudadanos.

De lo dicho hasta aquí se deduce que las tarjetas inteligentes son de utilidad para la provisión de **todos** los servicios de seguridad y no sólo para la provisión del servicio de anonimato. El hecho de que en el título del presente epígrafe hagamos hincapié en este último servicio es debido a la constatación, antes comentada, de su presencia en los escenarios de comunicación que servirán de soporte a la Sociedad de la Información. Por ello, más que hablar de las tarjetas inteligentes al servicio del anonimato, en este epígrafe lo que realmente haremos es hablar de las tarjetas inteligentes al servicio de las aplicaciones que posibilitan la seguridad y la privacidad en las comunicaciones.

En entornos más profesionales, por ejemplo en la gestión de claves de una CA, se dispone de otros dispositivos físicos de tamaño y grosor no mucho mayores que los de una tarjeta inteligente, que también son resistentes a manipulaciones (*tamper proof*), pero con mucha mayor capacidad de proceso y de almacenamiento. El conexionado que les permite entrar en contacto con el exterior es diferente al normalizado para las tarjetas inteligentes, por lo que no existe posibilidad de compatibilizar su uso a través de un mismo periférico de lectura-escritura. Estos dispositivos no serán objeto de atención en el presente epígrafe, orientado a evaluar las posibilidades que ofrecen las tarjetas inteligentes a los usuarios finales.

Para entender mejor en qué medida son interesantes las tarjetas inteligentes vamos a estudiar de nuevo el comportamiento de estos dispositivos. Ya en el Capítulo 2, cuando planteamos una visión general de los conceptos y elementos presentes en la Seguridad en Redes Telemáticas, adelantamos cuáles son las características principales de las tarjetas inteligentes tradicionales, es decir, aquellas se que se ajustan estrictamente a la norma ISO 7816. Ahora ampliaremos algunos detalles acerca de su estructura y funcionamiento para poder analizar con mayor conocimiento de causa el tipo de aplicaciones en las que pueden ser utilizadas.

Además, analizaremos someramente las características y posibilidades de uso de una nueva generación de tarjetas no comentadas en el Capítulo 2. Se trata de las tarjetas que operan con pequeñas aplicaciones (*applets*) codificadas en Java: las denominadas *Java Cards* o *Tarjetas Java*. La experiencia demuestra que este tipo de componentes ofrece mayores posibilidades a los desarrolladores de aplicaciones.

ESTRUCTURA INTERNA DE LA TARJETA «CLÁSICA»

Las dimensiones externas de la tarjeta de plástico en la que se inserta el circuito integrado están perfectamente normalizadas (ISO 7816) para conseguir la compatibilidad de los dispositivos de lectura a través de los cuales se comunica con el mundo externo (habitualmente se comunicará con otro computador desde el que se regula su utilización).

Como ya hemos venido diciendo, una tarjeta inteligente es un minúsculo microcomputador contenido en un circuito integrado o *chip* que posee (Figura 11.8) la correspondiente Unidad Central de Proceso (CPU), un elemental sistema de entrada/salida y un espacio de memoria para almacenar datos y programas.

Para reforzar la eficacia del microprocesador que opera como CPU, las tarjetas más potentes disponen de un coprocesador matemático que facilita la ejecución de los algoritmos criptográficos. La mayor limitación en la potencia de las tarjetas inteligentes viene dada por lo reducido del espacio de memoria de que disponen, aunque la mejora de los procedimientos de fabricación consigue que éste vaya siendo cada vez mayor.

Cuando se diseña una aplicación concreta para operar en conjunción con una tarjeta inteligente, lo primero que se decide es qué parte de esa aplicación correrá fuera de la tarjeta y cuál dentro. Otro tanto puede decirse en relación con las estructuras de datos que se manejen.

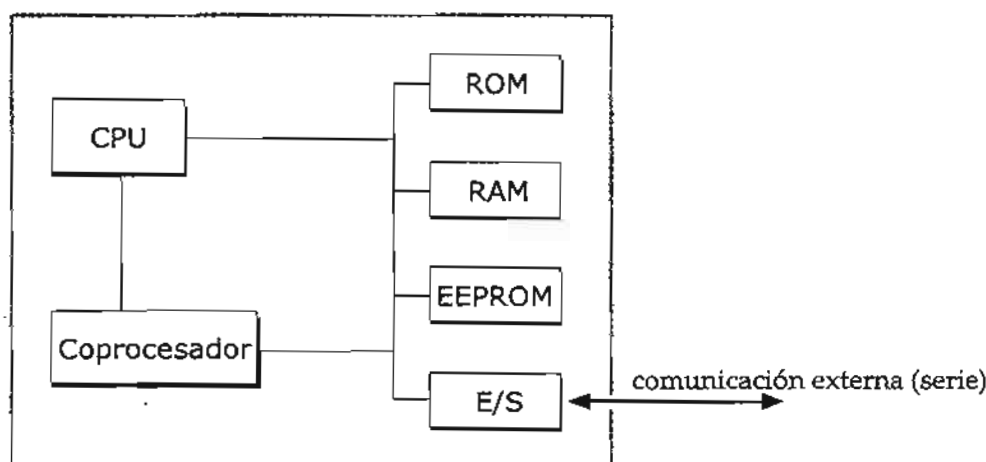


Figura 11.8. Elementos constitutivos de una tarjeta inteligente.

En lo que se refiere a la parte de los programas y los datos que residan dentro de la tarjeta, el pequeño sistema operativo que gobierne su funcionamiento y los programas que se consideren más estables irán grabados, desde el mismo proceso de fabricación, en la memoria de sólo lectura (ROM). Esta información es inalterable, lo que supone que una posible modificación motivada por una mejora o corrección de su comportamiento conllevaría desechar esa tarjeta y construir una nueva.

También, para guardar datos y estados que varíen durante la ejecución del programa, las tarjetas inteligentes disponen de un espacio de memoria de lectura-escritura y acceso directo (RAM). Por lo general, en este tipo de tarjetas, el tamaño de la memoria RAM es reducido (menos de mil octetos), lo que limita el tipo de operaciones que pueden llevarse a cabo a través del programa almacenado.

La memoria EEPROM* servirá para almacenar aquellos datos y programas que pueden verse modificados durante el tiempo de vida de la tarjeta. Esta memoria puede borrarse y regrabarse externamente por parte de personas que conozcan suficientes detalles sobre la aplicación concreta que se ha diseñado para esa tarjeta. De los ocho posibles contactos mediante los cuales la tarjeta se comunica con la unidad de lectura y escritura (ULE o CAD, *Card Acceptance Device*) existe uno de ellos, conocido como V_{pp} (Figura 11.9), que sirve para suministrar a la tarjeta el voltaje necesario (entre 12,5 y 21 voltios, según los casos) para poder regrabar esta EEPROM.

La forma, posición y cometido de estos ocho contactos está, como ya se ha dicho, perfectamente especificada en la norma ISO 7816. Aparte del ya mencionado V_{pp} , existen otros cinco contactos con la siguiente funcionalidad: a) V_{cc} para suministrar la tensión de cinco voltios necesaria para su funcionamiento; b) CLK para la entrada de reloj; c) un contacto GND para masa; d) un contacto RST de re arranque; y e) un solo contacto (I/O) para la comunicación de entrada/salida. Los otros dos contactos restantes (RFU) están reservados para otras posibles funciones.

Para evitar el desgaste de los contactos provocado por el uso, se han desarrollado tarjetas sin contactos que se comunican con el exterior mediante ondas electromagnéticas³. Este tipo de tarjetas son útiles en control de tránsito o medios de transporte en los que conviene evitar el trámite de insertar la tarjeta en el lector, aunque, debido a lo reducido de la potencia de emisión puesta en juego, es necesario que la tarjeta se aproxime mucho al dispositivo a través del cual se comunica.

Volviendo al esquema convencional de comunicación mediante contactos directos, el diálogo entre la tarjeta y lector se inicia cuando éste aplica una tensión de cinco voltios entre el punto V_{cc} y masa. A continuación, a través del contacto CLK introduce

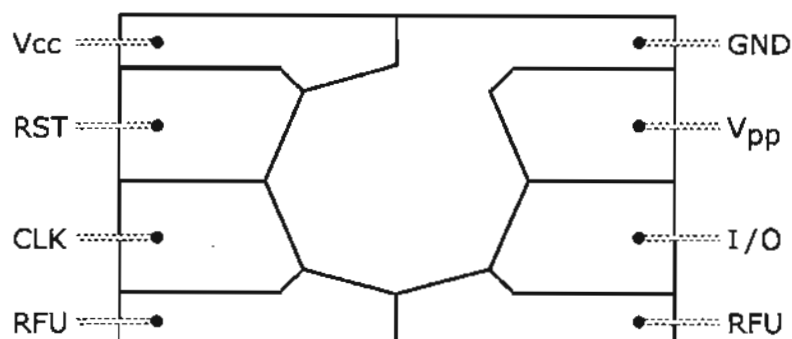


Figura 11.9. Los ocho contactos externos del microcomputador.

* Como es sabido, EEPROM es el acrónimo de *Electrically Erasable Programmable Read-Only Memory*.

la señal de reloj y activa el contacto de rearmado. Como respuesta a estos estímulos la tarjeta emite una respuesta (denominada ART) mediante la que comunica al mundo exterior los protocolos de transmisión que admite y los parámetros básicos [Dieg00] acerca de cómo ha de ser la comunicación (detalles sobre el reloj, lógica utilizada, etcétera). A partir de ahí será el lector el que lleve la iniciativa enviando peticiones a las que la tarjeta deberá responder.

Para posibilitar esa comunicación entre el lector y la tarjeta se han definido un conjunto de protocolos de transmisión (ISO 7816-3) en los que se especifica con detalle la estructura de las TPDUs intercambiadas (unidades de datos del protocolo de transmisión). Aunque hay definidos hasta 15 protocolos distintos, los más usados son el T=0 y el T=1. El primero de ellos realiza una transmisión semidúplex asíncrona de caracteres, y el segundo es similar aunque opera con bloques (de forma similar a un protocolo de nivel de enlace) en vez de con caracteres aislados. El hecho de que solamente se disponga de una línea serie limita bastante las posibilidades de la comunicación que puede establecerse.

Sobre este nivel de transmisión se define (ISO 7816-4) un protocolo de aplicación que regula la comunicación entre la tarjeta y el computador al que está conectado el CAD. Las unidades de dato de este protocolo de aplicación, APDUs, se intercambian de forma bidireccional alternada en forma de preguntas o comandos que genera el computador y respuestas de la tarjeta.

PROTECCIONES, FICHEROS Y AUTENTICACIONES

Para garantizar que sea muy difícil falsificar o manipular la tarjeta inteligente mediante ataques externos es necesario establecer protecciones tanto físicas como lógicas. Entre las primeras podemos destacar el hecho de que el microcomputador dejará de funcionar si alguien lo extrae de la tarjeta con la intención de analizarlo microscópicamente. Para ello, cuando el sistema integrado que contiene el microcomputador se introduce en la tarjeta, se somete a una presión mecánica permanente de forma que, detectando esta presión, el chip puede conocer si ha sido extraído o no.

Aunque el chip está protegido contra la luz ultravioleta y otras radiaciones, algún atacante podría intentar borrar algunas celdas de memoria. Por ello existen bits centinela distribuidos aleatoriamente en la memoria que serán comprobados por el microprocesador cada cierto tiempo y que, caso de que se borren, servirán de aviso para que el sistema deje de funcionar. Asimismo, otras posibles protecciones consisten en la ejecución de operaciones fingidas para evitar que se rastreen y averigüen las secuencias de instrucciones midiendo el consumo de energía.

Además de este tipo de protecciones, la tarjeta inteligente posee mecanismos que le permiten autenticar los datos que recibe y las entidades con las que se comunica. Los datos contenidos en la tarjeta están organizados (ISO 7816-4) según una estructura jerárquica en forma de árbol en cuya raíz se encuentra un denominado *Fichero Maestro*, o *MF*. La estructura puede contar con varios niveles y ficheros especializados en distintas tareas de administración o de almacenamiento de datos de usuario. El siguiente nivel jerárquico lo ocupa el *Fichero Dedicado*, o *DF*. Sólo existe un *MF* pero puede haber uno o varios *DFs* dependiendo de él (véase Figura 11.10) y pueden existir varios niveles jerárquicos de *DFs*. Por último, está normalizada la existencia de varios tipos de *ficheros elementales*, *EFs*, que son los verdaderos ficheros de datos y, como se refleja en la figura, pueden depender de cualquiera de los antes citados.

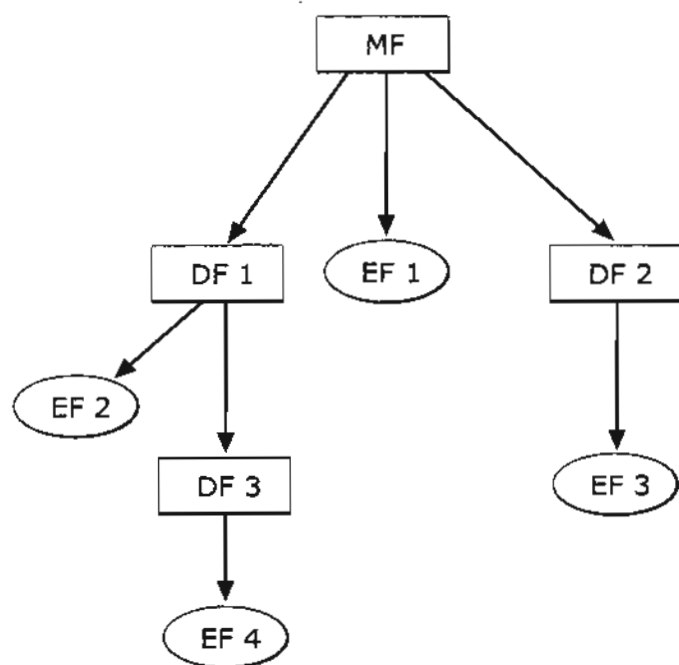


Figura 11.10. Sistema de ficheros en una tarjeta inteligente «clásica» o tradicional.

Sobre cada fichero se pueden realizar múltiples acciones (la norma enumera hasta 21) entre las que se encuentran las de selección, lectura, escritura, búsqueda, invalidación, etc., todas ellas llevadas a cabo mediante intercambio de PDUs. Cada fichero posee unas condiciones de acceso distintas, conforme a determinados niveles, para cada operación. El nivel más elemental consiste en que la entidad que pretende el acceso tiene que suministrar a la tarjeta un PIN o contraseña. Un nivel de acceso más riguroso se produce cuando para determinados ficheros se requiere que la entidad que accede (que se denomina «autoridad administrativa») comparta con la tarjeta inteligente una clave secreta cuya posesión tiene que demostrar, mediante un mecanismo de reto, para que se le permita el acceso.

En este último caso se produce un diálogo entre la tarjeta y la entidad externa que consiste, aproximadamente, en lo siguiente:

- La tarjeta genera un número aleatorio y se lo envía a la entidad.
- La entidad cifra ese número aleatorio con la clave secreta que comparte con la tarjeta.
- La tarjeta verifica el cifrado y autoriza o deniega el acceso.

La autenticación del usuario ante la tarjeta se consigue mediante la introducción de un PIN secreto que sólo debe conocer la persona que es su legítima propietaria. Más adelante discutiremos las circunstancias en que se puede aplicar este mecanismo de autenticación y las ventajas que aporta el uso de controles biométricos.

Como medidas de seguridad complementarias se utilizan determinadas restricciones en la autenticación de entidades externas consistentes en limitar el número de intentos fallidos en una misma sesión y exigir nuevos procesos de autenticación en el caso de que se produzcan anomalías en la alimentación o en las señales que se aplican en otros contactos.

Es decir, las tarjetas inteligentes tradicionales ofrecen bastantes posibilidades para garantizar un nivel de seguridad bastante razonable de cara al tipo de aplicaciones

sobre las que estamos centrando nuestro interés en el presente capítulo. No obstante, en cuanto a las posibilidades de introducir programas dentro de su memoria para que sean ejecutados internamente por su propio microprocesador, presentan grandes limitaciones que se pueden solventar utilizando una nueva generación de tarjetas inteligentes: es la generación de las *Tarjetas Java* o *Java Cards*. A ellas dedicaremos el siguiente apartado.

ESTRUCTURA Y FUNCIONAMIENTO DE LAS TARJETAS JAVA

Como su nombre sugiere, una *Tarjeta Java* o *Java Card* es una tarjeta inteligente en cuyo sistema microcomputador pueden ejecutarse pequeñas aplicaciones, llamadas *applets**, que operan y se programan conforme a la tecnología Java⁴. Estas *applets* se ejecutarán en la correspondiente Máquina Virtual Java, que en el caso de estas tarjetas recibe el nombre de JCVM (*Java Card Virtual Machine*).

El formato y las dimensiones externas de la tarjeta de plástico que les sirve de base son en todo compatibles con las que hemos denominado «clásicas» o tradicionales (aquellas que se ajustan a la norma ISO 7816 en sus siete partes). También la arquitectura del microcomputador contenido en el chip, así como la disposición y funcionalidad de los contactos que lo comunican con el mundo exterior, son conformes con dicha norma y responden al esquema representado en las Figuras 11.8 y 11.9 antes comentadas.

Esta compatibilidad se fundamenta en que las Tarjetas Java⁵ son totalmente conformes con la norma ISO 7816 en sus partes 1, 2 y 3. Debido a ello, los protocolos denominados *de transmisión* o *de transporte* mediante los cuales se comunican la tarjeta y el CAD (dispositivo de lectura/escritura) son coincidentes con los que ya

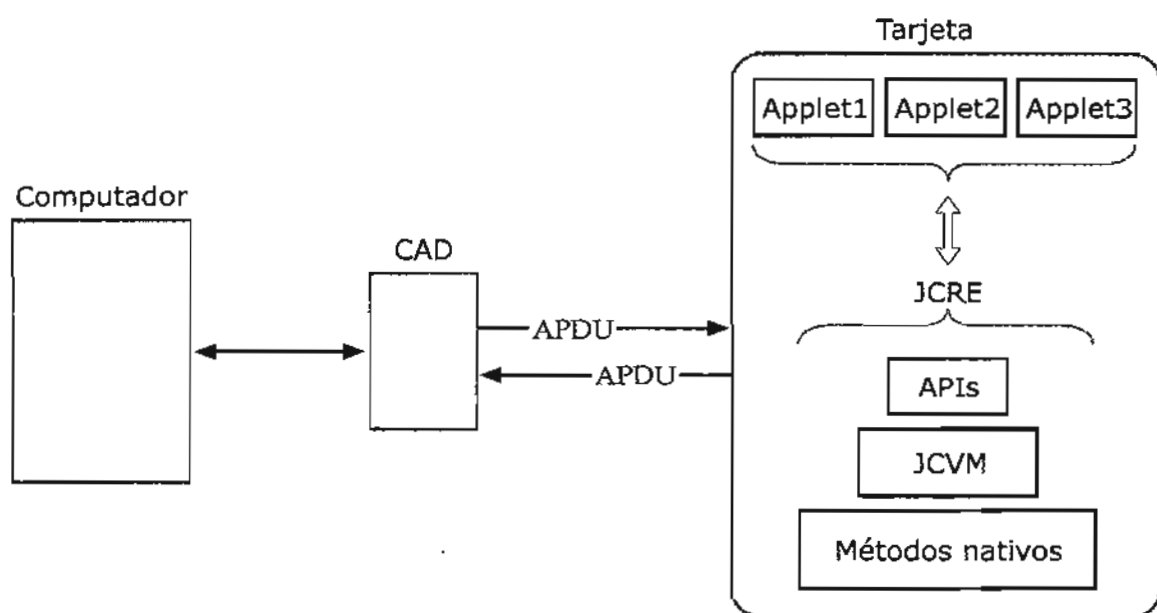


Figura 11.11. Estructura de la Tarjeta Java y esquema de comunicación.

* La palabra *applet* viene a significar «aplicacioncita» o «pequeña aplicación». Aunque en la jerga imperante en el mundillo de la programación es frecuente nombrarla en masculino, parece más razonable usarla en femenino, conservando el género de la palabra de la cual procede. En la versión en español del libro de Tanenbaum [Tane97] (un libro clásico sobre redes) se sigue este mismo criterio.

hemos referido. Por tanto, entre el CAD y la tarjeta existirá un intercambio de TPDU's (con una funcionalidad equivalente a la del nivel de enlace de los protocolos OSI), aunque en la Figura 11.11, que representa la estructura de una Java Card y el esquema de comunicación entre la tarjeta y el computador conectado a la red telemática, no se haya reflejado (por simplificar) este intercambio de PDU's.

En dicha figura sí se ha representado la existencia de unas APDU's, con semántica y funcionalidad propia, que se encastrarán, a efectos de la transferencia de información, en las correspondientes TPDU's.

La estructura lógica de la tarjeta se representa también, de forma muy simplificada, en la Figura 11.11 a la que venimos haciendo referencia. Como hemos dicho, los programas y los datos se alojarán en la memoria (materializada en componentes RAM, ROM y EEPROM) y las instrucciones de máquina serán ejecutadas por el correspondiente microprocesador (con ayuda, si existe, del coprocesador criptográfico). Cada fabricante podrá incorporar un microcomputador diferenciado con una arquitectura interna propia, de forma que, en último extremo, los programas deberán ejecutarse conforme a un conjunto específico de instrucciones de máquina. Debido a ello, en la parte inferior de la estructura lógica representada en la figura aparecen los *Métodos nativos* que se adaptarán a las características particulares de cada microcomputador y, además, servirán de soporte a la JCVM (*Java Card Virtual Machine*) y a otras clases del entorno de ejecución Java. Estos métodos serán los que proporcionen el acceso a los recursos físicos, protocolos de comunicación de bajo nivel, uso del coprocesador criptográfico, gestión de memoria, etc.

En la JCVM es donde se interpretan los códigos de bytes, y será la encargada de ejecutar las applets, garantizar la seguridad, gestionar objetos, organizar el reparto de memoria, etc. En un nivel inmediatamente superior se encuentra lo que hemos denominado con el nombre genérico de API, que englobaría las applets propias del sistema y las clases necesarias para proveer los distintos servicios soportados. Este bloque representa las funciones o tareas que pueden invocar las pequeñas aplicaciones de usuario (que se representan en la parte superior del diagrama), incluyendo los programas necesarios para cargarlas e instalarlas en la tarjeta.

El conjunto de estos tres bloques (métodos nativos, JCVM y APIs) constituye el entorno de ejecución de la Tarjeta Java o JCRE (*Java Card Runtime Environment*), que es el responsable global de la ejecución de los programas y de la gestión de todos los recursos. Es decir, se comporta, en la práctica, como el verdadero sistema operativo de la tarjeta.

Para adaptar la tarjeta inteligente a una aplicación concreta, las applets necesarias se programan cómodamente en un sistema informático convencional, se prueban y se depuran y después se instalan en la tarjeta. La tecnología Java permite al programador diseñar una applet sin tener en cuenta la plataforma física concreta sobre la que va a ejecutarse en la tarjeta, reduciendo los costes del desarrollo. Además, al trabajar con una programación orientada a objetos se facilita el diseño modular y la reutilización de los módulos.

Debido a que el programa se ejecuta mediante un intérprete, es posible probar las instrucciones antes de que se ejecuten, teniendo garantías de que la applet va a respetar los datos y los recursos de otras applets y no va a realizar operaciones indebidas. Esta característica es especialmente interesante cuando se requiere que en una tarjeta cohabiten varias aplicaciones o cuando se necesite ampliar la funcionalidad de una tarjeta añadiendo una nueva applet.

La limitación que tiene todo esto es que debido a la poca potencia del microprocesador que gobierna la tarjeta y a lo reducido del espacio de memoria en que pueden

residir los programas y los datos, la Java Card solamente soporta un subconjunto de los elementos * definidos en el lenguaje Java. Entre los elementos que no soporta cabe destacar:

- Tipos de datos primitivos y estructuras de datos que ocupan gran cantidad de octetos (*float, double, long, arrays multidimensionales...*).
- Hilos y sincronización.
- Clonación de objetos.

Es razonable que, dadas sus características, nunca pueda una tarjeta soportar definiciones de datos que ocupen gran cantidad de octetos, ni estructuras dinámicas complejas, pero también es cierto que la evolución de la tecnología hace que cada vez los microprocesadores sean más rápidos y que puedan implementarse memorias con mayor capacidad de almacenamiento. Por ello, es de esperar que, progresivamente, las limitaciones impuestas al lenguaje Java vayan siendo cada vez menores.

De lo que venimos diciendo acerca de la estructura lógica de una Tarjeta Java, donde el funcionamiento global está focalizado en la existencia de applets «independientes», cabe deducir que no tiene mucho sentido normalizar aquí un sistema de ficheros como el definido en la ISO 7816-4 para las tarjetas tradicionales (que se representa en la Figura 11.10). Consecuentemente con ello, en las Java Cards no está definida la autenticación que allí se hacía de entidades externas a la tarjeta (apoyándose en el intercambio de contraseñas o en la compartición de claves) para permitirles o denegarles el acceso a determinados ficheros.

En las Java Cards la autenticación de entidades externas (básicamente entidades clientes de una aplicación) se centra en la comunicación que dicha entidad desee establecer con una determinada applet. El primer paso en una comunicación entre la tarjeta y una entidad externa consiste en llevar a cabo una *autenticación mutua* entre ambas. Esto se realiza mediante un proceso de diálogo en el que se comparan unas claves, contenidas en un fichero, que deben ser las mismas en ambos extremos de la comunicación. La autenticación consiste en que la tarjeta comprueba que las claves que le presenta la entidad externa coinciden con las suyas propias y la entidad externa hace otro tanto con las claves que le presenta la tarjeta.

Una vez que la autenticación se ha llevado a cabo satisfactoriamente, se establece un canal seguro entre las dos entidades, apoyándose en el cual se intercambian las distintas piezas de información que constituyen los protocolos mediante los que se comunican.

Como dijimos más arriba al comentar la Figura 11.11, en las Tarjetas Java también están definidos los protocolos de transmisión y de aplicación y las correspondientes unidades de datos TPDUs y APDUs. No obstante, si bien en lo que se refiere al transporte las Tarjetas Java son totalmente conformes con la ISO 7816-3, en lo que se refiere a las APDUs normalizadas en la ISO 7816-4, esa conformidad sólo se produce en parte. Esto es así porque al no estar definido en esta nueva generación de tarjetas el sistema jerárquico de ficheros, todas las APDUs cuyo cometido era regular el acceso a los ficheros dejan de tener sentido. En contrapartida, será necesario definir nuevas APDUs que sirvan para establecer la comunicación entre las applets y el mundo exterior.

* El conjunto de elementos soportados está definido en la *Java Card Application Programming Interface Specification*.

A modo de resumen, podemos decir que ambos tipos de tarjetas, las Java Card y las que hemos dado en denominar tradicionales o «clásicas», tienen algunos elementos diferenciales en cuanto a la organización lógica de los datos y los programas residentes en el microcomputador, pero tienen bastantes elementos estructurales comunes: aquellos reflejados en la ISO 7816, partes 1, 2 y 3. Por esta razón, aquí no hemos utilizado el término *tarjetas ISO* para referirnos a las tarjetas convencionales que son conformes con la totalidad de la norma, y para diferenciar ambos tipos de tarjetas hemos huido, por tanto, de establecer el binomio *tarjetas ISO/Tarjetas Java*, porque ello induce a pensar que, por contraposición, el segundo tipo carece por completo de conformidad con la norma ISO. De forma que, para consumo interno, nos hemos despachado llamándole «clásica», así, entre comillas, a la tarjeta tradicional con microprocesador (que no es la misma que otra tarjeta más clásica todavía: la tarjeta de memoria sin microprocesador).

11.3. ANONIMATO EN LOS MEDIOS DE PAGO

Una de las actividades usuales en las sociedades organizadas es el intercambio de bienes y servicios a cambio de una prestación económica. Dentro de la Sociedad de la Información, este tipo de actividades han de tener su plasmación en el mundo de las Comunicaciones Mediante Computador (CMC). Es por ello que ya en la actualidad existe un elevado número de aplicaciones que podemos calificar bajo el nombre genérico de Comercio Electrónico, las cuales tratan de conjugar las ventajas indudables que ofrecen las comunicaciones telemáticas con las garantías que tradicionalmente ofrecen las operaciones comerciales convencionales.

La insuficiencia de esas garantías es lo que ha motivado que la implantación de este tipo de actividades en la Red haya sido, hasta el presente, más lenta (y menos lucrativa) de lo que algunos imaginaron frotándose las manos.

La firma electrónica es, por las razones que ya vimos en los capítulos en los que abordamos su estudio, uno de los pilares en que se sustentan esas garantías, por lo que es de esperar que cuando den fruto las iniciativas técnicas y legales que promueven su uso generalizado, ello generará un impulso en las actividades comerciales que se apoyan en redes telemáticas.

Otro aspecto que afecta de lleno a la implantación masiva de actividades de intercambio comercial en la Red es el relacionado con la pérdida de privacidad que puede representar el que determinados poderes económicos conozcan exhaustiva y pormenorizadamente cuándo y en qué los ciudadanos se gastan su dinero. Es decir, el efecto que causa la ausencia del necesario anonimato en determinadas operaciones de compraventa. Bien es verdad que este es un aspecto que, en principio, no preocupa a esos centros de poder económico (porque reporta un beneficio para ellos), pero sí es algo que preocupa a la ciudadanía por cuanto tiene de lesivo para sus derechos y libertades. Y por esa razón debería importar a los poderes públicos si, como proclaman, son garantes e impulsores de esos derechos cívicos.

En este contexto, existen en la actualidad (y es de esperar que aumenten en un futuro) numerosas propuestas acerca de Comercio Electrónico, con y sin servicios de anonimato incluido. En los párrafos que siguen no describiremos ni unas ni otras (como tampoco hemos descrito las aplicaciones convencionales de correo, web, transferencia de ficheros, etc.), sino que solamente nos centraremos en detectar los aspectos genéricos que afectan al anonimato, verdadero objetivo de este capítulo. Lo que sí haremos es hacer referencia a algunas partes de algunas propuestas existentes.

ESCENARIOS DE DINERO DIGITAL ANÓNIMO

Para la realización de compras de bienes y servicios a través de redes telemáticas será necesario emular los comportamientos existentes en el «mundo real». Tradicionalmente, los dos medios de pago (Figura 11.12) que han venido utilizándose han sido:

- a) *Dinero en efectivo*. También llamado *dinero en metálico* o *dinero corriente*, es el constituido por monedas o billetes expedidos por un banco central de confianza para las partes que intervienen en una transacción.
- b) *Dinero en forma de pagarés, cheques o tarjetas*. Al no tratarse de dinero a la vista, cuando se usa esta forma de pago, de alguna manera entra en juego la credibilidad y la solvencia de la persona que entrega ese dinero.

Por emulación, llamaremos genéricamente *dinero digital* o *dinero electrónico* a la plasmación en la Red de estos medios de pago. Por *dinero digital* podemos entender, por tanto, cualquier intercambio de piezas de información, en formato electrónico, del que resulte una transferencia de fondos entre dos o más partes. En una operación de compraventa el intercambio de esa pieza de información debe ofrecer garantías al vendedor de que obtendrá, directamente o a través de una entidad bancaria, una cantidad de dinero equivalente al precio de la mercancía que acaba de ceder.

Emular al dinero en metálico no es tarea sencilla. Es cierto que su uso resulta a veces incómodo y sujeto a múltiples limitaciones, pero es necesario resaltar que el dinero en metálico es un viejo mecanismo que permite mantener totalmente el anonimato del comprador. El vendedor tiene una garantía razonable de que el comprador está en posesión de un recurso que le permite llevar a cabo una determinada transacción comercial, siendo habitualmente desconocida la identidad del cliente. Por otra parte, el banco puede conservar un registro de todas las cantidades de dinero que hemos sacado de la cuenta corriente, pero no puede saber en qué hemos gastado ese dinero.

Los cheques o pagarés son propiamente documentos que se intercambian entre las partes que intervienen en la transacción, y nunca ofrecen anonimato (de ningún tipo), por lo que el análisis de su emulación en forma de dinero digital es algo que no nos interesa de cara a los objetivos que nos hemos planteado en este capítulo.

Caso aparte son las tarjetas. Por tres razones: a) porque en gran medida ya están de por sí integradas en las redes telemáticas (por lo que no ha lugar hablar de su «plasmación digital»); b) porque, generalmente, en su uso cotidiano no ofrecen anonimato (situación que no nos interesa emular); y c) porque nos interesa hablar de propuestas emergentes de tarjetas (soportadas en tarjetas inteligentes) que sean más seguras que las convencionales y además ofrezcan anonimato.

Es decir, en este caso no se trata de garantizar en la Red los derechos de los ciudadanos que ya garantizan las tarjetas, sino de evaluar propuestas que posibilitan la inclusión de servicios de seguridad que sirvan para que las nuevas tarjetas ofrezcan mayor privacidad que las convencionales.

Básicamente, en cuanto a su utilización como medios de pago, podemos considerar que existen tres tipos de tarjetas:

- **Tarjeta-monedero**. Suele denominarse así a aquella que una vez cargada en un cajero automático o dispositivo *ad hoc*, permite ir haciendo uso de ese dinero hasta su finalización. En el momento de la carga será deducido el importe de la cuenta corriente de su titular.

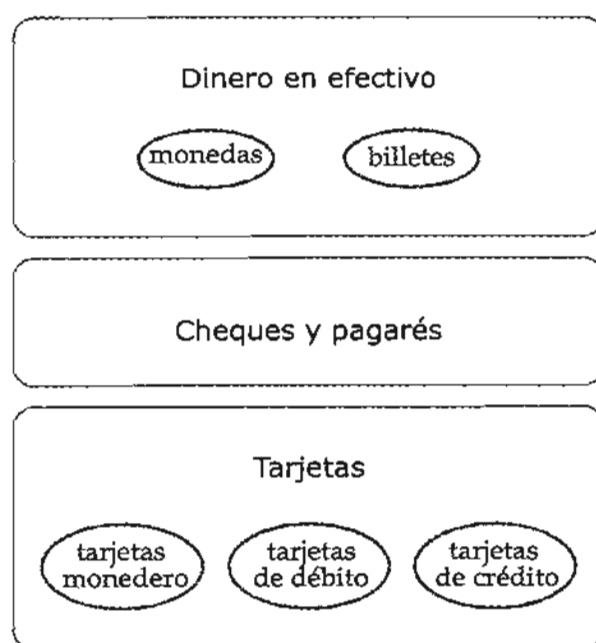


Figura 11.12. Medios de pago tradicionales.

- **Tarjetas de débito.** Por tarjeta de débito se entiende aquella en la que el banco deduce de la cuenta corriente la cantidad equivalente a la compra en el mismo momento en que ésta se produce.
- **Tarjetas de crédito.** Son aquellas en las que el banco, tras comprobar que la persona titular de la tarjeta tiene crédito suficiente, anota la cantidad resultante de la compra efectuada y transcurrido un plazo de tiempo prefijado (generalmente un día concreto de cada mes) lo deduce de la cuenta corriente junto con el montante de otras compras realizadas.

De cara a su plasmación en escenarios de pago con anonimato, sólo nos interesan cuando son de propósito general, es decir, carece de interés para nosotros pensar en una tarjeta telefónica (aunque sí sean anónimas) que solamente sirve para llamar por teléfono. Por otra parte, a los efectos de lo que aquí nos interesa, no existen diferencias sustanciales entre ellas, ya que en lo único que se diferencian es en el momento en que se realiza el descuento de fondos. Debido a esto, cuando en los párrafos que siguen se haga alusión a las «tarjetas de crédito» debe entenderse que los razonamientos que se utilicen pueden extenderse a las tarjetas de los otros dos tipos. Evidentemente, en cada aplicación y en cada escenario concreto se definirá qué tipo de tarjeta se usa, pero esto no afecta significativamente al tipo de mecanismos criptográficos que se necesitan para conseguir el anonimato.

Ya en la actualidad se constata que, de forma creciente, un gran número de operaciones comerciales son llevadas a cabo por los ciudadanos mediante tarjetas de crédito. El uso masivo de tarjetas de crédito y los problemas de vigilancia que ello conlleva ha alertado a amplios sectores de la sociedad y atraído la atención de muchos investigadores sociales⁶, que consideran este hecho como sumamente pernicioso para la conservación de la privacidad (entendida ésta, en el sentido que recogíamos en el Capítulo 1, como el derecho ciudadano a mantener protegido aquello que afecta a comportamientos sociales que sólo incumben a una persona o un grupo reducido de ellas).

Esto es así porque puede darse la situación, a medida que se vaya implantando la Sociedad de la Información, de que muchísimas de nuestras actividades, desde que nos levantemos hasta que nos acostemos, estén grabadas en registros informáticos. Frente a este peligro, la inclusión de servicios de anonimato en las transacciones puede suponer un conjuro contra esa pesadilla de vigilancia y totalitarismo que algunos investigadores sociales temen que pueda producirse con la introducción masiva de medios de pago electrónico.

Para ir centrando el tema, sin excluir el interés que, en general, puede tener el uso de tarjetas de crédito con soporte telemático e incluso con servicios de seguridad y de certificación (como puede ser el caso de SET, comentado en el Capítulo 7), de cara a la tipificación de escenarios de dinero digital anónimo, un primer aspecto que conviene anotar es que las **tarjetas de pago** que tienen mayor interés (y que serán, por tanto, en las que nos fijemos) serán aquellas que se materializan en **tarjetas inteligentes**.

(El solo uso de **tarjetas inteligentes**, sin servicios de anonimato, para configurar tarjetas de crédito o tarjetas de débito no sólo no resuelve el problema de la vigilancia social, sino que lo magnifica, debido a que facilitaría aún más el establecimiento de registros informáticos, con el enorme peligro que ello conlleva de pérdida de privacidad.)

Otro aspecto importante para caracterizar estos escenarios es el relacionado con la naturaleza de las partes que intervienen en las transacciones. Básicamente podemos considerar dos tipos:

- a) Transacciones de compraventa en las que interviene una entidad *cliente* y una entidad *vendedora*.
- b) Transacciones o transferencias de fondos entre ciudadanos.

A efectos de necesidades de anonimato, estas últimas no tienen ese requisito, ya que lo normal es que sean nominales. Aunque algunos sistemas, como es el caso de *Mondex**, que utilizan tarjetas inteligentes como tarjetas de pago, contemplan esta posibilidad de intercambio de fondos entre tarjetas, aquí **nos centraremos en escenarios cliente-vendedor**, que es en los que tiene interés la provisión de anonimato.

Un tercer aspecto a tener en cuenta tiene que ver con la forma en que la entidad cliente accede a la red telemática en la que se sustenta el sistema de pago. Para el caso que nos ocupa, el cliente puede acceder:

- a) Mediante una tarjeta inteligente especialmente adaptada a las necesidades del sistema que actúe ante él como un testigo de seguridad reconocible y resistente ante manipulaciones. En este caso, la tarjeta puede contener (además de la clave privada del cliente que le servirá de respaldo en las operaciones de firma digital) otras claves o datos de la aplicación e incluso determinados algoritmos específicos. La tarjeta inteligente puede facilitar grandemente la autenticación mutua entre la entidad cliente y el sistema global.
- b) Directamente a partir de los sistemas informáticos de que disponga el cliente, de forma que las claves, los datos y los algoritmos que le permiten acceder al sistema están contenidos y programados en sus propios recursos computacionales y tanto la custodia como la exactitud de esa información está bajo su exclusiva responsabilidad.

* *Mondex* (<http://www.mondex.com>) es un sistema de pago en el que el dinero está almacenado en tarjetas inteligentes.

De la propia redacción de ambas opciones se deduce que, aunque la opción reflejada en segundo lugar es perfectamente válida, la alternativa más interesante es la de acceder con testigos de seguridad especializados, es decir, **acceder sistemáticamente mediante tarjetas inteligentes**. Podríamos hablar en ese caso de *tarjetas monedero anónimas* o de *tarjetas de débito anónimas* o de *tarjetas de crédito anónimas* o, de forma genérica, de *tarjetas de pago anónimas*. Lo que ocurre es que, debido a las posibilidades que ofrecen las redes telemáticas y los servicios de seguridad, estas tarjetas pueden tener, como veremos más adelante, un comportamiento mucho más parecido al dinero en metálico que el que pueden ofrecer las tarjetas tradicionales de las cuales, por evolución, proceden.

Otro aspecto también relacionado con el acceso, que podemos considerar como un desglose de la opción que acabamos de catalogar como a), es el relativo al terminal de acceso al sistema. Si partimos de que la persona que actúa como cliente lo hace mediante su tarjeta inteligente, será necesario, obviamente, que el terminal de acceso disponga de un dispositivo de lectura-escritura (CAD) para este tipo de componentes. No obstante, cabría contemplar dos alternativas:

- a1) El terminal de acceso puede ser un computador cualquiera que disponga de un CAD para comunicarse con la tarjeta. En ese computador deberá residir el programa correspondiente a la Aplicación Cliente que se haya diseñado en ese sistema de pago.
- a2) El terminal de acceso es un subsistema especialmente diseñado para servir de intermediario con la tarjeta inteligente del cliente. Lo razonable es que esté desarrollado sobre un computador de reducidas prestaciones pero de forma que sea resistente a manipulaciones (*tamper resistant*), es decir, que ni los programas ni los recursos que gobierna puedan ser alterados fraudulentamente.

Aunque la primera puede ser perfectamente válida, es fácil colegir que es la segunda alternativa la que ofrece más posibilidades a los diseñadores de la aplicación global. En efecto, si se dispone de un **terminal de acceso específico**, podrán asignársele algunas de las funcionalidades del sistema que, de forma conjunta, garantizan el anonimato y la seguridad del conjunto global.

Un último aspecto que vamos a tener en cuenta (dentro de muchos otros que sería posible considerar) es el referido a la «distancia» entre el comprador y el vendedor. Es un error bastante extendido el considerar que la principal aplicación de pago con dinero digital a través de redes telemáticas es la que se produce cuando el comprador se encuentra en un sitio remoto y separado, por tanto, del vendedor. Es decir, se trataría típicamente de una compra que se hace desde el domicilio a través de Internet. Esta acepción excluiría, o dejaría en un segundo plano, a las transacciones comerciales que se realizan *in situ*, es decir, en el propio establecimiento del vendedor.

Por contra, la mayoría de las operaciones que se realizan con las tarjetas de crédito, que tanto desasosiego y atención despiertan en los investigadores sociales, se realizan de forma «presencial» en los propios locales del vendedor. Si bien es cierto que determinados productos como libros, discos, material informático, etc., se prestan a la compra remota, la compra de muchos otros requiere (al menos dentro de la cultura en que nos hallamos) tener a la vista, y al tacto, la mercancía que se desea adquirir. (En este sentido, la experiencia de los Estados Unidos, con mucha costumbre previa de compra por correo y una mentalidad muy especial, no es transportable a otras latitudes.) Por todo ello, si se consiguiera sustituir las transacciones convencionales

mediante tarjetas de crédito por otras en las que se respetase el anonimato del comprador, se habría dado un paso de gigante de cara a respetar las garantías de privacidad de los ciudadanos.

De cara a la tipificación de escenarios que venimos abordando, la compra *in situ* facilitaría la implantación de **terminal de acceso específico** en los puntos de venta, en tanto que la compra remota a través de Internet lo dificultaría grandemente. No obstante, es factible concebir un sistema en el que todos los participantes afiliados dispongan de un terminal especialmente diseñado para ese sistema. Un sistema que contempla esta alternativa es el sistema *Mondex* antes citado, en el que se proporciona a sus miembros un pequeño terminal «de bolsillo» con lector de tarjeta, teclado y pantalla incluidos.

En resumidas cuentas, de lo que hemos venido comentando en los párrafos precedentes se deduce que, combinando distintas opciones para distintos aspectos, pueden establecerse escenarios muy diversos. No obstante, sin desdeñar ninguna otra alternativa, podríamos seleccionar lo que podríamos llamar un «escenario interesante» (conforme está representado en la Figura 11.13) cuya constitución sería de la siguiente forma:

- *Cliente*. Es la persona que desea realizar la compra. Deberá tener cuenta corriente o cuenta de crédito en el *banco* integrado en el sistema.
- *Tarjeta de pago anónima*. Es una tarjeta inteligente especialmente adaptada a las necesidades del sistema. Actúa ante él como un testigo de seguridad reconocible y resistente ante manipulaciones (*tamper resistant*). Además de la clave privada del Cliente, contendrá otras claves o datos de la aplicación y, en algunos casos, podrá ejecutar determinados algoritmos específicos.
- *Terminal de acceso al sistema*. Se trata de un terminal especialmente diseñado, instalado en los puntos de venta, que dispone del correspondiente CAD para establecer comunicación con la *tarjeta de pago anónima*. Tendrá asignadas algunas de las funcionalidades del sistema que ayuden a garantizar el anonimato y la seguridad del conjunto global. Estará desarrollado sobre un computador de reducidas prestaciones y será resistente a manipulaciones.
- *Vendedor*. Es el titular de la *tarjeta inteligente de punto de venta* con la que se tiene acceso y se gobierna el *terminal de acceso al sistema*, interactuando, llegado el caso, con la *tarjeta de pago anónima*.

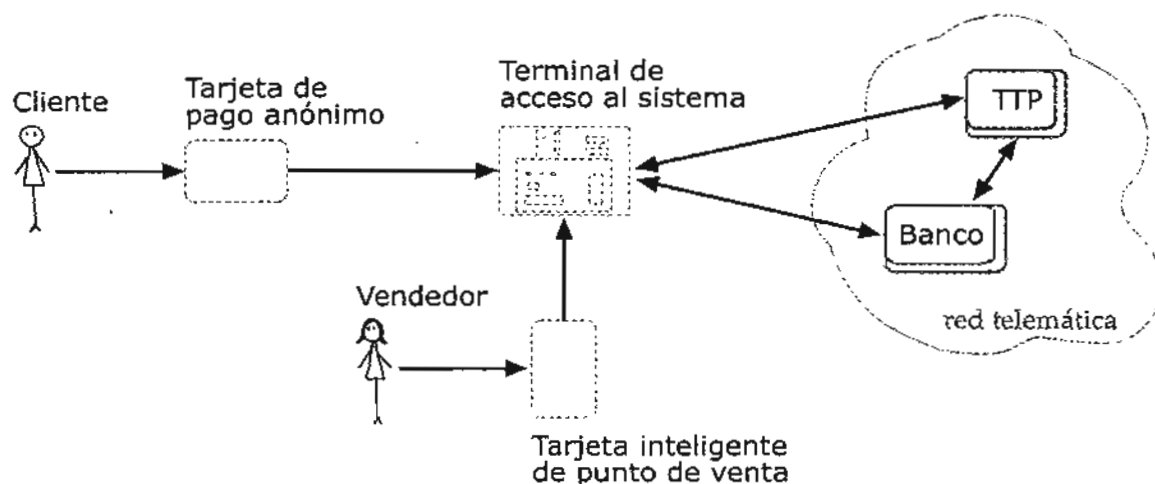


Figura 11.13. Un escenario «interesante» para dinero digital con anonimato.

- *Banco (o Central de Autorización)*. Es la entidad bancaria que reconoce el dinero digital manejado en el sistema y que está habilitada para deducir fondos de la cuenta del comprador e ingresarlos en la cuenta del Vendedor.
- *Tercera parte de confianza*. Es una entidad que puede aparecer de forma optativa. Caso de estar presente, puede servir para facilitar el comportamiento del sistema en cuanto a la provisión del necesario anonimato y/o actuar como garante en caso de un uso indebido del sistema.

Un escenario tal como este podría permitir, utilizando dinero digital, un comportamiento equivalente al que tiene lugar usando dinero en efectivo. De forma muy general, podríamos resumirlo diciendo que el banco *sabe cuánto se gasta* la persona que realiza las compras, pero *no sabe ni en qué ni dónde*, y el Vendedor sabe que *recibirá el dinero* equivalente al bien que cede, pero *no sabe quién es* la persona que compra ese bien. En el siguiente apartado trataremos de caracterizar más pormenorizadamente esta idea.

El escenario de la Figura 11.13 está adaptado al caso en el que la compra se realiza *in situ* y el terminal de acceso está ubicado en el punto de venta. No obstante, es adaptable de forma inmediata al caso en el que la compra sea remota y el terminal de acceso esté al lado del Cliente pero no al alcance del Vendedor. En este último caso no existiría la tarjeta inteligente del Vendedor como elemento que gobierna el terminal de acceso e interactúa, en algunos casos, con la tarjeta del comprador.

CARACTERÍSTICAS DEL DINERO DIGITAL ANÓNIMO

Como hemos dicho, el dinero en efectivo siempre es anónimo. Por ello, cualquier proyección digital de los mecanismos de pago debería preservar algún tipo de anonimato del comprador. Podemos distinguir al menos ocho características [Carr02] que sirven para catalogar un sistema anónimo de pago usando dinero digital:

1. **Verificabilidad** (*verifiability*). Todo participante en una transacción monetaria digital debe ser capaz de verificar el valor del dinero recibido y su autenticidad, lo que conlleva identificar a la entidad financiera emisora.
2. **Seguridad** (*security*). El dinero digital no puede ser copiado, o rehusado por el mismo comprador. Tanto el Comprador como el Vendedor tienen serias dificultades para perpetrar un fraude.
3. **Anonimato** (*anonymity*). La identidad del comprador debe ser protegida (esta característica ha sido ampliamente discutida en apartados anteriores).
4. **Irrastreabilidad** (*untraceability*). Nadie puede rastrear o detectar la relación entre el consumidor y los bienes adquiridos.
5. **Transferibilidad** (*transferability*). El dinero recibido puede a su vez ser utilizado por el Vendedor para realizar él mismo otras compras o transacciones, además de aquella que le permite el ingreso de esos fondos en su cuenta del banco.
6. **Divisibilidad** (*divisibility*). Quien recibe una cantidad de dinero electrónico está capacitado para transferir a terceros el total o solamente una parte de esos fondos.
7. **Devolución de cambio o vuelto**. Un Comprador puede entregar al Vendedor una cantidad de dinero superior al valor del bien adquirido y recibir del Vendedor una transferencia monetaria correspondiente a la diferencia.

8. **Pago sin conexión** (*off-line payment*). Los protocolos de venta se llevan a cabo entre el consumidor y el comerciante sin necesidad de que el punto de compra establezca una conexión directa con cualquier banco o agencia financiera.

De esas ocho características, que se corresponderían punto por punto con la plasmación en la Red del dinero en efectivo (*digital cash*), podemos afirmar que:

- Las cuatro primeras son imprescindibles para que el sistema pueda ser considerado de pago con anonimato.
- Las tres siguientes, *transferibilidad*, *divisibilidad* y *devolución de cambio*, son complementarias y útiles. Pero un sistema puede considerarse eficaz y robusto sin necesidad de soportarlas.
- La clasificada en octavo lugar, *pago sin conexión*, es menos importante. En algunos casos podría ser de utilidad porque es una propiedad que tiene el dinero en metálico. No obstante, no aparece recogida en el escenario que hemos considerado «interesante», el cual está focalizado al caso del uso de las redes telemáticas para realizar pagos mediante tarjetas inteligentes. Cabe observar, por otra parte, que, actualmente, en el uso convencional de tarjetas de crédito, los puntos de venta ya están mayoritariamente conectados telemáticamente con las entidades bancarias.

Entre las cuatro imprescindibles, la *irrastreabilidad* aparece diferenciada del *anonimato* porque puede que el banco, al expedir el dinero, no tenga forma de conocer dónde se ha gastado ese dinero (anonimato), pero sea capaz de buscar relaciones entre ambas operaciones. Por ejemplo, supongamos que en una fecha concreta a una hora concreta se va a realizar una compra de combustible en una gasolinera por valor de 50,12 euros. Cuando el comprador solicite ese dinero al banco, aunque el sistema garantice que la operación es anónima (el banco no podrá conocer ni la hora, ni la identidad del Vendedor, ni la mercancía que se compre con ese dinero), si el banco tiene voluntad de rastrear los apuntes realizados, no sería muy difícil diseñar programas que comparasen reservas de dinero (realizadas por el Cliente) con operaciones de cobro (realizadas por parte de los vendedores), llevadas a cabo en horas próximas, con lo cual, una cantidad tan específica como es 50,12 sería fácilmente relacionable con ambas operaciones. Por tanto, para conseguir la *no rastreabilidad* será necesario diseñar las cosas para que una situación como la que acabamos de describir no sea factible. Una posible solución sería fraccionar las reservas de dinero o fraccionar los pagos.

En el escenario que se recoge en la Figura 11.13, la *transferibilidad* representaría que el dinero que paga el Cliente pudiera almacenarse en algún elemento del entorno del Vendedor, por ejemplo en su *tarjeta inteligente de punto de venta*, y que el Vendedor pudiera más tarde utilizarla como *tarjeta de pago anónima* ante otro Vendedor.

En ese mismo escenario, la *divisibilidad* significaría que cuando el Cliente, al comunicarse con el banco a través del terminal de acceso, obtuviese una cierta cantidad de dinero y la guardase, por ejemplo, en su *tarjeta de pago anónima*, tuviese la posibilidad de gastar solamente una parte de él reservando el resto para otra ocasión. De otra parte, la *devolución de cambio o vuelto* (que es una característica menos aprovechable que las dos antes descritas) lo que podría representar en ese escenario es que, por ejemplo, desde la *tarjeta de pago anónima* se pudiese pasar al entorno del

Vendedor una cantidad de dinero y desde este último se le pudiese hacer una transferencia de fondos en sentido inverso.

11.4. MECANISMOS Y PROTOCOLOS PARA CONSEGUIR DINERO DIGITAL ANÓNIMO

Como ya hemos dicho antes, el escenario representado en la Figura 11.13 es uno de tantos que pueden establecerse, aunque nosotros lo hayamos seleccionado por entender que reúne los elementos más interesantes de cara a la evolución que han de experimentar en un futuro los sistemas de pago con anonimato. A pesar de su falta de generalidad, nos ha servido hasta aquí para explicar las características de este tipo de sistemas y nos servirá en el presente apartado para discutir acerca de los mecanismos y protocolos que pueden establecerse para conseguir ese comportamiento.

Antes comentamos que no vamos a tratar de describir las distintas aplicaciones y propuestas que se han venido realizando acerca de los medios de pago anónimos. Del mismo modo, tampoco vamos a abordar aquí un análisis del estado del arte en relación con este tipo de propuestas⁷.

Lo que sí haremos es analizar por partes los principales problemas que se presentan y tratar de ver cómo se solucionan en dos propuestas prácticas distintas⁸, compatibles con el escenario representado en la Figura 11.13, que vamos a denominar respectivamente *Sistema1* y *Sistema2*. A medida que avancemos en la explicación iremos añadiendo alguna de las particularidades de ambos sistemas.

FORMATO DEL DINERO DIGITAL

Aunque es viable imaginar muchas formas distintas de representar el dinero digital que el Cliente necesita obtener del banco para realizar la compra, cabe seleccionar dos de ellas bastante diferentes entre sí. Una de ellas consiste en considerar este dinero como un documento digital en el que figura la cantidad total y está firmado por el banco: una especie de pagaré al portador. La otra es considerarlo descompuesto en un conjunto de billetes digitales firmados individualmente por el banco.

En el *Sistema2* se trabaja con la primera de estas dos representaciones, más próxima al concepto de dinero manejado en las tarjetas de crédito o de débito. Para entendernos, a esa cantidad de dinero obtenido por el Cliente vamos a denominarla *autorización*. En principio, vamos a suponer que el formato que tendrá será el de un documento en el que figura un identificador único y la cantidad total de dinero, todo ello firmado por el banco. A ese identificador lo denominaremos Número de Identificación de la Autorización (NIA). Este número será elegido al azar por el Cliente y tendrá un número de bits suficientes como para garantizar que la probabilidad de repetición sea muy baja.

La segunda opción consiste en la definición de un conjunto de *billetes digitales*, cada uno representando un valor monetario distinto. Por ejemplo, 1 euro, 2 euros, 5 euros, 100 euros, etc., y sus correspondientes valores decimales. El valor monetario del billete está determinado por la clave privada monetaria con la que lo firme el Banco. Para ello, el Banco dispone de tantos pares de claves (pública y privada) como valores de billetes sean puestos en juego. En el *Sistema1* será ésta la representación elegida y cada billete concreto será un documento digital con un valor monetario y con un identificador único al que denominaremos Número de Identificación del Billeto

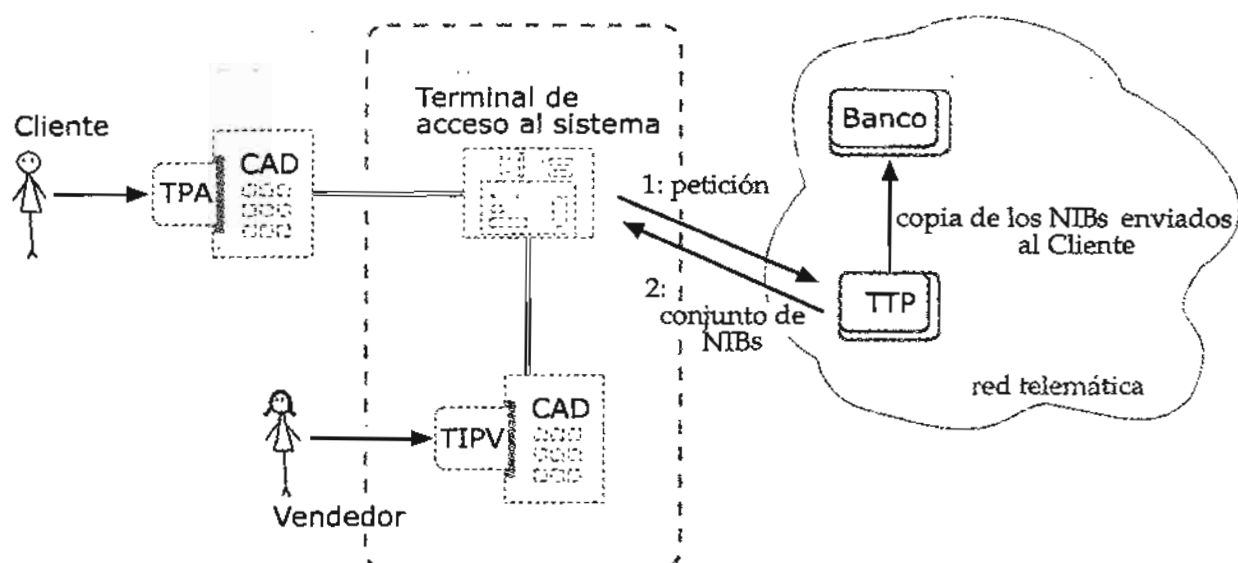


Figura 11.14. Solicitud de identificadores de billetes en Sistema1.

(NIB). Estos identificadores son generados por una TTP y entregados, a petición del Cliente, tanto a su *tarjeta de pago anónimo* (a la que, para abreviar, denominaremos TPA) como al Banco, en las condiciones que más adelante detallaremos.

En la Figura 11.14 se representa este intercambio de información. Como puede observarse, la TTP es parte imprescindible en el funcionamiento del Sistema1. Es como si lo que hiciese la TTP es enviar a la tarjeta billetes en blanco sin valor monetario y sin firmar, y al Banco le manda la relación de los identificadores que ha distribuido. En cambio, la estructura del Sistema2 sería casi igual a la representada en esta figura, pero sin la presencia de la TTP. Como veremos más adelante, el prescindir de la TTP simplifica el sistema pero obliga a que los protocolos sean más complejos.

LA FIRMA DEL DINERO POR PARTE DEL BANCO

El proceso de compra comenzaría de forma similar tanto si la operación se hace sobre el Sistema1 como si se hace sobre el Sistema2. Más o menos sería:

1. El Cliente elige una serie de productos.
2. El Cliente introduce su tarjeta (TPA) en el CAD. Se produce una autenticación mutua entre la TPA y el terminal de acceso al sistema.
3. El Cliente se autentica ante su TPA mediante un PIN o prueba biométrica.
4. El Vendedor a través del terminal le indica a la TPA la cantidad de dinero a pagar.

Esto es así suponiendo que todo vaya bien. Obviamente, si alguno de los pasos falla, se interrumpe el proceso. A partir de aquí, la tarjeta tendrá que comunicarle al Banco la cantidad a pagar y éste tendrá que firmar ese dinero para que posteriormente la tarjeta del Cliente se lo entregue al terminal de venta.

a) Hay que convencer al Banco para que firme

Como la representación del dinero es diferente en Sistema1 que en Sistema2, el proceso seguido para obtener la firma a ciegas del Banco no será igual en ambos.

Veamos cómo se comportaría el Sistema1:

1. La tarjeta TPA selecciona de entre los billetes en blanco que tiene almacenados la cantidad necesaria para hacer frente al pago.
2. La tarjeta incorpora un elemento de opacidad distinto en cada billete (véase apartado de firma a ciegas) para que los identificadores NIBs no sean visibles, y se los envía al Banco, indicándole, en claro, el valor monetario de cada uno de ellos.
3. El Banco comprueba la disponibilidad de fondos del Cliente y realiza una firma a ciegas sobre los NIBs opacados recibidos, cada uno con la clave privada monetaria que le corresponda. Descuenta el dinero de la cuenta del Cliente* y le envía los billetes firmados.
4. La tarjeta, tras verificar que lo recibido se corresponde con su petición y que la firma es correcta (dispone de las correspondientes claves públicas monetarias), retira la opacidad a los billetes y está lista para entregárselos al terminal del punto de venta.

Como veremos más adelante, el Vendedor, con la colaboración del terminal del punto de venta (con quien colabora la tarjeta inteligente del Vendedor), estará encantado de recibir estos billetes que puede reconocer como válidos, según detallaremos más adelante, verificando las firmas. Pero, ¿y el Banco? ¿Aceptará esta forma de proceder? Indudablemente, sí. Porque billete que firma, dinero que le descuenta al Cliente. Si la tarjeta del Cliente quiere engañar al Banco y en lugar de rellenar uno de los billetes en blanco (los NIBs) generados por la TTP le envía un número inventado, el Banco lo va a firmar igual (no puede ver el identificador) y lo va a descontar igual, pero será un billete inválido que el Cliente no podrá circular satisfactoriamente. Es decir, será una ganancia neta para el Banco porque le descuenta el dinero al Cliente y nadie podrá cobrar después ese «billete».

En el Sistema2, en cambio, la *autorización* (la cantidad de dinero que le va a pedir al Banco) va a llevar un número identificador que lo va a poner la propia tarjeta. Si le mandase al Banco la autorización opacada (de forma equivalente al punto 2 de la anterior secuencia) indicándole en claro el valor de esa cantidad de dinero, el Banco se negaría a firmar porque puede sospechar que la cantidad opacada sea superior a la cantidad de la que se le informa en claro.

¿Cómo convencerle? Una forma de hacerlo es asegurar que tanto la tarjeta del Cliente como el terminal del punto de venta están diseñados a prueba de manipulaciones (*tamper-proof*), por lo que la tarjeta va a seguir el algoritmo que tiene programado y, además, ninguna tarjeta puede ser falsificada y ninguna tarjeta falsa va a ser reconocida por el terminal. Este es un buen argumento que también vale para el caso del Sistema1, pero sería conveniente que introdujésemos un elemento primario de confianza basado en el protocolo seguido y en los mecanismos criptográficos.

* Si el contrato establecido entre el Cliente y el Banco es en forma de débito o pago inmediato, lo descuenta directamente de su cuenta corriente. Si el contrato es en forma de crédito lo descuenta de la cantidad de pago aplazado que tenga pactado y anota la operación, que le será facturada a fin de mes o en la fecha previamente acordada. En lo sucesivo, para simplificar, diremos que «lo descuenta» queriendo significar ambos casos.

Una forma de proceder podría ser la siguiente:

1. La tarjeta TPA genera n candidatos de autorización, todos ellos con la misma cantidad de dinero pero cada uno de ellos con un número de identificación (NIA) distinto.
2. La tarjeta incorpora un factor de opacidad distinto en cada candidato a autorización, y se los envía al Banco, indicándole, en claro, el montante de la cantidad pedida.
3. El Banco, que no puede ver ninguno de los candidatos opacados, selecciona al azar $n-1$ de ellos y se los devuelve a la tarjeta del Cliente, apartando uno de los candidatos recibidos.
4. La tarjeta del Cliente (TPA) le envía al Banco los factores de opacidad que utilizó para oscurecer esos $n-1$ candidatos.
5. El Banco retira la opacidad a esos $n-1$ candidatos y comprueba que todos ellos estaban generados con la misma cantidad de dinero (la que aparece en claro en la petición). Entonces, el Banco se fía de que el candidato restante también está generado por esa misma cantidad. El Banco comprueba la disponibilidad de fondos del Cliente y realiza una firma a ciegas sobre el candidato que permanece opaco. Descuenta el dinero de la cuenta del Cliente y le envía la autorización firmada.
6. La tarjeta, tras verificar que lo recibido se corresponde con su petición (dispone de la clave pública del Banco), retira el factor de opacidad de la autorización y está lista para entregársela al terminal del punto de venta.

A cambio de no usar una TTP hay que soportar el peso de un protocolo más complejo y con más intercambios de datos. Puede quedar la duda, razonable a primera vista, de que siempre existirá la probabilidad de que el Cliente engañe al Banco rellenando $n-1$ candidatos con una cantidad pequeña y el restante con una cantidad sustancialmente mayor. Si tiene la suerte de que es ese candidato el que el Banco aparta, el fraude está servido. Según esto, para reducir esa probabilidad habría que elegir un valor de n razonablemente grande, con la carga que eso acarrea.

Analizado desde un punto de vista estrictamente «protocolario» así parece que son las cosas, pero una reflexión más detenida ayuda a matizar la cuestión. En efecto, supongamos que n es bastante pequeño, por ejemplo 5, lo que supone que existe un 20 por 100 de posibilidades de perpetrar el fraude. Pero aquí no estamos pensando en juegos criptográficos sino en sistemas que cumplan con las exigencias de lo que en el Capítulo 1 hemos caracterizamos como Seguridad Cívica, esto es, sistemas que sirvan para que los ciudadanos puedan tener en la Red razonablemente protegidos sus derechos y garantizada su privacidad. En el caso que nos ocupa, para poder realizar compras sin ser vigilado y clasificado.

Por eso, caso de implementarse este hipotético Sistema2 se haría limitando el valor máximo de las transacciones⁹. Entonces, caso de que un ciudadano se afilie a él y pague la correspondiente cuota de alta y desee usarlo regularmente y por mucho tiempo, el jueguito del fraude le costaría caro. Le costaría lo que cuesta falsificar la tarjeta (muchísimo) y la suerte le duraría poco porque la ruleta rusa del 20 por 100 iría estrechando los márgenes de la suerte cada vez que pruebe con una compra nueva, y la primera vez que sea descubierto se le expulsaría del Sistema2 para los restos, y se quedaría con su pequeño fraude, y perdería su cuota de afiliación, y se quedaría sin hacer lo que realmente quería: hacer pagos de forma anónima para que no lo vigilen los poderes económicos.

En resumen, tanto en el esquema de comportamiento del Sistema1 como en el del Sistema2, como en muchos otros existentes, es factible, con ingenio y estableciendo garantías, convencer al Banco para que firme un pagaré cuyo contenido no ve.

Una última cosa hay que comentar acerca de los sistemas que, como el Sistema2, no hacen uso de una TTP. Si el mecanismo de firma opaca es muy robusto y se sospecha que se está produciendo un uso del sistema para fines delictivos, no existe manera de descubrir al infractor. Para descubrir al infractor puede usarse un esquema de firma a ciegas arbitrada por una entidad juez como el representado en la Figura 11.5. En este caso se podría incorporar también una TTP al Sistema2 que no intervendría en el proceso normal del protocolo salvo que por decisión judicial (o bajo normas pactadas entre todos los afiliados al sistema) se decidiese romper el anonimato de la firma.

EL PROBLEMA DE LA REUTILIZACIÓN DEL DINERO

Uno de los problemas inherentes a los sistemas de pago con dinero digital es que pueda existir la posibilidad de utilizarlo más de una vez. No se olvide que este tipo de dinero está representado, de una u otra forma, en una pieza de información que se puede copiar cuantas veces se desee. A tenor de lo descrito hasta aquí, tanto el Sistema1 como el Sistema2 previenen de este doble uso aunque con alguna deficiencia.

b) Hay que evitar que el dinero firmado se use más de una vez

Veamos en primer lugar cómo continuaría el proceso de entrega de billetes en el Sistema1 una vez que se han recibido correctamente firmados en la tarjeta del Cliente (TPA). El proceso seguiría así:

- a) La tarjeta, una vez comprobada la validez de los billetes recibidos, selecciona los billetes necesarios para satisfacer el precio de la mercancía que está adquiriendo y se los entrega al Vendedor a través del terminal del punto de venta.
- b) El terminal hace una primera comprobación del formato de los billetes verificando la corrección de las firmas del Banco. No puede saber si los identificadores de billete (NIBs) son correctos antes de consultarlo con el Banco.
- c) Si las anteriores comprobaciones resultan favorables, el terminal envía al Banco una solicitud de ingreso de los billetes que ha recibido del Comprador.
- d) El Banco verifica la validez de los billetes comprobando que han sido firmados con las claves privadas monetarias correspondientes y que los NIBs están en la lista de NIBs válidos remitida por la TTP **y que no han sido usados previamente**. Si todo es correcto, los marca como usados con la identidad del Vendedor y transfiere a su cuenta corriente la cantidad total resultante. A continuación informa al terminal del punto de venta del resultado de la operación.
- e) Si el resultado es favorable, el terminal avisa al Vendedor para que entregue la mercancía al Cliente y le entrega a la tarjeta TPA un comprobante firmado de la operación, dando por finalizado el proceso.

Es decir, el Banco detectará claramente el intento de cobro indebido de un billete usado pero no sabe si es que ha sido el Cliente (manipulando su tarjeta) el que ha

querido pagar dos veces con los mismos billetes o ha sido el Vendedor el que (manipulando el sistema del punto de ventas del Sistema1) ha conseguido rescatar billetes usados.

El comportamiento del Sistema2 para la entrega del dinero desde el Cliente al Vendedor es bastante similar al descrito para el Sistema1, y vamos a obviar su descripción. En este caso, en el paso equivalente al punto d) anterior, el Banco una vez que ha hecho el ingreso al Vendedor por importe de la autorización que le ha presentado, lo que hace es grabar el número de identificación de esa autorización usada en una lista. Y lo primero que hace cuando el terminal del punto de venta le presenta una autorización es cotejar su identificador para comprobar que no había sido utilizada anteriormente. Por esta razón, igual que el Sistema1, detectará el intento de doble uso, pero, también al igual que el Sistema1, no sabrá quién ha sido el responsable del intento.

Una mejora del Sistema1 podría consistir en que cuando la TTP genere los identificadores de billetes (los billetes en blanco) les añada una cabecera con una fecha de caducidad. Eso permitiría al terminal del punto de venta almacenar en una base de datos los identificadores que ha recibido con anterioridad para detectar si el Cliente quiere endosarle de nuevo un billete usado en ese mismo establecimiento. Para que esta base de datos pueda tener un tamaño manejable, el periodo de validez de los billetes no debe ser muy amplio (el cliente puede obtener nuevos NIBs de la TTP cuando le haga falta). De esa manera, todos los billetes pagados pero pasados de fecha pueden ser borrados de esa base de datos.

Conforme a esto, el comportamiento descrito anteriormente en el punto b), pasaría a ser el siguiente:

- b) El terminal hace una primera comprobación del formato de los billetes verificando la corrección de las firmas del Banco y comprobando las fechas de caducidad. Comprueba además, consultando su base de datos, que esos NIBs no han sido ingresados antes en este terminal de punto de venta. No puede saber si los identificadores de billete (NIBs) son correctos antes de consultarlo con el Banco.

De esta forma, el Vendedor puede detectar si el Cliente trata de cometer fraude y rechazaría la venta, aunque no puede identificar al presunto estafador (aunque el terminal del punto de venta sí puede quedarse con su tarjeta y no devolverla hasta que se den determinadas circunstancias previamente pactadas). Por eso, si el que consigue enviar al Banco un billete usado es el mismo Vendedor a través de su terminal, el Banco detectaría la duplicación y le achacaría el intento de fraude. Si un Cliente consiguiese entregar un billete ya usado anteriormente en otro terminal de ventas distinto, el proceso prosperaría hasta llegar al Banco, que detectaría que había sido ya cobrado por otro comerciante, con la posible punición del Cliente que ello conllevaría.

Para trasladar una mejora equivalente al Sistema2 podría exigirse que cuando la tarjeta del Cliente genere el número de identificación de la autorización, le añada una cabecera con la fecha del día en que se hace la solicitud al Banco. De esa forma puede establecerse un periodo de validez limitado para las autorizaciones firmadas. También en este caso se dotaría al terminal de venta de una base de datos donde almacenase las autorizaciones que ha tramitado y las borrarse cuando hayan caducado. Gracias a ello, de forma similar a lo que hace el Sistema1, el terminal podrá detectar el intento de doble uso (si ha sido en su mismo establecimiento) y el Banco, además de detectarlo, podrá atribuir al Cliente o al Vendedor la responsabilidad del intento.

EL PROBLEMA DE LA RASTREABILIDAD

Tal y como hemos descrito hasta ahora, del comportamiento tanto del Sistema1 como del Sistema2 se podrían establecer relaciones entre las operaciones de obtención de dinero (billetes en un caso, autorizaciones en el otro) y el pago de esa cantidad al Vendedor (recuérdese el ejemplo del pago de 50,12 euros en una gasolinera, que comentamos en el apartado *Características del dinero digital anónimo* al hablar de la *irrastreabilidad*).

c) Hay que evitar que los pagos sean rastreados

Para evitar ese riesgo, la solución, tanto en el Sistema1 como en el Sistema2, es no pedir nunca al Banco la cantidad exacta que se necesita pagar al Vendedor. Para eso la tarjeta del Cliente tiene que tener capacidad de almacenar dinero ya firmado y calcular qué cantidad debe solicitar para completar la suma requerida. En el Sistema2 lo que necesita la tarjeta es disponer de autorizaciones firmadas por valores intermedios y, en el momento de la compra, solicitar la cantidad necesaria para completar la cantidad exacta que tiene que entregarle al Vendedor.

En el Sistema1 esta operación es más fácil de llevar a cabo porque el dinero lo tiene dividido en billetes digitales. El algoritmo para calcular los billetes que la tarjeta necesita solicitar al Banco se ha denominado *Algoritmo de Redondeo Inteligente* y consiste en determinar, contando con el resto que ya tiene, una cantidad de billetes que exceda en algo a la que realmente necesita. De esta forma, cuando el Vendedor ingrese en el Banco los billetes digitales que ha recibido, este importe nunca coincidirá con el solicitado por la tarjeta del Cliente, evitando así el cotejo de datos. Además, el disponer de un ligero saldo de billetes almacenados en la tarjeta puede dar lugar a que en operaciones de poco valor no sea necesario establecer comunicación con el Banco. Naturalmente, para incluir estos redondeos que evitan la rastreabilidad sería necesario cambiar el protocolo antes descrito, incluyendo nuevos pasos y modificando algunos otros.

d) Es necesario especificar muchos otros detalles

De lo antes descrito hemos podido conocer los elementos principales que están presentes tanto en Sistema1 como en Sistema2 para conseguir el anonimato de las operaciones de compra realizadas con la tarjeta de Cliente. Pero quedan muchos otros por aclarar. Por ejemplo, en el Sistema1 quedarían por discutir muchas otras cosas como puede ser la gestión de los identificadores, el almacenamiento de billetes para que se comporte como una tarjeta monedero, la posibilidad de obtener billetes en puntos específicos de carga además de los terminales dependientes de los vendedores, los mecanismos para evitar que el Banco conozca la dirección telemática desde la que se conecta la tarjeta del Cliente, y un largo etcétera.

Otro tanto podría decirse del que hemos denominado Sistema2 y de cualquier otro de los muchos sistemas que pudiésemos analizar. Además, quedaría por determinar las condiciones de seguridad (confidencialidad, integridad, etc.) en las que se lleve a cabo el intercambio de las PDUs que conforman los protocolos telemáticos que se establezcan a través de la red.

No obstante estas carencias, en los párrafos anteriores sí se han tocado los principales temas y los principales problemas presentes en el amplio y emergente asunto

de las transacciones comerciales de compraventa a través de redes telemáticas, en las que se considera conveniente mantener el anonimato del comprador para garantizar su privacidad.

11.5. VOTACIÓN TELEMÁTICA

Entre los pasos que se están dando de cara a la edificación paulatina de la Sociedad de la Información ocupan un lugar importante las propuestas relacionadas con procesos de votación a través de redes telemáticas. No es esta una tarea fácil porque los sistemas tradicionales de votación mediante papeletas, aunque adolecen de serios inconvenientes en cuanto a rapidez y flexibilidad, sí ofrecen unos mecanismos de salvaguarda y unos procedimientos organizativos muy seguros, que gozan, en general, de una valoración muy positiva por parte de los ciudadanos, lo que redundará en un nivel de confianza bastante alto. En su plasmación en entornos telemáticos será necesario no sólo emular las ventajas de estos sistemas sino superar tanto la seguridad como las prestaciones que pueden obtenerse. Y esto, desde un punto de vista tecnológico, no es tarea fácil.

Si bien es cierto que la existencia de votaciones no implica automáticamente la presencia de Democracia, sí que podemos afirmar que para la contribución a la Democracia mediante sistemas telemáticos (*Democracia Digital*) es imprescindible disponer de estructuras que posibiliten el voto en su doble acepción de elección entre alternativas predeterminadas y de participación de los ciudadanos en las decisiones colectivas. La primera de ellas, a imitación de los esquemas de votación contruidos en los últimos siglos, consiste en la elección de representantes o en la decisión entre opciones alternativas previamente planteadas. La segunda servirá para dar soporte a modelos en los cuales los miembros de una comunidad estén capacitados para proponer, discutir y consensuar alternativas que afecten a la gestión y organización de recursos que les son comunes (ya sean de carácter político, económico, culturales, etc.).

El presente epígrafe y el siguiente estarán orientados al análisis y posibles soluciones de sistemas que den satisfacción a la primera de las dos acepciones de votación planteadas en el párrafo anterior, que es la que aparece más insistentemente en las propuestas planteadas actualmente (quizás por su carácter emulativo de procesos convencionales). La segunda de ellas será discutida de forma genérica en el epígrafe 11.8, al hablar de las plataformas que facilitan la Democracia Digital.

DEL VOTO ELECTRÓNICO AL VOTO TELEMÁTICO

Aunque las primeras referencias al término *voto electrónico* proceden de mediados de los años sesenta, que fue cuando por primera vez se utilizaron computadores para realizar ciertas tareas relacionadas con procesos electorales, ha sido en la década de los noventa cuando con más insistencia han aparecido propuestas y se han llevado a cabo experiencias que han merecido tal calificativo. No obstante, el término *voto electrónico* ha venido empleándose para identificar sistemas de votación de naturaleza muy diversa en los que las garantías de seguridad requeridas en los procedimientos de autenticación, emisión del voto y recuento son proporcionadas de muy diferentes formas¹⁰. Con objeto de clarificar adecuadamente toda esta gama diversa de sistemas, conviene hacer una tipificación [GoCa03] que nos permita distinguir las características principales de cada uno de ellos.

Conforme a esta clasificación, en el **primer nivel** estaría lo que podemos denominar el *escenario clásico* de votación con algunos elementos automatizados. En este escenario se englobarían tanto las votaciones mediante papeletas como aquellas que se sirven de tarjetas perforadas o de lectores ópticos. Estas experiencias no pueden ser consideradas como un sistema de voto electrónico propiamente dicho, pero hasta ahora han sido una referencia para los distintos escenarios electrónicos que se han propuesto.

En un **segundo nivel** se encontrarían los escenarios de votación que, basándose en la forma de operar del método convencional mediante papeletas, sustituyen alguno de sus elementos físicos y procedimientos manuales por algún tipo de sistema o de proceso electrónico. Estos sistemas serían los que podemos denominar propiamente como sistemas de *voto electrónico*. Entre estos posibles sistemas tenemos aquellos que utilizan alguno o varios de los siguientes elementos: tarjetas magnéticas (para autenticar al votante o incluso para emitir el voto), urna electrónica (para la recepción y recuento de votos), pantalla (tablero) de votación (para seleccionar la opción de voto elegida), cabina electrónica (para depositar el voto), computadores con programas de distintos tipos (para el proceso de escrutinio).

En todos estos escenarios, de lo que se trata es de automatizar alguno de los procesos que se llevan a cabo en la votación convencional mediante papeleta. Podemos sintetizarlos en tres: el primero es el de la autenticación del votante, el segundo es el proceso de votar propiamente dicho y el tercero, todo lo relativo a la gestión y procesado del contenido de la urna electoral. Casi la totalidad de las experiencias y actuaciones gubernamentales encaminadas a la automatización de los procesos de votación en los distintos países democráticos se encuadran en este nivel.

Un **tercer nivel** más avanzado en la automatización del proceso de votación sería el determinado por los sistemas de votación que hacen uso de las redes telemáticas y de agentes telemáticos a los que se accede de forma remota. Al voto llevado a cabo bajo estos escenarios es al que podríamos denominar como *voto telemático*. De esta manera, dejando el término de *voto electrónico* para designar a los sistemas que hemos clasificado en el segundo nivel, ganamos en claridad y en precisión. En los sistemas de voto telemático, la urna no se encuentra a la vista del votante (caso del voto electrónico antes citado), sino que es un agente telemático ubicado físicamente en un lugar remoto, al igual que el resto de los agentes que intervienen en la supervisión del sistema.

Aquí podríamos distinguir dos grupos: aquellos que utilizan las redes telemáticas (públicas o privadas) para la interconexión de los distintos colegios electorales, o bien los que proponen la votación desde casa (normalmente a través de Internet). En los escenarios del primer grupo, el elector tiene que desplazarse hasta lugares predeterminados (equivalentes a colegios electorales) para emitir su voto. El uso de redes telemáticas para la interconexión de esos «colegios electorales» por parte de un organismo encargado de la supervisión final (con un papel equivalente al que en España desempeña la Junta Electoral Central) permitiría una rápida recolección de los datos y la publicación de los resultados.

El segundo grupo, **votación desde casa** a través de Internet, es el más atractivo desde un punto de vista tecnológico, debido a los retos técnicos y de seguridad que plantea (venta de votos, coacción, monitorización clandestina, denegación abusiva del derecho a voto y entrega de resultados finales oficiales distintos de los verdaderos). Pero, a su vez, desde un punto de vista sociopolítico, plantea serios interrogantes debidos, en gran parte, al problema que representa el que no todo el mundo tenga las mismas oportunidades de acceso.

Para que las propuestas de sistemas de voto telemático lleguen a tener aceptación por parte de los ciudadanos deberán, al menos, ofrecer las mismas garantías que nos

brinda el sistema tradicional de voto, que, entre otras cosas, permite llevar a cabo un recuento visible de los votos y una revisión manual del proceso. No obstante, a la hora de diseñar nuevos esquemas de votación con soporte telemático es conveniente no limitarse simplemente a emular los procesos convencionales ni ceñirse a sus restricciones, sino aprovechar los recursos que nos brindan las nuevas tecnologías para conseguir sistemas aún más seguros, robustos y flexibles que sus precursores. De la determinación de los requisitos que deberían cumplir los sistemas de este tipo es de lo que trata el siguiente apartado.

DETERMINACIÓN DE REQUISITOS PARA LA VOTACIÓN TELEMÁTICA

En la Figura 11.15 se propone una estrategia para determinar adecuadamente cuáles son los requisitos necesarios para que un sistema de voto telemático cumpla con las demandas sociales, políticas y jurídicas exigibles a unos sistemas que pretenden cubrir actividades tan sensibles para la democracia como son aquellas que sirven para que los ciudadanos ejerzan su derecho al voto.

Huyendo de posturas tecnocéntricas, tan fáciles de encontrar en esta como en otras actividades relacionadas con el diseño de sistemas para la Sociedad de la Información, se propone aquí la colaboración multidisciplinar con el objetivo conjunto de diseñar sistemas que satisfagan las necesidades sociales de los ciudadanos y garanticen los derechos democráticos jurídicamente reconocidos.

En primer lugar se propone una actividad, a la que hemos denominado *Análisis teórico*, en la que se debería analizar qué se debe y qué se puede hacer desde un punto de vista tanto tecnológico como jurídico y sociopolítico. Para ello se partirá de la demanda concreta bajo la que surge el sistema que se quiere desarrollar y de los conocimientos previos que poseen las personas que llevan a cabo ese análisis. Como resultado, se deberá obtener una lista de requisitos exigibles al sistema. En esta actividad se deberán evaluar las posibilidades que ofrecen los protocolos y los mecanismos de seguridad en redes, así como los condicionantes jurídicos existentes (o las modificaciones legales que conllevaría asumir ciertos requisitos) y la determinación sociológica (preferentemente a través de estudios de campo) de las expectativas y temores que tiene la ciudadanía en relación con la implantación de este tipo de sistemas de votación.

En segundo lugar, a través de una *Investigación social*, esos requisitos previamente detectados deberían ser evaluados de cara a detectar la aceptación y la usabilidad del sistema. Ello permitiría matizar los requisitos previamente detectados, encontrar otros nuevos y evaluar la importancia relativa que la ciudadanía otorga a cada uno de ellos.

Teniendo en cuenta los requisitos así detectados y evaluados, se reduce el riesgo de que los ciudadanos presenten rechazos y temores una vez que se haya diseñado el sistema. En la Figura 11.15 se ha querido representar que la fase de diseño puede dar comienzo una vez se haya obtenido una primera relación de requisitos, aunque posteriormente deban ser tenidas en cuenta las mejoras y matizaciones derivadas de la investigación sociológica.

Por último, en el diagrama que estamos comentando, se ha recogido una fase de *Evaluación*, que preferentemente debería ser una demostración práctica del sistema con votantes reales, de la que se deduzcan defectos y se exploren nuevas posibilidades que deben ser incorporadas, de forma que se retoquen los requisitos que marcaron el diseño inicial, todo ello con objeto de obtener un sistema final más robusto y con mayor aceptación por parte de los ciudadanos.

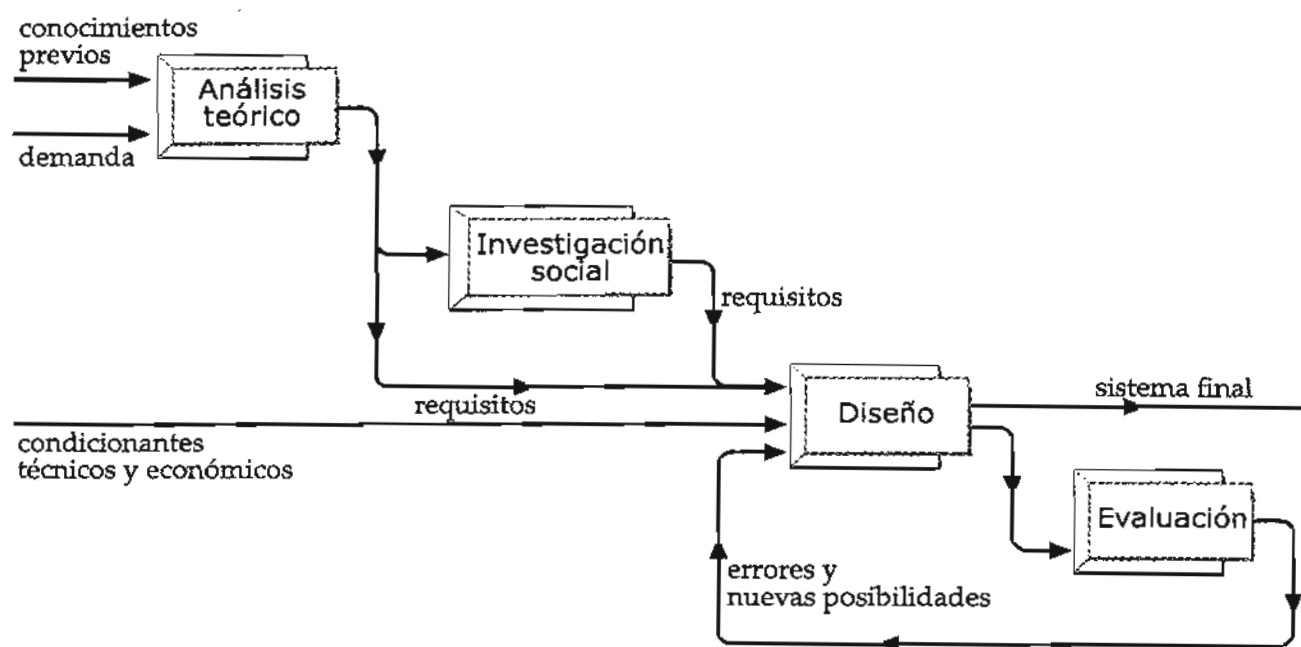


Figura 11.15. Determinación de requisitos para el diseño de un sistema de voto telemático.

Después de todo esto, es posible obtener una relación de requisitos que garantice, con bastante aproximación, que el sistema que los satisfaga cumplirá adecuadamente con todas las exigencias existentes y será bien aceptado por los ciudadanos. Apoyándonos en los resultados de una experiencia práctica realizada¹¹ bajo un esquema similar al recogido en la Figura 11.15, expondremos a continuación una *relación de requisitos* [CGMP02] exigibles a los sistemas de votación telemática. Es la siguiente:

1. **Autenticidad** o **Autenticación**. Sólo los votantes autorizados pueden votar.
2. **Acotabilidad** o **Singularidad** (*Uniqueness*). El sistema tan sólo autentica la votación dentro de las reglas establecidas. Es decir, por regla general, cada votante sólo puede votar una vez.
3. **Anonimato**. No se puede relacionar un voto con el votante que lo ha emitido.
4. **Imposibilidad de coacción**. Ningún votante debe ser capaz de demostrar ante terceros qué voto ha emitido.

5. **Verificabilidad individual**. Cada votante deberá poder asegurarse de que su voto ha sido considerado adecuadamente, de forma que pueda obtener una prueba palpable de este hecho. Podemos distinguir dos tipos de verificabilidad individual:

- Verificabilidad individual del contenido del voto emitido.
- Verificabilidad individual de que el voto ha sido tenido en cuenta adecuadamente. Se trataría de una prueba menos contundente que la requerida en el caso anterior. Si bien no puede conocer con puntualidad cuál de los votos contabilizados es el suyo, al menos deberá tener pruebas de que ha sido tenido en cuenta y, confiando en la corrección del sistema, confiar que lo haya sido a favor de la opción elegida (esto es lo que ocurre en la votación convencional mediante papeletas).

Definida de esta forma, en la verificabilidad individual puede aparecer una cierta contradicción con el requisito de imposibilidad de coacción. Cuanto más explícita es la verificación individual más riesgos de coacción pueden aparecer. En el sistema

convencional, el votante sabe lo que vota, y confía en que su voto será contabilizado correctamente cuando comprueba que es introducido en la urna. Además, si usa una cabina para cumplimentar su voto, no hay peligro evidente de coacción. Como puede intuirse, un estudio mínimamente riguroso del balance entre los requisitos de verificabilidad y coacción requeriría la inclusión y análisis de más parámetros, dependiendo de los distintos condicionantes sociales que aparecen en cada situación concreta.

6. **Verificabilidad global.** Otra forma mediante la cual el votante puede asegurarse de que su voto ha sido considerado adecuadamente es que dentro del propio sistema existan mecanismos que permitan a los ciudadanos autorizados comprobar la validez del recuento final.

7. **Fiabilidad.** El sistema debe garantizar que no se produce ninguna alteración de los resultados, ya sea mediante ataques intencionados, fallos en el sistema o incluso si las autoridades del sistema se ponen de acuerdo para coludir.

- Fiabilidad en los procedimientos (*Reliability*). El sistema debe trabajar de forma robusta, incluso en el caso de numerosos fallos, incluyendo fallos masivos en las máquinas de votación o pérdida total de las comunicaciones.
- Exactitud en el recuento (*Accuracy*). El sistema debe registrar correctamente todos los votos. Todos los votos son tenidos en cuenta sin que sea posible cambiar, borrar o extraviar ningún voto. El sistema ha de proporcionar 100 por 100 de exactitud.
- Integridad de los datos (*Data Integrity*). Se garantiza que el contenido del voto u opinión es exactamente el que fue enviado, de tal forma que al texto original no le ha sido añadida, ni modificada ni sustraída alguna de sus partes.

8. **Certificabilidad o Auditabilidad.** Durante el proceso de votación deberían registrarse las pruebas de voto y elementos de auditoría que permitieran a las personas autorizadas disponer de pruebas para comprobar que todo el proceso de votación es correcto (funcionamiento del sistema, programas, equipos, protocolos y demás elementos), todo ello sin comprometer la integridad de la elección o la privacidad y anonimato de los votantes. Se pueden distinguir dos tipos de auditabilidad:

- Auditabilidad al desarrollo y ejecución del sistema durante el proceso de votación.
- Auditabilidad global del sistema utilizando pruebas obtenidas durante el proceso de votación. Estas pruebas o registros deberán ser físicamente almacenables, recuperables y comparables una vez haya terminado el proceso de recuento y generación de resultados. Este requisito se puede considerar en sus objetivos bastante coincidente con el requisito de *verificabilidad* global, reseñado con el número 6.

9. **Neutralidad.** No debe ser posible conocer resultados parciales hasta que no finalice el tiempo de la elección, para evitar que puedan influir en la libre decisión de los votantes.

10. **Movilidad de los votantes.** El sistema debería permitir a los participantes que emitieran su opinión o voto desde cualquier *cabina o punto de votación*, eliminando la restricción actual de hacerlo en el centro de votación de la zona en la cual están censados.

11. **Facilidad de uso.** El votante debe necesitar el mínimo de habilidades y conocimientos especiales para emitir el voto. De esa manera, se propicia la igualdad de oportunidades de todos los votantes a la hora de emitir su opinión.

12. **Voto rápido.** El votante debería poder emitir el voto en un tiempo mínimo y razonable.

13. **Voto nulo o de rechazo.** El sistema de votación telemática debería posibilitar que se emitiera un voto sin que fuese contabilizado como válido para ninguna de las candidaturas propuestas ni ser considerado dentro del bloque de los votos en blanco. En la votación convencional mediante papeletas existe esta alternativa, que puede usarse como muestra de rechazo ante la oferta de opciones que se presenta ante el votante.

14. **Código abierto.** El código fuente de todos los programas que gobiernan el sistema debería ser conocido y verificable por los auditores que actúen en representación de los electores. La seguridad del sistema no debería estar basada, por tanto, en mantener este código secreto, sino en la fortaleza de los protocolos y de las claves de cifrado utilizadas en todas las fases del proceso de votación. Para garantizar el carácter abierto de los programas, el modelo de licencia más conveniente sería el comúnmente denominado *copyleft*.

15. **Coste mínimo.** El coste del sistema de elección debería ser abordable y estar en consonancia con el coste de los sistemas convencionales mediante papeletas (o ser aún menor).

16. **Utilización de una red dedicada.** Tanto si se vota a través de Internet como si se vota desde cabinas especializadas, la red telemática en la que se apoye el sistema deberá ser, desde un punto de vista lógico, totalmente cerrada, de forma que el acceso a ella sólo esté permitido a los agentes y actores contemplados en el sistema.

17. **Compatibilidad con otros mecanismos de votación convencionales.** La implantación de los nuevos sistemas de votación telemática deberá hacerse de forma gradual, permitiendo que los ciudadanos puedan elegir entre este y el sistema tradicional mediante papeletas con «urna a la vista» (o voto por correo allí donde exista esta posibilidad).

18. **Igualdad de oportunidades en la votación.** Todo ciudadano ha de tener acceso al equipamiento técnico y procedimientos organizativos a la hora de votar.

El acceso desde casa, quizás a través de Internet, plantea innumerables ventajas, pero, en la situación actual, conlleva unos riesgos capitales en lo relativo a lo que en sociología se conoce como Estratificación Digital. Se entiende por *Estratificación Digital* (en inglés *Digital Divide*) los trabajos que abarcan el estudio de los discursos y prácticas asociadas con las desigualdades y diferencias en el acceso a computadores, infraestructura de entrada a la red y adquisición de conocimientos, que se dan entre las distintas clases sociales, así como por etnia, género, nivel educativo, convicciones políticas o religiosas, etc.

La preocupación de los ciudadanos por este tipo de desigualdades ha sido fuertemente detectada en las investigaciones sociológicas que se han llevados a cabo. Un sistema de Democracia Digital necesariamente conlleva el derecho de acceso del conjunto de la ciudadanía: sería una proyección telemática del concepto de Sufragio Universal.

19. **Flexibilidad física.** El equipamiento debe disponer de diferentes alternativas que hagan que pueda ser usado por gente con alguna discapacidad física.

20. **Digno de confianza.** Al igual que los ciudadanos entienden, de forma genérica, la estructura y modo de operación del sistema convencional de votación mediante papeletas, deberán entender el proceso de votación telemática para fortalecer su confianza en el sistema y en las personas que lo gobiernan y lo supervisan.

Hasta aquí la relación de los requisitos detectados. Resulta evidente que esta no es una relación canónica sino que utilizando otros métodos de análisis se llegaría a otra relación diferente. Pero para los objetivos que aquí pretendemos nos va a servir como un punto de referencia a la hora de analizar las funcionalidades que

debe soportar un sistema de voto telemático (análisis que realizaremos en el siguiente epígrafe).

En algunos sectores se han realizado objeciones y críticas muy rigurosas de cara a la viabilidad de los sistemas de voto telemático. Las más significativas de ellas fueron recogidas por R. Mercuri* en su intervención en la Cámara de Representantes de Estados Unidos, y pueden resumirse en:

- Que es imposible superar aspectos tan críticos como son el riesgo de venta de votos, coacción, monitorización clandestina y denegación del derecho a voto.
- Que no hay forma de ofrecer al votante la seguridad de que el voto se ha registrado tal cual ha sido emitido, o que el recuento es el correcto.
- Que no ofrece control por parte de los partidos políticos.
- Que desde cualquier lugar del mundo se pueden atacar los sistemas telemáticos.
- Que los defectos del sistema pueden ser conocidos años después de la elección y que no hay elementos de auditoría.
- Que los mecanismos criptográficos se pueden romper tarde o temprano.

Como puede comprobarse, las cinco primeras objeciones que detectó Mercuri en los sistemas que analizaron han sido incluidas como requisitos en la relación antes expuesta, debido a lo cual será obligado que aquellos sistemas que cumplan esos requisitos queden a salvo de esas deficiencias. En cuanto a la última de ellas, será también necesario tenerla en cuenta a la hora de evaluar los procedimientos empleados por los sistemas de votación telemática que se nos presenten.

11.6. ESCENARIOS Y PROTOCOLOS DE VOTACIÓN TELEMÁTICA

Al igual que cuando analizamos los escenarios de dinero digital anónimo dijimos que no describiríamos las distintas aplicaciones existentes ni analizaríamos el estado del arte al respecto, tampoco ahora (por similares razones) abordaremos ni lo uno ni lo otro en lo relativo al voto telemático.

También ahora, para analizar por partes los principales problemas que se presentan en la votación telemática y la búsqueda de soluciones que **satisfagan los requisitos**, usaremos una estrategia pareja a la utilizada para el dinero anónimo: nos apoyaremos en la simplificación de dos propuestas prácticas¹² distintas que vamos a denominar, respectivamente, *SistemaA* y *SistemaB*. Con posterioridad, el lector estará en condiciones de aplicar los criterios obtenidos de ese análisis para evaluar los distintos sistemas de votación telemática que se le presentan en la actualidad y los que se le presentarán en un futuro.

Inicialmente haremos un breve comentario de una propuesta que ha supuesto un hito importante en la configuración de lo que nosotros denominamos voto telemático: se trata del esquema propuesto por Fujioka y otros [FOO93], cuya representación simplificada se muestra en la Figura 11.16, que contempla un escenario de comunicación en el que intervienen agentes telemáticos que se comunican a través de una red. Esta propuesta define distintos agentes participantes en el proceso de votación y pone el énfasis en el cumplimiento de los elementos básicos del proceso electoral tradicional. También hace la aportación de ofrecer al votante la posibilidad de ejercer cierto grado de control en el proceso.

* Mercuri R. *Testimony presented to the U.S. House of Representatives Committee on Science*, mayo 2001. <http://www.house.gov/science/full/may22/mercuri.htm>.

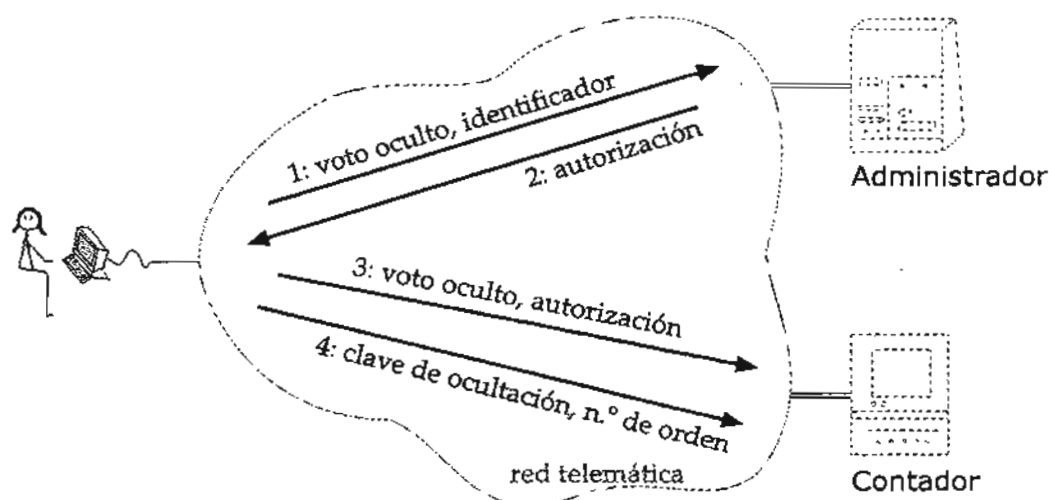


Figura 11.16. Esquema de votación en un entorno telemático (según la propuesta de Fujioka).

La votación se realiza en dos fases separadas en el tiempo. En la primera de ellas el votante selecciona el voto, lo oculta, lo firma con su clave privada y se lo envía al Administrador junto con su identificador personal (paso 1); si todo es correcto, el Administrador lo tacha de la lista y le devuelve una autorización consistente en el voto oculto firmado (paso 2). A continuación el votante envía esa autorización junto con el voto oculto al Contador (paso 3), el cual asigna **un número** a cada entrada que recibe. Cuando termina la votación y todos los votantes han realizado los pasos 1, 2 y 3, termina la primera fase y empieza la segunda. Tanto el Administrador como el Contador publican una lista con las informaciones recibidas. Consultándolas, el votante puede comprobar que su voto se ha tenido en cuenta y, extrayendo **el número** de orden que corresponde a su voto en la lista del Contador, se lo envía de nuevo al Contador (paso 4) junto con la clave que utilizó para ocultar el voto. Una vez que todos los votantes han cumplimentado la segunda fase, el Contador puede realizar el recuento de votos y publicar una lista en la que se muestra el voto en claro y los componentes criptográficos en base a los que se ha obtenido.

Como puede observarse, esta propuesta es muy interesante y muy completa, pero tiene el defecto de que requiere de dos fases diferenciadas (con todos los problemas de comunicación que ello acarrea y la ausencia del requisito de *rapidez*). Además, todo está demasiado a la vista, con lo cual existe el riesgo de *coacción* y *compra de votos*, así como el peligro que representa el hecho de que esas protecciones criptográficas (que ahora son irrompibles) transcurridos unos años dejarán de serlo, y los resultados de una votación pasada podrán ser conocidos de la A a la Z. Es decir, este sistema cumple perfectamente los requisitos elementales de anonimato en la entrega del voto; pero no hay que esforzarse mucho para comprobar, repasando la lista de requisitos que se recoge en el apartado anterior, que no cumple muchos de ellos. Aunque sea un recurso retórico algo manido, merece la pena recordar que una cadena es fuerte si son fuertes todos sus eslabones, pero basta que uno solo sea débil para que eso afecte a la robustez del conjunto.

Por todo ello, a partir de la visión global sobre el voto telemático que podemos extraer de esta propuesta pionera, pasaremos a desglosar las distintas tareas que son necesarias para culminar el proceso con las garantías necesarias.

AUTENTICACIÓN Y AUTORIZACIÓN DEL VOTANTE

En los procesos de votación convencionales, el primer paso que hay que dar es identificarse como votante ante la mesa electoral para así obtener la autorización necesaria para votar. Aunque, como ya hemos dicho, en los procesos de voto telemático no es conveniente limitarse a emular los procesos convencionales ni ceñirse a sus restricciones, valga esta comparación para entender mejor cuál es el objetivo de la fase que estamos analizando.

Es por ello que en los sistemas de voto telemático la primera fase del proceso sea también la de autenticar debidamente al votante y, caso de que sea elector, proporcionarle la correspondiente autorización para votar. Para analizar cómo se lleva a cabo esta fase en los sistemas que vamos a tomar de ejemplo, veamos inicialmente la estructura del primero de ellos: el SistemaA.

En la Figura 11.17 se presenta un esquema reducido del SistemaA en el que puede detectarse la presencia de los siguientes agentes telemáticos:

- *Puntos de Autenticación, PAs.* Se trata de un tipo de cabinas, dotadas de lector de tarjetas pero sin capacidades criptográficas, en las que el Votante inicia el proceso de votación. En la figura aparece uno solamente, pero pueden existir varios.
- *Puntos de Votación, PVs.* Se trata de cabinas, también sin capacidades criptográficas, dotadas de lector de tarjetas que ayudan al Votante a determinar y entregar el voto que desea emitir.
- *Un Administrador.* Es un sistema que sirve para la autenticación de los votantes.
- *Una Urna* que va recogiendo los votos.
- *Un Contador* que realizará el recuento de los votos una vez finalizado el periodo de recepción de los mismos.

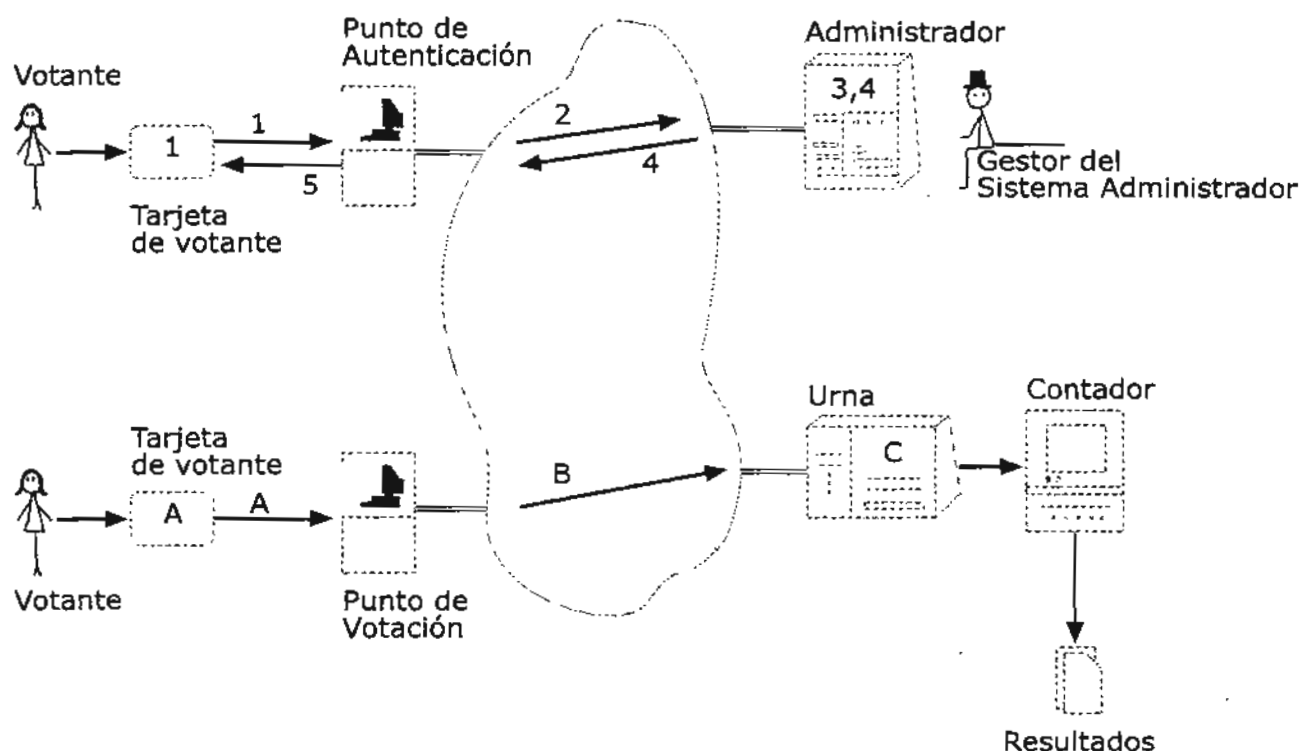


Figura 11.17. Esquema reducido del SistemaA.

Además, el sistema contempla la existencia de un conjunto de personas que intervienen de forma directa en el proceso de votación y recuento. En esta versión reducida son:

- *Votantes*. Cada Votante está en posesión de una Tarjeta inteligente de Votación, TV, que permite llevar a cabo múltiples operaciones criptográficas.
- Un *Gestor del Sistema Administrador* que también estará presente en «la apertura» de la Urna.
- Una *Autoridad de Elección*. Es la persona encargada de la supervisión general del sistema.

a) Hay que autenticar al Votante y autorizarle a votar una sola vez

Esta exigencia (como recordábamos en los requisitos 1 y 2 reseñados en el anterior apartado) es de las más elementales que deben plantearse. Analicemos si se cumple adecuadamente en el SistemaA.

El Votante se acerca a la cabina que contiene el Punto de Autenticación y procede de la siguiente forma:

1. El Votante introduce su tarjeta inteligente (TV) en el lector de tarjetas del PA. Se produce una autenticación mutua entre ambos.
2. El Votante se autentica ante su TV mediante un PIN o un mecanismo de identificación biométrico.

A partir de aquí se lleva a cabo, ante el sistema, el proceso de autenticación propiamente dicho, que podemos describirlo (véase Figura 11.17) en los cinco pasos siguientes:

1. La Tarjeta de Votante, TV, genera una clave de solicitud de autorización, k_A . La TV contiene un par de claves (pública y privada) identificativas del Votante. La tarjeta genera un factor de opacidad y opaca k_A para el Administrador. La TV le entrega al PV la clave opacada $O_{Ad}(k_A)$ y el identificador del Votante, todo ello firmado con su clave privada.
2. El PV genera una APDU con los datos recibidos de la TV y se la envía al Administrador.
3. El Administrador lee la APDU. Luego deberá comprobar si el identificador de votante recibido es correcto. Es decir, comprueba que el identificador está dentro de la lista de identificadores válidos, que la firma del Votante que realiza la solicitud es correcta y que no se ha recibido (y por tanto firmado ya) una clave opacada $O_{Ad}(k_A)$ asociada a dicho identificador (es decir, que esa tarjeta no ha realizado previamente la autenticación). En caso contrario se rechaza lo recibido.
4. Una vez comprobado que el identificador de votante recibido es válido, el Administrador firmará a ciegas la clave opacada de solicitud de autorización, $O_{Ad}(k_A)$. Esta firma opaca $A_{bsig}[O_{Ad}(k_A)]$ representa la autorización que el Administrador emite a favor del Votante.

Todo ello lo volverá a cifrar con la clave pública del Votante, tras lo cual lo enviará (mediante una APDU) al Punto de Autenticación. De esta forma tan sólo la TV podrá leer esa información (confidencialidad de los datos) y además, gracias a la firma opaca, la tarjeta tendrá la garantía de que fue el Administrador quien le devolvió la autorización.

5. El Punto de Autenticación entrega a la Tarjeta del Votante los datos contenidos en la APDU que ha recibido del Administrador. La TV elimina el cifrado que los protege haciendo uso de la clave privada del Votante.

A continuación la TV elimina el factor de opacidad de la autorización recibida obteniendo la clave de autorización k_A firmada por el Administrador [esa firma será $Ad_s(k_A)$ si se ha usado un algoritmo equivalente al de Chaum] (véase el apartado correspondiente a la firma opaca en el epígrafe 11.1). A continuación verifica que la firma del Administrador es correcta. Si es así, el Votante ha finalizado el proceso de autenticación y de obtención de autorización para votar.

Esta k_A **firmada** por el Administrador constituye la verdadera *autorización* para votar.

Visto en su conjunto, el proceso no es muy complicado. Consiste en que el Votante, a través de su tarjeta inteligente, envía su identidad más una clave de solicitud de autorización (k_A) opacada. El Administrador lo tacha de la lista de votantes y le firma de forma ciega la clave opacada que le envió. El Votante (su tarjeta) retira el factor de opacidad y obtiene la k_A firmada por el Administrador. Cualquiera que conozca la clave pública del Administrador podrá verificar que, en efecto, el Administrador ha sido quien le ha firmado esa autorización. El Administrador sabe que el Votante ha obtenido la autorización para votar, pero no tiene ni idea del valor real de esa k_A que ha firmado.

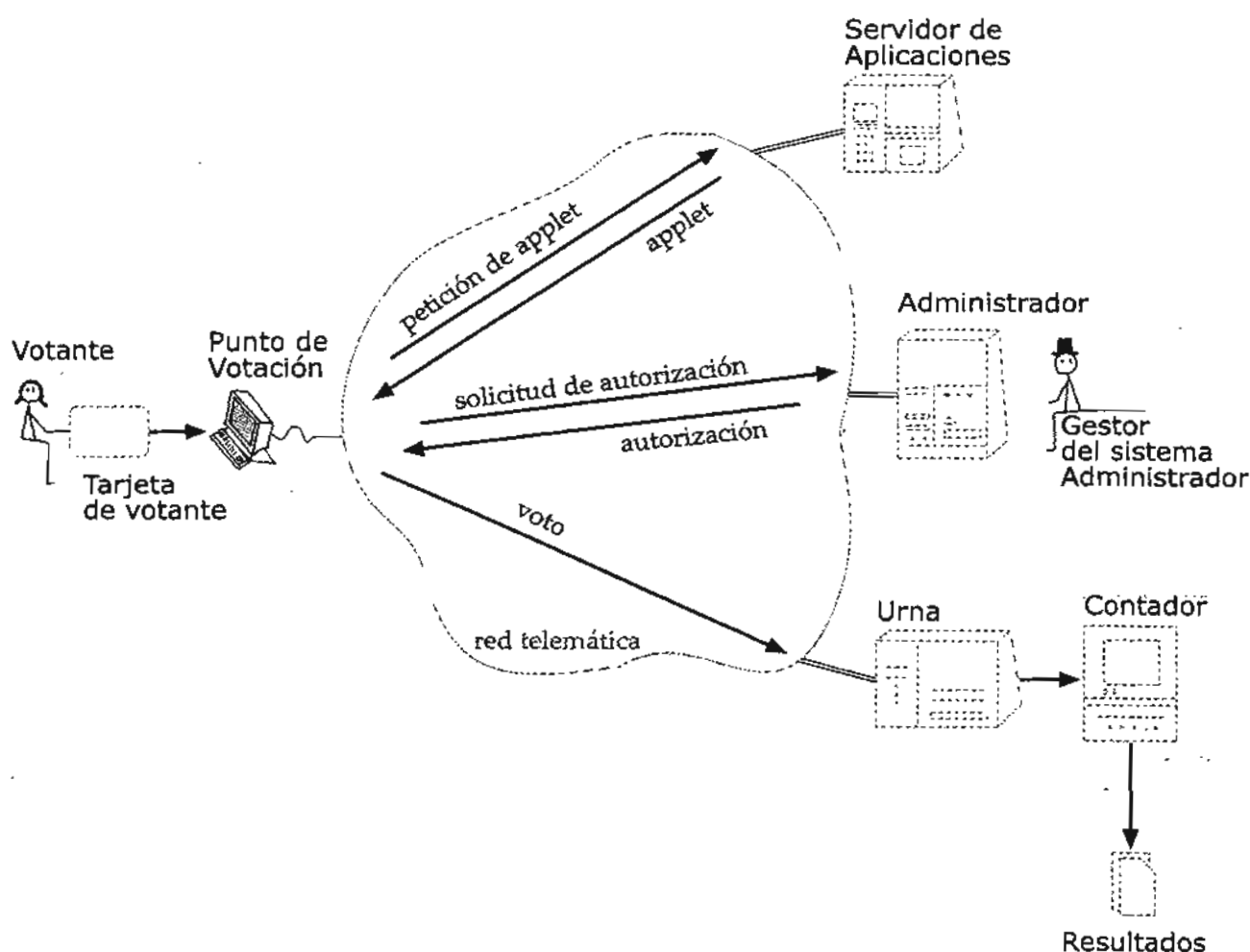


Figura 11.18. Esquema reducido del SistemaB.

Piénsese que aquí, a diferencia de lo que ocurriría con el Banco en la firma a ciegas de pagarés digitales, el Administrador no se juega nada firmando a ciegas. Suponiendo que un votante se saltase la protección de la tarjeta y falsificase los programas y enviara en lugar de k_A un valor mal configurado, lo único que ganaría sería ser tachado de la lista y (como veremos más adelante) no poder votar.

En resumen, todo parece indicar que la exigencia de *autenticidad* y *acotabilidad* que estamos comentando se cumple en el SistemaA. Veamos si ocurre lo mismo con el SistemaB.

Una simplificación de la arquitectura del SistemaA la constituye el hipotético sistema que hemos denominado SistemaB, cuya configuración (a su vez reducida) se muestra en la Figura 11.18. A diferencia del esquema anterior en el que los Puntos de Votación (y los de Autenticación) son máquinas con periféricos y programas específicos, sin capacidades criptográficas, diseñadas exclusivamente para la aplicación de voto, los Puntos de Votación en el SistemaB son computadores personales convencionales dotados de lector de tarjeta y con capacidad de acceso a través de Internet. Además, como se aprecia en la figura, este mismo computador sirve tanto para la autenticación como para la emisión de voto. Así pues, el escenario de comunicación del SistemaB se diferencia del que ya hemos visto para el SistemaA en que:

- Las dos cabinas se sustituyen por un Punto de Votación constituido por un computador convencional conectado a la red telemática.
- Aparece un nuevo agente: el Servidor de Aplicaciones.

La primera parte del proceso de autenticación consiste en que:

- i) El Votante introduce su tarjeta inteligente (TV) en el lector de tarjetas del computador.
- ii) El Votante se autentica ante su TV mediante un PIN (o un mecanismo de identificación biométrico).
- iii) El computador que actúa como Punto de Votación se conecta con el Servidor de Aplicaciones e invoca a la *applet* que reside en él. La *applet* se carga en el computador que actúa como Punto de Votación. A partir de ese momento es esa *applet* o *aplicación cliente*, que sí tiene capacidades criptográficas, la que gobierna todo el proceso de votación.

En el SistemaB la TV tiene menos competencias que las que tiene asignadas la TV del SistemaA. Por esa razón, puede ser una tarjeta inteligente más elemental, en cuyo caso no soportaría adecuadamente los mecanismos de autenticación biométrica, lo que supondría menos fortaleza en la autenticación del Votante. Además, al tratarse de un computador convencional, no es posible ofrecer las mismas garantías de resistencia ante manipulaciones que pueden implementarse en el PA y PV del SistemaA y no puede implantarse el reconocimiento mutuo entre el terminal y la tarjeta, con el riesgo que supone que una tarjeta pirata intente hacer tonterías.

En cuanto a los cinco puntos restantes, el comportamiento es muy similar al que ya hemos visto para el SistemaA salvo que aquí las tareas de cifrado y verificación se reparten entre la tarjeta y la *applet* que reside en el computador de acceso. No obstante, la generación de claves y el firmado con la clave privada del Votante han de realizarse obligatoriamente en la tarjeta inteligente del Votante.

Aunque la *applet* que se carga estará firmada por el Servidor de Aplicaciones para evitar que realice tareas indebidas, siempre será posible crear *applets* ficticias que

averigüen el valor de algunos de los datos que son manejados desde ella. En el SistemaA, sin embargo, todo lo que se procesa dentro de la tarjeta y no sale fuera de ella es computacionalmente impracticable conocerlo.

En resumen, también el SistemaB cumple bastante aceptablemente la exigencia de *autenticidad y acotabilidad* que estamos comentando, aunque de forma menos robusta que el SistemaA. Por contra, el SistemaB tiene un aspecto más atractivo, ágil y «moderno» que el sistema «padre» del que procede.

ENTREGA DEL VOTO A LA URNA

El Votante, una vez que ha completado con éxito el proceso de autenticación y que ha obtenido la autorización, se dirige a un Punto de Votación para proceder a la entrega del voto. Nuevamente cabe recordar el paralelismo que esto tiene con los procesos convencionales de votación: una vez que la persona que va a votar se ha identificado ante la mesa electoral y su nombre es tachado de la lista correspondiente, se le permite que el sobre que contiene su papeleta sea introducido en la Urna.

b) El voto debe entregarse de forma anónima y sin coacciones

Fijemos nuestra atención de nuevo en el SistemaA para analizar si cumple esta exigencia (números 3, 4 y parte del 7, según la relación de requisitos). Antes, cuando contamos que la tarjeta generaba la *clave de solicitud de autorización* (k_A), no contamos toda la verdad. Lo que genera realmente es un par de claves asimétricas de votación (k_{dv} , k_{cv}), tipo RSA, que se almacenan de forma que ni el propio Votante pueda leerlas (las dos son secretas). La *clave de cifrado de voto* (k_{cv}) se utilizará para cifrar el voto en claro y, como su nombre también señala, la *clave de descifrado de voto* (k_{dv}) servirá para leer el voto cifrado. Esta segunda clave k_{dv} es la que se utiliza para solicitar la autorización y es a la que en la fase anterior denominamos k_A . Es decir: $k_A \equiv k_{dv}$. Y la *autorización firmada* (si se usa un algoritmo equivalente al de Chaum) es: $\text{Ad}_S(k_A) \equiv \text{Ad}_S(k_{dv})$.

A partir de ahora, para mejor entendimiento de las operaciones, vamos a utilizar para esta clave el nombre y el símbolo de *clave de descifrado de voto*. Dicho esto, digamos que el Votante se dirige al Punto de Votación e introduce su tarjeta en el lector correspondiente, llevándose a cabo un proceso de autenticación del Votante ante su tarjeta y de la tarjeta ante el PV, que es en todo igual al que ya describimos como preámbulo de la fase de autenticación.

A partir de ahí el proceso se divide en tres pasos (Figura 11.17), de la forma siguiente:

- A) El Punto de Votación solicita al Votante su voto mediante un diálogo interactivo en el que, a través de texto e imágenes, se facilita al Votante la elección de la opción deseada. En la Tarjeta de Votante se cifra con k_{cv} (*clave de cifrado de voto*) el voto a entregar. Ello implica que el voto sólo podrá ser descifrado usando la clave k_{dv} (*clave de descifrado de voto*) pareja de la anterior.

Mediante un proceso de diálogo entre el PV y la Tarjeta, ésta crea una pieza de información con el voto cifrado, la clave k_{dv} y la *autorización firmada* (la clave k_{dv} firmada por el Administrador). A continuación se «guarda»

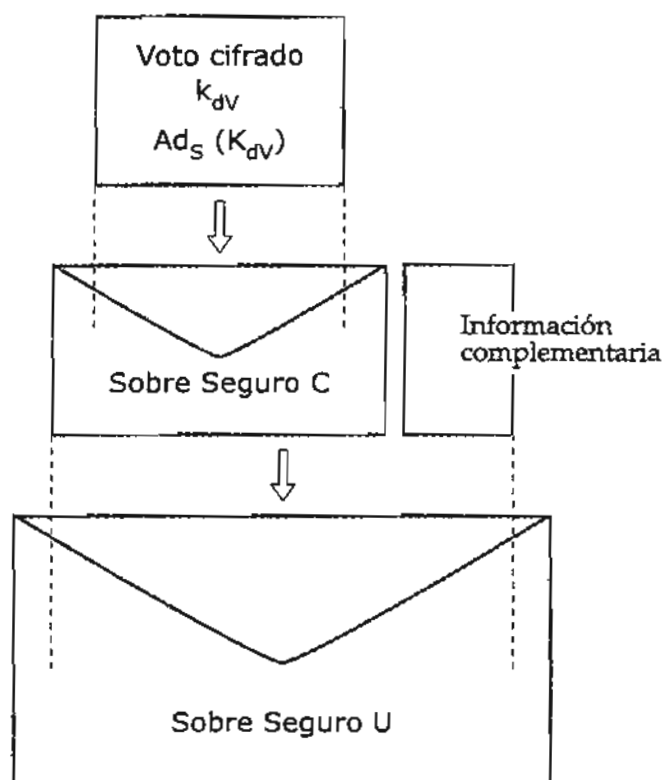


Figura 11.19. *Sobre Seguro C dentro de Sobre Seguro U.*

esta pieza de información en un *Sobre Seguro* * C entre la TV y el Contador, que sólo este último puede abrir¹³. Tras esto, en la Tarjeta del Votante se genera una «información adicional» (cuyo significado no veremos por ahora) que se junta con el *Sobre Seguro C* y se «guarda» en un nuevo *Sobre Seguro U* que sólo la Urna puede abrir. A continuación la Tarjeta le entrega al Punto de Votación este *Sobre Seguro U* para su posterior envío a la Urna. (Véase Figura 11.19).

- B) Con el *Sobre Seguro U* referido en el paso anterior, el Punto de Votación genera una APDU y la envía a la Urna (solamente la Urna sabe leer esa APDU).
- C) La Urna, tras eliminar el *Sobre Seguro U* que protege lo recibido (y que sólo la Urna es capaz de «abrir»), obtiene la «información adicional» (cuyo significado no veremos por ahora) y la pieza de información que constituye el *Sobre Seguro C* (que sólo el Contador puede «abrir» y que contiene, repetimos, el voto cifrado, la clave k_{dv} y la clave k_{dv} firmada por el Administrador). La Urna almacena estos *Sobres Seguros C* hasta el final del periodo de votación. El Punto de Votación informa al Votante de que ha terminado el proceso de entrega de su voto.

* Este *Sobre Seguro* sirve para que el destinatario obtenga el *mensaje* con total garantía de confidencialidad e integridad y sin información alguna que permita asociarlo a su origen. Si únicamente se cifrara el voto con la clave pública del destinatario sería muy sencillo el que un atacante externo adivinase el voto entregado por un votante simplemente cifrando todas las posibles opciones de voto con la clave pública correspondiente y comparando los resultados obtenidos con la pieza de información entregada por el Votante.

Una Tarjeta de Votante que haya completado el proceso anterior y haya emitido el voto almacenará un indicador y rechazará realizar de nuevo el proceso de entrega de voto, aunque su Votante propietario lo intente (la tarjeta es resistente ante manipulaciones), lo que garantiza que cada Votante vota sólo una vez. No obstante, aunque consiguiera romper la protección de la tarjeta no conseguiría que se le contabilizase más de un voto (como más adelante veremos).

A tenor de las explicaciones anteriores, parece bastante claro que el *anonimato* está bastante bien garantizado porque aunque el votante se autentica **ante su tarjeta**, no se autentica ante el Punto de Votación. Además, aunque posteriormente el Administrador vea la relación que hay entre la k_{dv} y el voto, no podrá saber nada de la identidad del Votante propietario de esa clave que el Administrador firmó a ciegas.

En principio, podemos pensar que tal y como está diseñado el proceso podríamos juntar el Punto de Autenticación y el Punto de Votación en una sola cabina. Eso simplificaría las cosas y no parece que tenga mayor problema si pensamos que los terminales son *tamper-resistant* y no van a hacer nada que no esté especificado en su funcionamiento. Pero el tema del anonimato es muy serio y no podemos permitir que sea una misma máquina la que maneje primero la identidad y luego el voto.

Por eso es mejor poner pared por medio y separar ambas máquinas. Así, aunque alguien consiguiese manipularlos o sustituyese un terminal manipulado haciéndolo pasar por bueno, ese Punto de Autenticación puede llegar a conocer la identidad pero no puede relacionarla con la k_{dv} , mientras el Punto de Votación conoce el voto pero no puede conocer la identidad. Además, las dos actuaciones pueden estar distanciadas en el tiempo a voluntad del Votante (puede quedarse deshojando la margarita antes de votar) y en el espacio (puede depositar el voto en otro lugar donde también exista un Punto de Votación). Esta posibilidad de distancia en el tiempo y en el espacio es una de las posibilidades que ofrecen los sistemas de votación telemática (requisito número 10, *movilidad de los votantes*) que no existe en los sistemas de votación mediante papeleta ni en otros de los que hemos denominado de *voto electrónico convencional*. Además, conviene que no exista una relación uno a uno entre las dos máquinas.

Cuanta menos relación, mejor. Así evitaremos que la fila de votantes ante el PV se produzca en el mismo orden que la fila de votantes ante el PA, y evitaremos la tentación de que alguien pueda establecer registros que luego puedan cotejarse. Es sencillo idear varios procedimientos organizativos para romper la relación entre ambas máquinas. Puede pensarse que es una exageración ponerse tan puntillosos y andar con tantas sospechas y conjeturas. Pero en lo tocante al anonimato, todas las prevenciones están justificadas (piénsese que los sistemas de voto telemático deben ser no sólo **tan seguros** como los sistemas de votación mediante papeleta, sino **más seguros** aún).

En cuanto al requisito de *imposibilidad de coacción*, parece claro que el Votante no puede conocer su k_{dv} porque su tarjeta inteligente no se la deja ver. Así, no podrá posteriormente demostrar ante nadie qué es lo que ha votado y podrá evitar votar bajo coacción o vender su voto al mejor postor.

En cuanto al tercer y último requisito que estamos analizando, del análisis del procedimiento empleado y de la robustez de los mecanismos criptográficos utilizados, se deduce que la fiabilidad del sistema ante ataques externos (según se analizó en el punto A y en las notas aclaratorias) es suficientemente aceptable, así como la garantía de la *integridad de los datos* que se manejan.

En resumen, podemos afirmar que el SistemaA descrito cumple bastante bien con la exigencia de anonimato, no coacción y fiabilidad que remarcábamos al principio.

¿La cumple también el SistemaB? Analicemos brevemente su comportamiento. Los tres pasos A, B y C que antes hemos señalado para el SistemaA se llevan a cabo

de forma muy similar para el SistemaB. Nuevamente hay que recordar que las tareas de transformación criptográfica de los datos y la conformación de las APDUs se llevan a cabo cooperativamente entre la tarjeta y la applet. Debido a ello, la applet sí «ve» en algún momento el valor de la k_{dv} en claro. Por esa razón, generando una applet falsa podría llegar a conocerse k_{dv} . Este riesgo lo tendremos en cuenta a la hora de decidir qué es lo que publicará el Contador después del recuento.

Si la votación se realiza desde casa, o desde sitios dispersos y desconocidos, el anonimato está aceptablemente garantizado porque es evidente que el Votante disperso no es fácilmente espiable. Pero, en cambio, la coacción está servida. Al calor del hogar (o de la residencia de ancianos) todo se reblandece. Lo mismo cabe decir de la compra de votos a domicilio. Este es un riesgo excesivamente grande.

Hay que decir que el uso para el que están pensados sistemas con una estructura similar al SistemaB están relacionados con la sustitución del voto por correo, que lleva consigo los mismos riesgos de coacción, pero aumentados. Si además el voto por correo está pensado para electores que viven fuera del país en el que se realiza la votación, con el añadido de las embajadas y de la privatización de los servicios postales a nivel internacional, la cosa es todavía más imperfecta. Es decir, que un sistema de las características del SistemaB es mucho más fiable que lo que existe actualmente para el voto de esos electores ausentes. Si se concentrasen los puntos de votación en locales especializados (por ejemplo embajadas o casas regionales) este riesgo disminuiría notablemente, porque la solución se aproximaría a la de los PA y PV diseñados para el SistemaA.

La concentración de los computadores que hacen las veces de puntos de votación en el SistemaB acarrearía, no obstante, posibles riesgos de manipulación del computador desde el que se vota (según hemos discutido al analizar el SistemaA). Pero con todo, al tratarse de una applet firmada, los riesgos de manipulación son menores que los riesgos de coacción que antes hemos comentado.

En resumen, un sistema de las características del SistemaB no soportaría la comparación con un sistema convencional de urnas y papeletas y sólo se justificaría su implantación si el modelo al que trata de sustituir, como es el voto por correo, conlleva todavía mayores riesgos.

APERTURA DE LA URNA Y RECuento

En los sistemas convencionales de votación mediante papeleta, los votos permanecen guardados en la urna hasta que finaliza el periodo de votación, instante en el cual se abre la urna. Si las urnas son transparentes, como es el caso del Estado español, las papeletas de votación deberán estar guardadas en sobres para mantener oculto su contenido ante miradas indiscretas. En el momento de la apertura, se sacan las papeletas de los sobres y las personas autorizadas proceden al recuento de los votos obtenidos por las distintas opciones. Algo similar (y si es posible, más seguro aún) debe ocurrir con los sistemas de votación telemática.

c) El recuento debe hacerse de forma fiable y auditable

En primer lugar, la Urna debe guardar los votos de forma fiable de manera que no sea factible alterarlos durante el tiempo que dura el proceso de votación (requisito número 7: *fiabilidad en los procedimientos e integridad* de los votos guardados). Además,

hasta que no se llegue al final de la votación nadie puede conocer resultados parciales (requisito 9: *neutralidad*) para que no se puedan establecer influencias en el electorado.

Por otra parte, en los sistemas de votación mediante papeletas, la presencia de representantes (interventores) de las candidaturas o de los electores representa una garantía tanto para el proceso de entrega de voto como para el proceso de recuento. En lo que a este último se refiere, en los sistemas de votación telemática deberán existir mecanismos robustos y procedimientos organizacionales que garanticen la corrección de los resultados finales (requisito 8: *certificabilidad* o *auditabilidad*).

Analicemos el comportamiento del SistemaA para evaluar si cumple adecuadamente estas exigencias. Todos los votos permanecen en la Urna hasta el final del periodo de recepción de votos. Además, la Urna los guarda pero ni ella ni nadie (excepto el Contador) puede leerlos por estar protegidos en *Sobres Seguros C*. La apertura se realizará de la manera siguiente:

1. Para proceder a la apertura de la Urna se necesitará la presencia física y simultánea del Gestor del Sistema Administrador y de la Autoridad de Elección que aportarán una información secreta introduciendo sus respectivas tarjetas inteligentes en los lectores habilitados para ello en la Urna y autenticándose biométricamente ante sus respectivas tarjetas.
2. La Urna aleatoriza todo lo que ha ido recibiendo (modifica el orden de aparición de los registros) y lo envía al Contador, publicando a su vez una lista con lo enviado (aquí están los votos cifrados dentro de *Sobres Seguros C*).
3. En ese momento, la Urna borra toda la información que ha ido adquiriendo durante su funcionamiento. El proceso mediante el cual se produce el borrado será auditable por parte de aquellas personas o entidades que políticamente se determine que tengan autorización para ello.

A continuación se procederá al recuento de los votos, que se llevará a cabo de la forma siguiente:

1. Antes de iniciar la lectura de los resultados, nuevamente el Gestor del Sistema Administrador y la Autoridad de Elección, con sus tarjetas inteligentes (mediante un procedimiento de secreto compartido) proporcionarán de forma conjunta al Contador su clave privada (que ha permanecido guardada y oculta hasta este momento) necesaria para que el Contador entre en funcionamiento.
2. El Contador, después de recibir toda la información procedente de la Urna «abre» los *Sobres Seguros C* que contienen la información sobre el voto antes descrita: voto cifrado con k_{cv} , la clave k_{dv} y la clave k_{dv} firmada por el Administrador. Para cada uno de los votos, el Contador verifica que la k_{dv} (que sirve para abrir el voto cifrado) esté correctamente firmada y, a continuación, lo descifra.
3. La clave k_{dv} firmada por el Administrador sirve de garantía para el Contador de que nadie excepto un Votante autorizado puede entregarle un voto válido. Aunque la Tarjeta de Votante impide que el Votante pueda votar más de una vez, si esta protección fuese violada, el Contador lo detectaría (y subsanaría esa incidencia) porque aparecería más de un voto con la misma k_{dv} .
4. Por último, una vez realizado el recuento, el Contador hace públicos los resultados de la votación y genera una lista en la que para cada una de las entradas aparecen: a) el voto en claro, b) la clave k_{dv} , c) la clave k_{dv} firmada por el Administrador.

Los resultados globales de la votación serán, obviamente, públicos. La información generada por la Urna y la lista generada por el Contador deberán permanecer bajo custodia de la Autoridad de Elección (que pertenecerá, presumiblemente, al ámbito judicial) para atender a posibles reclamaciones o verificaciones por parte de personas autorizadas. Transcurrido un tiempo previamente regulado, toda esa información se destruirá de forma auditada. De esta manera, se suprime el peligro que representa el hecho de que las protecciones criptográficas, mediante las cuales se mantiene el anonimato de forma robusta, puedan ser superadas cuando, transcurridos unos años, evolucionen las técnicas criptográficas.

En cuanto al SistemaB, soslayaremos, por el momento, hacer un comentario detenido. Baste decir por ahora que el proceso de apertura de Urna y recuento puede ser exactamente igual al descrito para el SistemaA, pero que en la lista que genera el Contador no puede aparecer la k_{av} en claro porque, como hemos dicho, es factible que el Votante la averigüe, con lo cual podría demostrarle al Gestor del Sistema Administrador o a la Autoridad de Elección cuál ha sido exactamente su voto. Ese peligro no existe en el SistemaA porque la k_{av} permanece en la tarjeta inteligente del Votante sin que nadie, ni siquiera éste, pueda leerla.

De lo que acabamos de describir puede deducirse que los requisitos de fiabilidad y neutralidad se cumplen de manera bastante satisfactoria en este SistemaA, pero que la auditabilidad está sólo al alcance del Gestor del Sistema Administrador y de la Autoridad de Elección. No aparece de forma efectiva la presencia de la ciudadanía o de representantes de las candidaturas o de los electores.

En los procesos de votación mediante papeletas el Votante puede ver cómo su voto se guarda en la urna y asistir al proceso de recuento. Además, la presencia de **interventores** en representación de las candidaturas (o de los simples electores) introduce unas garantías que no aparecen en lo que hasta ahora hemos descrito del SistemaA y del SistemaB. Dicho de otra forma, el esquema reducido que hemos presentado en las Figuras 11.17 y 11.18 no satisface adecuadamente los requisitos relacionados con la *verificación individual* ni los que exigen la *verificación global* ni permite una *auditabilidad* satisfactoria del proceso (requisitos 5, 6 y 8 de la relación que antes presentamos). Habrá que mejorar su arquitectura y los procedimientos empleados para conseguirlo. De eso es de lo que hablaremos en los siguientes apartados.

VERIFICACIÓN INDIVIDUAL DEL PROCESO

El Votante pertenece a un tipo de mamíferos cuyos antepasados se irguieron sobre las patas traseras hace ya muchos miles de años (aproximadamente los mismos que lleva complicándole la existencia a toda especie de animal, vegetal o mineral que se ponga a su alcance), debido a lo cual, cuando el Votante se acerca a la urna pisando sobre el suelo y mirando la papeleta que introduce en ella, se encuentra «en su medio» y adquiere bastante confianza en que luego, cuando se abra, su papeleta será una de las que se tengan en cuenta (debido también a la presencia de interventores que vigilan que no se saque nada de la urna hasta que el proceso termine).

En el voto telemático, en cambio, todo es remoto, todo es virtual, nada está al alcance de la mirada que el Votante puede lanzar desde la ancestral posición erguida con la que participa en los procesos convencionales de voto mediante papeleta. Por todo ello, será necesario diseñar mecanismos robustos e ingeniosos para que el Votante (de por sí, o fiándose de la opinión de gente experta de su confianza) acepte

convencidamente su implantación. Estos mecanismos serán los que sirvan para dar satisfacción al requisito de *verificabilidad individual* recogido con el número 5 en la relación que estamos usando de referencia. Para que haya democracia es necesario que exista convencimiento, no resignación.

d) El Votante debe convencerse de que su voto ha sido tenido en cuenta correctamente

La dificultad que conlleva el generar una prueba palpable para que el Votante se convenza de que su voto ha sido tenido en cuenta es que tiene que llegar a ese convencimiento (como ocurre en la votación con papeletas) sin que se le entregue una prueba material legible porque en ese caso podría demostrar su voto ante terceros para aceptar una coacción o venderlo (se incumpliría el requisito número 4).

Dejando para más adelante el SistemaB, cabe decir que en el SistemaA se puede introducir una comprobación de voto tal que la representada en la Figura 11.20. Para ello se incorporan nuevos agentes telemáticos:

- *Puntos de Verificación.* Se trata de cabinas, sin capacidades criptográficas, dotadas de lector de tarjetas inteligentes que permiten al Votante comprobar que su voto ha sido tenido en cuenta y ha sido correctamente contabilizado.

Una vez haya finalizado la votación, y durante un período de tiempo prefijado, cada Votante puede comprobar de forma independiente que su voto ha sido correctamente tenido en cuenta. Para ello basta con que el Votante se dirija a un Punto de Verificación y, siempre de forma individual, utilice su tarjeta y pida que se le muestre el voto asociado. El sistema ubicado en el Punto de Verificación está habilitado para leer la k_{dv} almacenada en la tarjeta del Votante, de manera que, una vez obtenida dicha clave de la tarjeta, accede vía telemática a la lista de parejas k_{dv} en claro-voto generada por el Contador y le muestra el voto asociado, de forma que el Votante y sólo éste pueda leerlo. Pero no se le entrega ninguna prueba fehaciente.

En realidad se trata más de una comprobación que de una verificación robusta. Este procedimiento sirve para comprobar que no han existido errores técnicos, pero no protege contra la falta de honradez del sistema global. Por ejemplo, si el Contador ha obtenido unos resultados correctos y ha publicado unos resultados falsos, cuando un Punto de Verificación le pregunte, le informará del resultado correcto, con lo cual el Votante quedará contento y defraudado.

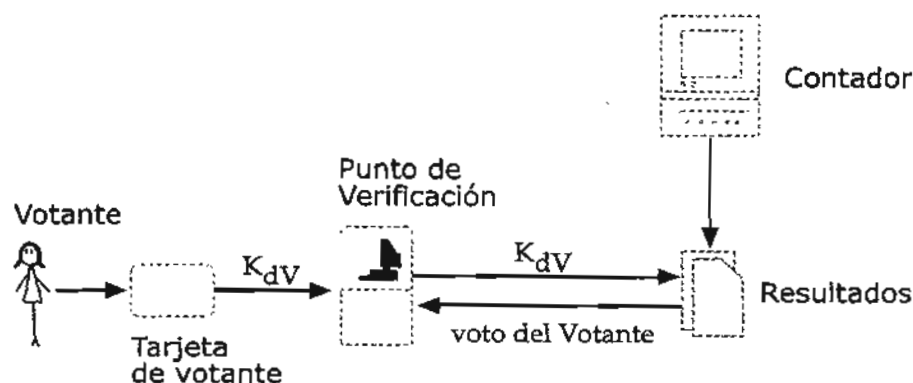


Figura 11.20. Comprobación por parte del Votante de la presencia de su voto en el resultado.

Este problema se puede eliminar o disminuir mejorando los procesos colectivos de auditoría y control que más adelante comentaremos. No obstante hay que prever que el Votante, de forma individual, quiera tener pruebas criptográficamente robustas que le sirvan en caso de duda o sospecha. En ese caso, o en el caso de que el Votante no esté conforme con la opción visualizada, es necesario buscar un procedimiento más seguro, tal y como se discute a continuación.

e) En caso de reclamación, el Votante debe recibir pruebas del sentido de su voto

De lo que se trata es de cumplir de forma más rigurosa el requisito número 5 relativo a la *verificabilidad individual*, sin vulnerar otros requisitos que pueden aparecer como contradictorios. El mecanismo criptográfico que puede resolver este compromiso será un *comprobante de votación* que reciba el Votante con el contenido de su voto pero que esté suficientemente protegido como para que no se pueda «negociar» con él ni usarlo fuera de circunstancias muy especiales y en presencia de testigos relevantes.

En la Figura 11.21 se muestra la forma de obtener el *comprobante* tanto en el SistemaA como en el SistemaB, aunque para explicar su utilización nos centraremos en el primero de ellos debido a que aún no hemos dicho cómo publica los resultados el SistemaB. La incorporación del comprobante se consigue modificando el proceso que describimos en el apartado (de este mismo epígrafe) *Entrega del voto a la Urna*. Así:

- En el punto A) decíamos que en la Tarjeta del Votante se genera una «información adicional» que se junta con el *Sobre Seguro C* y se «guarda» en un nuevo *Sobre Seguro U* que sólo la Urna puede abrir (Figura 11.19). Esa es en realidad una *clave simétrica* que la TV manda a la Urna.
- En el punto C) la Urna, tras abrir el *Sobre Seguro U* a ella destinado, obtiene, como ya se dijo, además del *Sobre Seguro C*, esa «información adicional» (la *clave simétrica*). A partir de esos datos, devuelve al Punto de Votación un *comprobante* de la votación realizada. Para generar el comprobante realiza las siguientes operaciones: a) la Urna cifra el *Sobre Seguro C* con la clave pública de la Autoridad de Elección, y b) la Urna firma lo anterior con su clave privada. A continuación la Urna cifra el comprobante con la *clave simétrica* que recibió del Punto de Votación. Una vez hechas las tres operaciones envía lo resultante al Punto de Votación.
- Es necesario añadir a la descripción un nuevo punto, el D), que consiste en que el Punto de Votación entrega lo recibido a la Tarjeta de Votación, la cual elimina el cifrado con la clave simétrica, obteniendo *el comprobante*. Después verifica la corrección de la firma por parte de la Urna (si bien no puede conocer

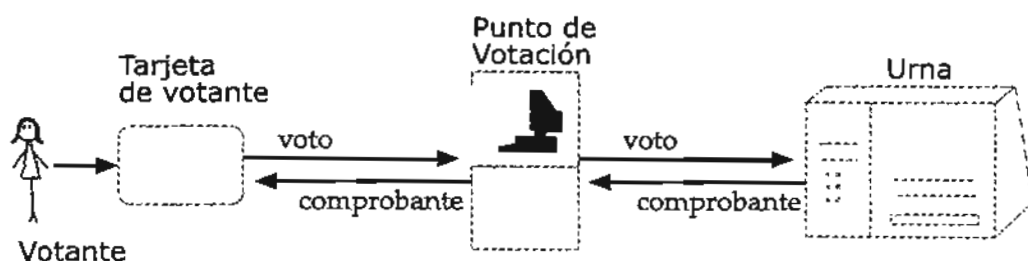


Figura 11.21. Incorporación de un comprobante de votación.

el contenido del comprobante por estar cifrado con la clave pública de la Autoridad de Elección). El comprobante de voto es guardado en la Tarjeta del Votante y sólo la Autoridad de Elección podrá acceder a los datos de ese comprobante en caso de reclamación (según estipulen las reglas que se dicten), una vez haya finalizado el proceso de votación.

Este comprobante así generado es una prueba que el Votante no puede verificar en el momento de la votación (sólo sabe que ha sido firmada por la Urna), pero que sabe que podrá usar en caso de reclamación. El procedimiento podría modificarse para que la TV sí pudiese verificar que el comprobante es exactamente el *sobre Seguro C* que ella envió, pero firmado por la Urna. Eso dejaría la protección de la no posible lectura del comprobante «sólo» en brazos de la cualidad *tamper-resistant* de la tarjeta. Por ello, para dejar más cubierto el requisito número 4, lo mejor es optar por la solución antes descrita.

Una vez finalizado el proceso de votación y publicados los resultados, con el *comprobante* almacenado en su tarjeta, cifrado con la clave pública de la Autoridad de Elección, el Votante puede decidir iniciar un *Proceso de Reclamación* (o verificación robusta) ante esa misma Autoridad de Elección. Para ello deberá solicitar esa comprobación y acudir en compañía de algún representante legal que, además, tenga conocimientos criptográficos para poder dar fe de la corrección o no de las pruebas que se le presenten. En esa comparecencia se le entregará la tarjeta inteligente del Votante a la Autoridad de Elección (que, como hemos dicho, lo razonable es que pertenezca al ámbito judicial).

En ese mismo acto la Autoridad de Elección, tras recibir la tarjeta del Votante, puede demostrar sin ninguna ambigüedad un tratamiento correcto o incorrecto del voto, pues tiene acceso a:

- La k_{av} del Votante almacenada en su tarjeta.
- El comprobante enviado por la Urna al Votante (firmado por la Urna y garantizada su inviolabilidad por la clave pública de la Autoridad de Elección) donde se contiene el voto emitido.
- Los registros del Contador que relacionan la k_{av} firmada por el Administrador, la k_{av} en claro y el voto cifrado con k_{cv} .

Con todo ello la Autoridad de Elección, apoyándose en pruebas criptográficas robustas, que puede exhibir ante el Votante o ante su representante-perito, dictaminará si el Votante no tiene razón o si ha existido una falsificación por parte del sistema.

La limitación de que en esta prueba el Votante tenga que desvelar su voto hace que esta verificación sólo pueda ser llevada a cabo por ciudadanos comprometidos (de los que tantos existen en un proceso de votación) que, incluso, hayan hecho pública previamente su decisión acerca de qué es lo que van a votar. Pero la robustez de la prueba es tal que con **un solo caso** comprobado en el que la autoridad judicial dictamine que *el Sistema*, o **sus gestores**, han cometido fraude, podría declarar NULO todo el proceso electoral. Lo cual representa una espada de Damocles tan cierta, que el Sistema se ve impelido a **ser honrado** y no cometer fraude alguno (todo esto además de que, como hemos visto, ni los agentes telemáticos ni los gestores tienen resquicios razonables para poder «meter mano» en el proceso).

AUDITABILIDAD Y VERIFICACIÓN GLOBAL

Por lo general, el Votante no se fía en exceso de lo que sus conciudadanos oponentes puedan hacer con su voto, quizás porque piense que sus adversarios políticos no deberían fiarse de lo que pudiese hacer él con los suyos. Por eso, para que el proceso electoral pueda considerarse aceptable es necesario que se permita la existencia de **interventores**, que serán determinados ciudadanos que representan a las candidaturas que compiten en la elección, o bien simplemente pertenecerán a agrupaciones de electores interesados en el proceso.

Algo semejante habrá que pensar en el diseño de sistemas de voto telemático. Como se pone de manifiesto en los requisitos detectados (número 6, *verificabilidad global*, y número 8, *auditabilidad*), esta supervisión deberá hacerse tanto durante el transcurso del proceso de votación (mediante el registro de pruebas que permitan a posteriori comprobar la corrección del sistema) como al final del proceso, cuando se proceda a la apertura de la Urna y al recuento de los votos. De la búsqueda de protocolos y mecanismos que generen pruebas criptográficas robustas para garantizar esta supervisión es de lo que nos ocuparemos en los párrafos que siguen.

f) Debe permitirse que ciudadanos autorizados supervisen todo el proceso

Pensemos por un momento cómo se desarrollan las cosas en un sistema convencional de votación mediante papeletas, por ejemplo el que está establecido en el Estado español (que será similar, con algunas variaciones, al que se sigue en otros países de nuestra misma zona cultural). La persona que va a votar selecciona primeramente su voto y lo introduce en un sobre opaco. Posteriormente se aproxima al lugar en el que está ubicada la urna, custodiada por un pequeño ejército de ciudadanos constituido por el presidente de la Mesa y dos vocales designados por sorteo entre todos los electores, además de un número indeterminado de *interventores* que actúan en representación de distintas candidaturas contendientes (pueden ser también simples ciudadanos que manifiesten, conforme a unas normas existentes, su deseo de participar en el proceso en calidad de observadores). Todos ellos constituyen la Mesa electoral y aguantan allí más de doce horas, unidos por el destino y sentados en sillas habitualmente diseñadas para escolares con una base de sustentación bastante menos generosa que la que han alcanzado los adultos constituyentes de la Mesa.

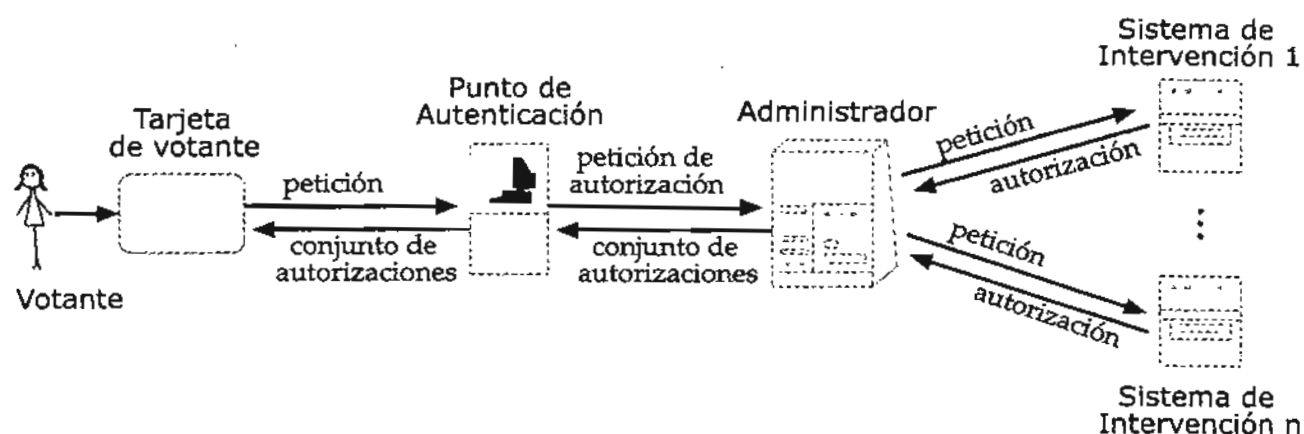


Figura 11.22. Incorporación de Sistemas de Intervención.

Los representantes «oficiales», el presidente y los dos vocales, disponen de un listado en el que aparecen todas y cada una de las personas con derecho a depositar su voto en esa urna. Una copia idéntica de dicho listado está en posesión de todos y cada uno de los interventores.

Cuando la persona que va a depositar su voto se acerca a la urna, se identifica documentalmente ante el presidente, el cual pronuncia en voz alta su nombre. Tanto el vocal que maneja el listado como las personas que actúan como interventoras localizan al votante en la lista, dicen en voz alta el número de orden que ocupa en ella, y lo tachan, indicando con ello que ya ha ejercido su derecho al voto. En ese momento se introduce su voto en la urna.

Estudiemos a continuación cómo puede trasladarse un escenario similar a un esquema de voto telemático. Para ello veamos una mejora que puede añadirse a los dos sistemas, SistemaA y SistemaB, que estamos utilizando como ejemplo a efectos didácticos. Con esta mejora aparecen nuevos agentes telemáticos y nuevos actores humanos. Éstos son:

- Varios Sistemas de Intervención (SIs) que complementan la labor del Administrador. En la Figura 11.22 se representa ese añadido.
- Interventores responsables de cada uno de los Sistemas de Intervención. Cada uno de estos sistemas está controlado por un Interventor nombrado por cada una de las agrupaciones de electores o candidaturas autorizadas para supervisar la votación.

La incorporación de estos nuevos Sistemas de Intervención sirve para que comprueben, en paralelo con el Administrador, que la solicitud realizada por el Votante es correcta y también para que esos sistemas firmen la autorización de votación que recibe el Votante. Ello permitirá que, en caso de producirse una incidencia, todos tengan constancia de ella (al igual que ocurre en la votación convencional por papeleta que acabamos de referir). Cabe señalar que la función fundamental de los SI es supervisar la actuación del Administrador para evitar manipulaciones. Así, el proceso que describimos en el apartado (de este mismo epígrafe) *Autenticación y autorización del Votante* se vería modificado de la manera siguiente:

- En el punto 1 decíamos que la tarjeta genera una *clave de solicitud de autorización* (k_A , a la que luego denominamos k_{dv}) y que la opaca para el Administrador. Ahora es necesario añadir que la opaca también para todos y cada uno de los SIs. Posteriormente, en el punto 3, el Administrador, una vez recibida la APDU procedente del Punto de Autenticación, debe enviar todos los datos a **todos** los Sistemas de Intervención.
- Cada uno de los Sistemas de Intervención, al igual que el Administrador, deberá comprobar si el identificador de votante recibido es correcto y si el Votante no ha realizado previamente otra autenticación. Si esto es así, cada uno de ellos firmará a ciegas la clave opacada de solicitud de autorización.
- Al final del proceso de autenticación, una vez que la TV retira todos los factores de opacidad de la autorización recibida, se queda con la clave de autorización k_A (o k_{dv}) firmada por el Administrador, la k_A (o k_{dv}) firmada por el Sistema de Intervención 1, la k_A (o k_{dv}) firmada por el Sistema de Intervención 2 y así sucesivamente. Esta k_A (o k_{dv}) **firmada** por el Administrador y por todos los SIs constituye la verdadera *autorización* para votar.

Consecuentemente, este nuevo valor de la autorización para votar repercute en todos los restantes pasos del proceso. Así, cuando la TV entrega el voto a la Urna, el contenido del *Sobre Seguro C* no será exactamente el mismo que describimos al explicar el SistemaA reducido (Figura 11.19), sino que la firma de la clave de descifrado será $Ad_s(k_{dv})$, $SI1_s(k_{dv})$, ..., $SI_n_s(k_{dv})$ (suponiendo que existan n Sistemas de Intervención). Asimismo, cuando el Contador hace públicos los resultados de la votación, junto al voto en claro y a la clave k_{dv} añadirá todas esas firmas de la clave de descifrado, que sirven para garantizar que esa fue la clave que supervisaron tanto el Administrador como todos los Sistemas de Intervención durante el proceso de autenticación y autorización.

g) Debe permitirse que ciudadanos autorizados puedan verificar la corrección del recuento

Como hemos visto, la presencia de Interventores y de los Sistemas de Intervención por ellos gestionados sirve para introducir registros y elementos de auditoría durante el transcurso del proceso de entrega de voto. Asimismo, podemos imponer mejoras en el procedimiento descrito en el apartado *Apertura de la Urna y recuento*, de manera que sea necesaria la presencia física y simultánea no sólo del Gestor del Sistema Administrador y de la Autoridad de Elección, sino también de los respectivos Interventores, los cuales (mediante un procedimiento de secreto compartido semejante al

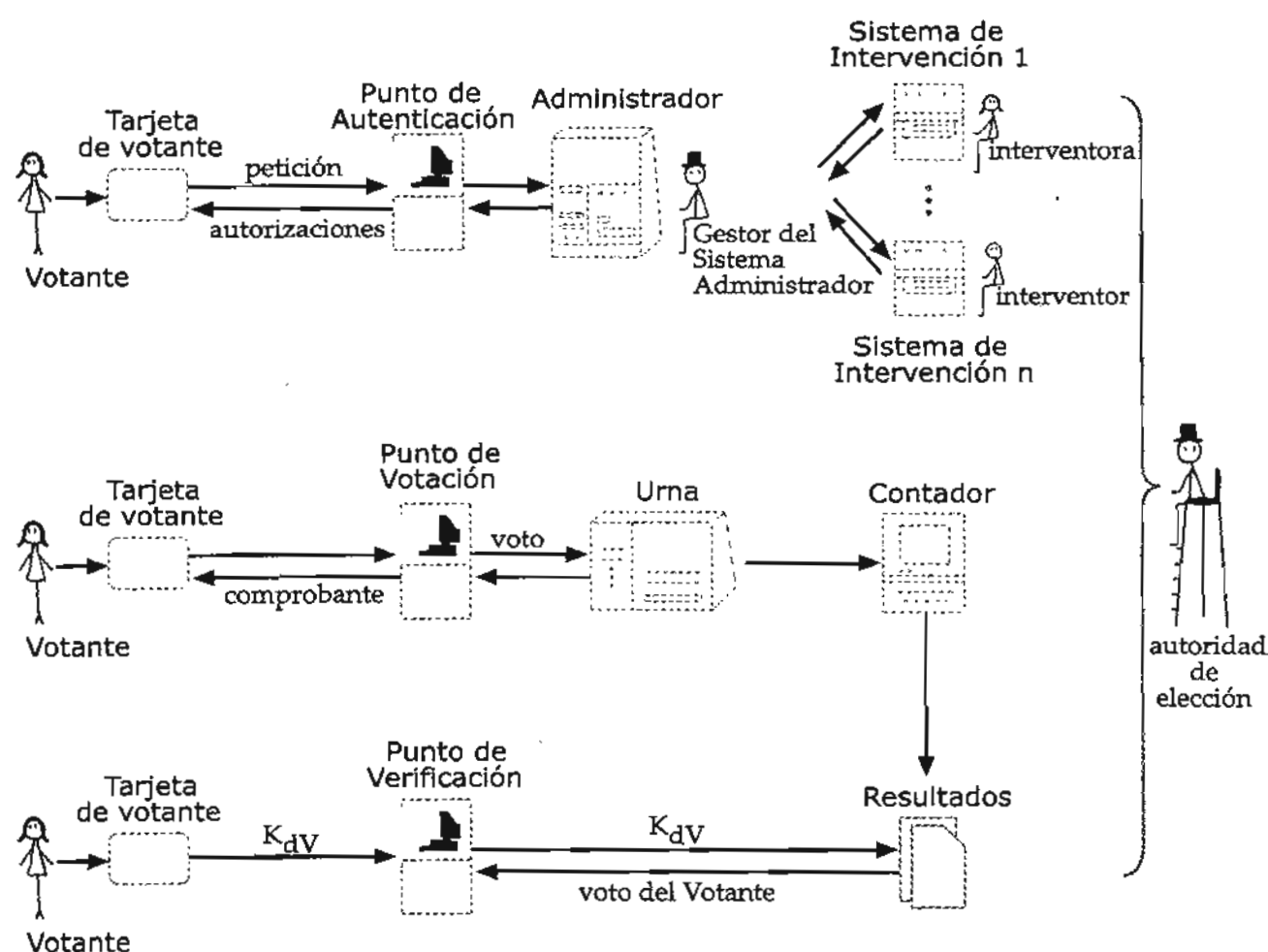


Figura 11.23. SistemaA completo.

que ejemplificamos en el epígrafe 11.1) sean protagonistas imprescindibles para que se pueda abrir la Urna y obtener la clave privada del Contador.

Esta presencia de los Interventores, que representan a las candidaturas o a agrupaciones de electores, puede servir, haciendo uso de toda la información complementaria que hemos ido comentando, para llevar a cabo una verdadera verificación global de los resultados. En efecto, con la intención de que los distintos Interventores obtengan una prueba del correcto funcionamiento del Contador, se le puede entregar a cada uno de ellos una copia de la lista generada por el Contador para que compruebe que son correctos los resultados finales publicados.

En realidad, se podría entregar a los Interventores la lista generada por la Urna para que la analizaran y evaluaran por sus propios medios. Ello no representaría ningún problema para garantizar el anonimato del voto debido a las protecciones criptográficas robustas de esos registros. Pero, como comentamos antes, lo que hoy se puede garantizar puede ser vulnerado años después cuando la evolución de la criptografía lo permita.

Para evitar ese riesgo, la *Verificación global* debe llevarse a cabo de forma que la copia de la lista que se entrega a los Interventores se cargue en las mismas máquinas en las que residen los Sistemas de Intervención (que forman parte del Sistema), que podrán ser previamente auditadas por peritos de confianza para comprobar que solamente sirvan para llevar a cabo las operaciones que tienen encomendadas y que no puedan almacenar información en discos u otros dispositivos periféricos.

Se deberá establecer un periodo de vigencia tanto para las listas de registros generados por el Contador como para las listas de información entregadas a cada candidatura, de manera que, una vez transcurrido el tiempo estipulado para permitir la verificación individual y la verificación global, toda esa información sea destruida de forma auditada (al igual que en las votaciones convencionales se destruyen las papeletas).

Con los añadidos analizados en los últimos apartados, puede deducirse que el SistemaA (u otro sistema que sea capaz de responder positivamente a los interrogantes que nos hemos ido planteando) cumple bastante satisfactoriamente los diez primeros requisitos enunciados en la relación que estamos usando de referencia. En la Figura 11.23 se muestra el escenario completo de este sistema simplificado al que, a efectos de su utilización didáctica, hemos dado en denominar SistemaA. Podríamos decir, incluso, que un sistema de votación telemática tal que este ofrece bastantes más garantías y seguridades que un sistema de votación mediante papeletas.

¿Y el otro sistema (el SistemaB) que estábamos usando como referencia didáctica para analizar las funcionalidades que debe cubrir un sistema de votación? Habíamos visto que cubre moderadamente bien los requisitos que condicionan la fase de autenticación y autorización, así como que cumplía deficitariamente los que condicionan la fase de entrega de voto. Una vez vista la información que, en el SistemaA, el Contador pone en conocimiento del Gestor del Sistema Administrador, de la Autoridad de Elección y de los Interventores, se deduce que el posible conocimiento que el Votante del SistemaB puede adquirir acerca del valor de la clave de descifrado de voto, k_{dv} , hace que una solución tal que esa posibilitaría la coacción y la venta de votos. Será necesario, por tanto, buscar soluciones que no exhiban la k_{dv} de forma tan manifiesta. Es posible encontrar soluciones que, aunque no de forma tan robusta como la descrita para el SistemaA, sirvan para cubrir esta exigencia. Por razones de espacio y de oportunidad, soslayaremos aquí analizar y evaluar estas posibles soluciones.

Bástenos aquí la descripción que de estos sistema hemos hecho. Nuestro objetivo inicial no era conocer cómo funcionan estos dos ejemplos simplificados sino apoyar-

nos en ellos para discutir acerca de los requisitos y exigencias que demandan los sistemas de votación telemática. El objetivo didáctico de estos análisis es, como antes decíamos, contribuir a que el lector esté en condiciones de aplicar esos criterios para evaluar los distintos sistemas de votación telemática que se le presentan en la actualidad y los que se le presentarán en un futuro.

En el siguiente y último apartado de este epígrafe comentaremos otros requisitos y exigencias que también es necesario tener en cuenta a la hora de proponer este tipo de sistemas de votación.

OTRAS CARACTERÍSTICAS Y REQUISITOS DEL VOTO TELEMÁTICO

En los apartados precedentes hemos estado discutiendo acerca de las características que deben tener los sistemas de votación telemática para dar satisfacción a algunos de los requisitos demandados. De hecho, nos hemos centrado sobre los nueve primeros requisitos (los que conllevan mayores retos desde un punto de vista tecnológico) y hemos tocado de pasada el requisito número 10 relativo a la movilidad de los votantes.

También hemos tenido muy en cuenta la salvaguarda que representa no guardar registros criptográficos que en el momento de la votación son computacionalmente muy seguros pero que pasados unos años pueden dejar de serlo.

Dos aspectos tecnológicos que no se han abordado adecuadamente en la descripción que antes se ha hecho son los relacionados con la imposibilidad de colusión entre los agentes del sistema y los fallos que pueden producirse en las comunicaciones (ambos incluidos en el requisito de *fiabilidad*). En cuanto al primero de ellos, antes de dar por bueno un sistema de votación telemática será necesario analizar detenidamente todas las condiciones de riesgo que pueden presentarse y analizar las protecciones que tiene implementadas el sistema para evitar que eso pueda producirse. En cuanto al segundo, será necesario especificar con precisión los protocolos telemáticos en los que se soporta el sistema y las medidas de recuperación ante posibles fallos que puedan producirse en las comunicaciones y que dejen «a medias» cualquiera de los procesos de intercambio de datos y claves que antes hemos comentado.

Pero quedan otros requisitos (once, según la relación en que nos estamos apoyando, si incluimos entre ellos el de *movilidad de los votantes*) que también deben ser satisfechos por los sistemas telemáticos que pretendan ser candidatos a sustituir a los actuales sistemas convencionales de votación mediante papeletas. ¿Podemos calificar como «menores» a estos requisitos porque no se traduzcan en mecanismos de seguridad avanzados e ingeniosos? Responder afirmativamente a esta pregunta sería caer en el error «tecnocéntrico», tan común, de considerar poco importante todo lo relacionado con los condicionantes sociopolíticos que marcan la implantación de los sistemas presentes en la Sociedad de la Información. La realidad es bien distinta: serán precisamente esos condicionantes los que determinen en mayor medida la viabilidad o improcedencia de los sistemas propuestos.

Este es un texto dedicado a la ingeniería de seguridad en redes telemáticas y, por cuestión de oportunidad, debe prestar mayor atención a los temas más directamente relacionados con esa temática, pero es necesario avisar al lector de la importancia que tienen todos esos otros temas. Consecuentemente, recomendamos a quienes pretendan abordar la implantación de sistemas de votación telemática el análisis detenido de estos requisitos y de las fuentes documentales en que se apoyan. Bástenos, no obstante, aquí, hacer una breve referencia a algunos de los aspectos que consideramos más llamativos.

El requisito de movilidad de los votantes presenta facetas muy interesantes. Tanto en el SistemaA como en el SistemaB -que hemos estado utilizando como ejemplo- hemos supuesto la existencia de **una sola** Urna, pero en un sistema de votación a gran escala parece razonable que existan distintos distritos electorales y distintas *mesas* electorales. En el caso de España, el actual sistema electoral divide a los electores en *mesas* atendiendo a cuestiones de distribución geográfica, de tal forma que el número de electores que depositan su voto en una urna concreta no sea excesivo (el tope suele imponerse en torno a los 1.000 electores) para posibilitar las posteriores tareas de recuento. Pero en una votación telemática una sola Urna puede atender a bastantes más electores, dependiendo de cómo se dimensione. No obstante, habrá que tener en cuenta problemas de congestión para asegurar el *voto rápido*. Además, en la actualidad se conocen los resultados parciales por mesa y por distritos, y en un futuro la reglamentación de la votación telemática deberá determinar en qué medida ese conocimiento parcial es conveniente o perjudicial, determinando en base a ello la agrupación de los electores en colegios electorales y en urnas. Y esa es una decisión política muy importante.

En la actualidad, los votantes, para depositar su voto, tienen la obligación de acudir al espacio físico en el que se ubica su colegio electoral. Por contra, en un sistema de votación telemática, aunque el Votante tenga que acudir a votar a las distintas cabinas en las que se concentran los puntos de votación, y aunque el voto se deposite en la Urna lógica que corresponda a la «*mesa*» en la que esté inscrito, el Votante podrá acudir a cualquier cabina de las existentes, independientemente del lugar geográfico en el que se encuentre.

Otro aspecto directamente relacionado con el anterior es el relativo al periodo de tiempo que puede durar el proceso electoral hasta que se dé por cerrada la votación y se proceda al recuento de los votos. En la actualidad esa duración está condicionada por el aguante que cabe esperar de todas las personas que actúan simultáneamente en el proceso, pero en un sistema de votación telemática ese tiempo puede ser mucho mayor, aunque exista la figura del Interventor, ya que la supervisión del sistema puede organizarse a través de equipos de supervisores coordinados entre sí. También es este un tema políticamente muy sensible (con repercusiones técnicas y organizativas) que deberá ser abordado en la reglamentación que en su día se apruebe.

Conviene remarcar que para que se pueda hablar con propiedad de sufragio universal dentro de la Sociedad de la Información será necesario tomar medidas de alfabetización digital que den como resultado la *igualdad de oportunidades en la votación*, de tal manera que se minimicen las diferencias que en tal sentido puedan existir entre ciudadanos procedentes de distintos estratos sociales.

Para cerrar estos comentarios, seleccionemos el condicionante reflejado en el requisito reseñado con el número 14. Se trata del que preconiza que el código fuente de todos los programas debería ser conocido y verificable por todos los auditores y supervisores presentes en el sistema. Podría añadirse algo más: debería estar publicado en el Boletín Oficial del Estado (o en un documento allí referido). Es razonable que todas las personas y todas las empresas que participan en los procesos de votación convencionales (fabricando urnas, montando instalaciones y trayendo y llevando papeletas y resultados) sean compensadas económicamente de forma adecuada. Pero lo que no parecería de recibo es que los procedimientos reglamentados fuesen propiedad de nadie que no sea el propio Estado.

Paralelamente, en la plasmación telemática de las votaciones oficiales, la definición de los sistemas, los procedimientos organizativos y el comportamiento definido en los programas informáticos deben ser conocibles y evaluables por la ciu-

dadanía en general y estar bajo titularidad pública. Todo ello sin menoscabo de que, como ocurre en los procesos convencionales, cada cual reciba una compensación proporcional a su esfuerzo y dedicación, pero nada más. Otra cosa bien distinta serán los sistemas de votación desarrollados e implementados para dar satisfacción a la demanda que existe en diversos sectores privados (empresariales, deportivos, culturales, etc.), en los cuales las cosas se registrarán según las consabidas y difícilmente evitables reglas del mercado.

11.7. LA TARJETA COMO ELEMENTO CONSTITUTIVO DE LOS SISTEMAS QUE PROPORCIONAN SERVICIOS DE ANONIMATO

De lo que hemos visto en epígrafes anteriores acerca de los escenarios que se presentan en aplicaciones como las de voto telemático o comercio electrónico se deduce que, en los sistemas que hemos tomado como modelo, la tarjeta inteligente aparece, en todos ellos, como un elemento constitutivo del sistema global.

No es que esto tenga que ser necesariamente así, pero dado lo rigurosos que son los requisitos que deben cumplir estos sistemas, la experiencia dicta que las soluciones que incluyen tarjetas inteligentes ofrecen muchísimas posibilidades para conjugar el doble requisito de autenticación de la persona que accede al sistema y la garantía de anonimato en algunas de las operaciones que allí se llevan a cabo.

En los dos apartados que siguen vamos a prestar atención a dos aspectos muy importantes de esa presencia de la tarjeta inteligente en la arquitectura global del sistema:

- a) La autenticación del usuario ante su tarjeta. Esta deberá ser siempre una fase preliminar. Con posterioridad, el usuario tendrá o no que autenticarse ante el Sistema global, pero inicialmente es insoslayable que se garantice que solamente la persona legítimamente propietaria puede usar una tarjeta inteligente concreta.
- b) Contribución de la tarjeta inteligente en el soporte de la Aplicación Cliente. Para utilizar una nomenclatura que nos permita hablar con soltura de las distintas partes de un sistema (complejo o no), denominaremos Aplicación Cliente a aquella que utiliza la Persona Usuaria para acceder a los servicios del Sistema global. Como veremos más adelante, son posibles varias alternativas a la hora de definir esa participación.

AUTENTICACIÓN ANTE LA TARJETA

De todo lo visto se deduce el gran interés que tienen las tarjetas inteligentes como elementos constitutivos de los sistemas que estamos analizando. Pero tienen un talón de Aquiles: la posibilidad de que alguien malicioso consiga la tarjeta de la que es titular otra persona y consiga autenticarse como legítimo titular de ésta.

Por remota que sea esta posibilidad, caso de producirse, echaría abajo todo el edificio que hemos venido montando en torno a considerar a la tarjeta inteligente como el testigo de seguridad más idóneo para almacenar la clave privada del titular. Si un usuario malicioso consigue este propósito (por descuido o por fuerza) puede realizar firmas y operaciones en la red de las que sería responsable administrativa y

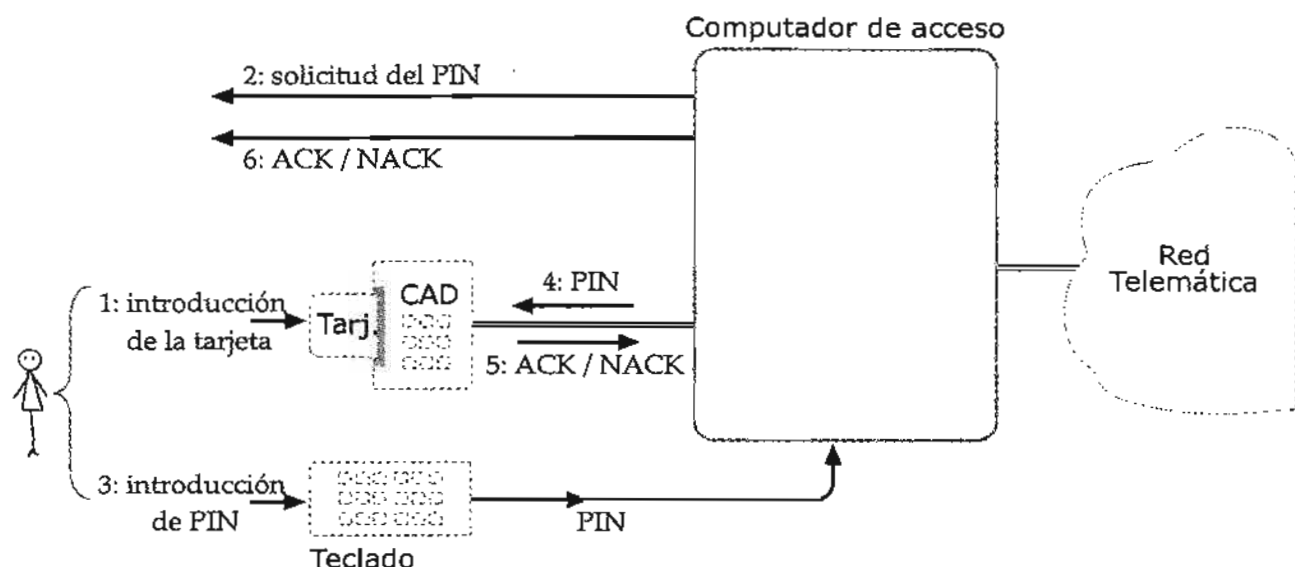


Figura 11.24. Autenticación del titular mediante PIN a través del computador.

judicialmente la entidad propietaria de la tarjeta. Y esto es así porque, en la Red, la clave privada actúa como un *alter ego* de su titular.

Por ello, los mecanismos de autenticación del usuario ante su tarjeta merecen toda nuestra atención. Como ya comentamos en el Capítulo 2, los más plausibles (en la doble acepción de la palabra) son el uso de un PIN o contraseña y el uso de mecanismos biométricos (principalmente los basados en huella digital).

Una cosa que hay que dejar clara es que en el proceso de autenticación del titular de una tarjeta ante su propia tarjeta, cuantos menos elementos intermediarios intervengan, tanto mejor. Es imprescindible que intervengan algunos porque la tarjeta inteligente no dispone de periféricos propios, así que «alguien» se los tendrá que proporcionar (aunque, según comentaremos más adelante, dentro de los procesos de mejora y evolución de las tarjetas inteligentes, una de las tendencias que existen es la de ir dotando de periféricos elementales a las tarjetas, convirtiéndolas, así, en dispositivos semiautónomos).

Un esquema clásico para este tipo de autenticación es, con algunas variaciones, el representado en la Figura 11.24. El computador, tras detectar que el titular ha introducido la tarjeta en el dispositivo de lectura-escritura (CAD), solicita que sea introducido el PIN a través del teclado. El computador envía el PIN recibido a la tarjeta y la aceptación o no (ACK o NACK) de la tarjeta se la comunica a la persona propietaria de ésta. En este esquema quien tiene el número secreto y quien realiza la comprobación es la tarjeta, pero el computador hace de intermediario y puede adquirir información muy sensible para la seguridad de la persona propietaria de la tarjeta. En efecto, un atacante malicioso puede alterar el normal funcionamiento del computador y capturar así el número secreto que le permitirá, consiguiendo la tarjeta, suplantar la personalidad de su legítimo titular.

Este esquema mejoraría notablemente si el teclado se comunicase directamente con la tarjeta a través del CAD (sin pasar a través del computador). De esta forma se acorta el camino y las posibilidades de ataque, ya que es más difícil y menos fiable proteger el programa del ordenador que gobierna el proceso representado en la Figura 11.24 que proteger el teclado y el CAD. Y si el CAD está integrado con el teclado en un subsistema autónomo, tanto mejor.

La autenticación mediante PIN se podría sustituir o reforzar mediante una autenticación biométrica, que es algo mucho más fiable. Según los entendidos, el análisis del

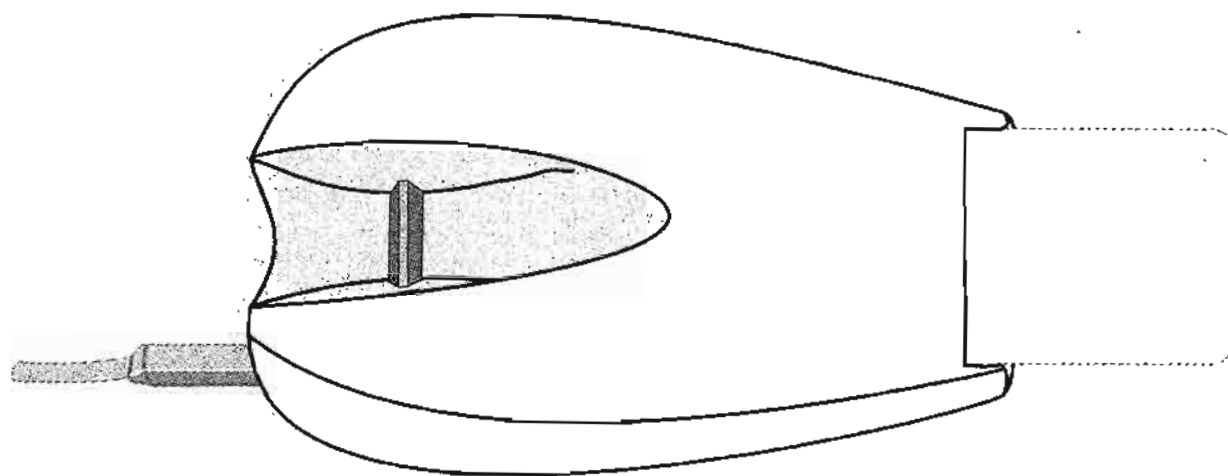


Figura 11.25. Lector de tarjeta con lector de huella incorporado.

iris del ojo es de las pruebas que mayor garantía ofrecen, debido a lo preciso de la información que aporta y al hecho de que se puede comprobar que no se trata de una muestra estática sino que la observación se está realizando sobre una persona «viva». No obstante, dentro del campo de la autenticación biométrica en tarjetas inteligentes, el análisis de la huella dactilar es la técnica que tiene mayor predicamento.

Esto es así porque al ser la tarjeta un componente manejable con la mano (valga la redundancia semántica) la autenticación mediante la huella de los dedos propicia un escenario bastante ergonómico y fácil de plantear. Aunque es relativamente sencillo capturar maliciosamente huellas dactilares de personas a las que se quiere suplantar la personalidad, la configuración de los lectores y de los escenarios donde se realice la comprobación puede ayudar a minimizar este riesgo. Así, algunos lectores de huella obligan a que el dedo sobre el que se tome la muestra se deslice sobre una especie de banda estrecha y en relieve (Figura 11.25), lo cual requiere que exista algo de movimiento y «vida» en el sujeto evaluado. (A buen seguro que a medida que avance el tiempo se irán introduciendo de forma creciente procedimientos ingeniosos que sirvan de salvaguarda contra los también ingeniosos métodos que puedan ocurrírseles a los atacantes para falsificar la prueba.)

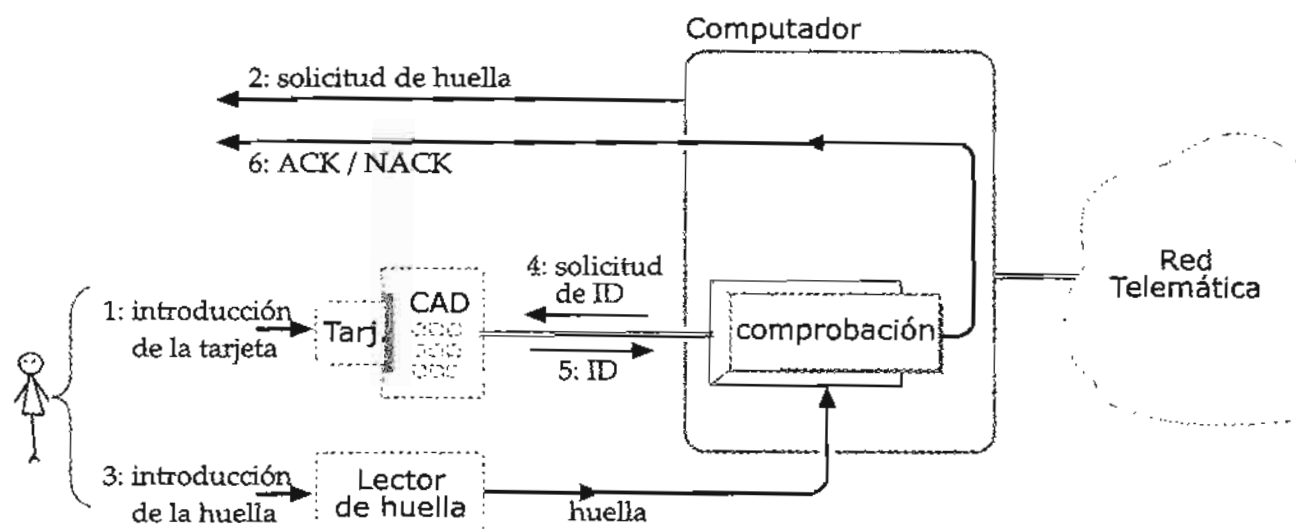


Figura 11.26. Autenticación mediante huella y verificación en el computador.

En la Figura 11.26 se representa un posible esquema para realizar la autenticación del titular de una tarjeta inteligente mediante análisis de la huella dactilar. En este caso, la huella está almacenada en el computador, el cual, al mismo tiempo que le solicita a la persona propietaria de la tarjeta que introduzca su huella dactilar, le solicita a la tarjeta un identificador que pueda permitirle verificar la validez de la huella.

Por las razones antes dichas, este esquema es muy poco atractivo de cara a las aplicaciones a las que antes hemos hecho referencia. En estos casos, en los que frecuentemente se requiere en alguna medida un servicio de anonimato, lo que debe plantearse es aprovechar las ventajas que proporcionan las técnicas biométricas, pero teniendo muy presente que hay que preservar los derechos de los ciudadanos que participan en esas comunicaciones. Por ello, será necesario minimizar lo más posible los problemas que puede acarrear la introducción masiva de muestras biométricas que vulneren la intimidad de los ciudadanos y la privacidad de sus actuaciones en la Red¹⁴.

Así pues, para garantizar esta privacidad resulta necesario que se realice dentro de la tarjeta la comparación entre la huella suministrada por la persona que se dice propietaria de la tarjeta y el patrón almacenado. Con esta funcionalidad, que se conoce en inglés como *match-on-card*, lo que se consigue es que sea la propia tarjeta la que compruebe que la huella de quien accede coincide con la que en su momento se tomó a la persona legítimamente propietaria de la tarjeta. Ello implica que en el momento de personalización de la tarjeta, antes de entregársela a su titular, será necesario almacenar la huella dentro de ella.

Un escenario para plasmar esta comprobación de la huella podría configurarse de forma semejante al representado en la Figura 11.24 (en ese caso la comprobación del PIN también se hace en la tarjeta, aunque ya comentábamos allí que sería más conveniente independizar la comunicación entre el lector del PIN y el CAD). Para materializar esa independencia, en el caso de la huella, casi siempre que se utiliza una verificación *match-on-card* lo que se hace es emplear un solo dispositivo (de aspecto similar a un ratón) en el que se integran el lector de huella y el CAD. En la Figura 11.27 se representa esquemáticamente una situación de este tipo en la que se ha

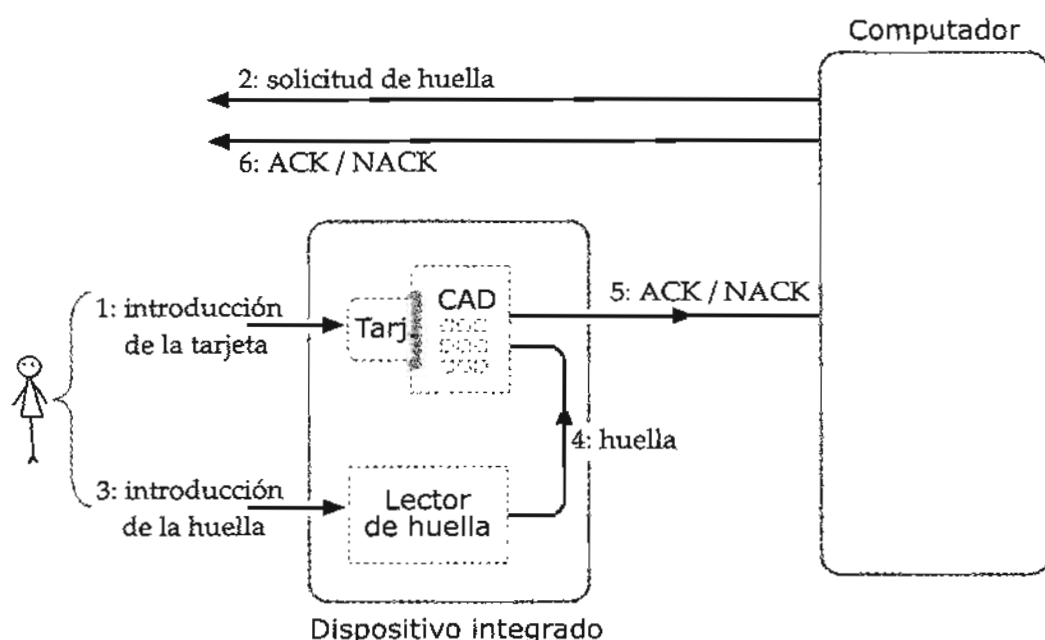


Figura 11.27. Autenticación mediante huella y comprobación de coincidencia en la tarjeta (*match-on-card*).

supuesto que el computador solicita la introducción de la huella e informa del resultado, pero este paso también se puede eliminar, porque el lector puede informar directamente (mediante una señal luminosa) del éxito o fracaso de la autenticación.

Para rizar más el rizo, se puede plantear incluso incrustar el lector de huella **en** (más bien **sobre**) la propia tarjeta de plástico. En este caso, la comunicación entre el lector y el microcomputador que constituye realmente la tarjeta inteligente se realiza por dentro de la tarjeta de plástico a través de un bus interno. Algunos de los fabricantes que ofrecen estos componentes emergentes los denominan *system-on-card*. Esta opción tiene la ventaja evidente de reducir al mínimo el riesgo de captura fraudulenta de la huella del titular, pero tiene dos inconvenientes: uno es el precio de la tarjeta (que se vería incrementado con el precio del lector de huella incorporado) y el otro inconveniente es que el grosor de la tarjeta en la zona donde está el lector es bastante mayor que el normalizado para todas las tarjetas conformes con la norma ISO 7816.

Algunos van más lejos y proponen incorporar también a la tarjeta una pequeña pantalla visualizadora en la que podrían aparecer, por ejemplo, el grupo sanguíneo y los medicamentos a los que es alérgica la persona portadora. Pero esto es en realidad un pequeño microcomputador portátil configurado en forma de tarjeta inteligente. Por el tipo de ejemplo que hemos puesto para justificar la posible utilidad de estos emergentes dispositivos, podemos deducir que su campo de aplicación está alejado de las necesidades que es necesario cubrir en las aplicaciones telemáticas seguras.

Para centrarnos en el campo que nos interesa, adoptando una posición práctica y realista, podemos concluir que nos sería suficiente disponer de tarjetas inteligentes que permitiesen, además de ejecutar cierto tipo de programas, garantizar la autenticación de su titular por medio de un PIN o una prueba biométrica y que, en ambos casos, la comprobación de coincidencia la realice, dentro de la tarjeta, el microcomputador incorporado en ella.

Como ya hemos dicho, en el caso del PIN, este tipo de comprobación es posible llevarla a cabo en tarjetas inteligentes tradicionales, es decir, totalmente conformes con la norma ISO 7816 (en sus siete partes). Teóricamente, no hay problemas insalvables para conseguir que la autenticación de huella *match-on-card* se pueda realizar también en este tipo de tarjetas, pero en la práctica la mayoría de las tarjetas que son capaces de llevarla a cabo son del tipo Java Card.

En este caso, la Tarjeta Java deberá disponer de un API biométrico añadido al conjunto de elementos que componen el entorno de ejecución Java Card (JCRE).

LA APLICACIÓN CLIENTE

Como antes decíamos, denominaremos Aplicación Cliente a aquella que utiliza la Persona Usuaria para acceder a los servicios del Sistema global. La Aplicación Cliente será, por tanto, una parte de la aplicación completa. El Sistema podrá proporcionar diversos servicios, a cada uno de los cuales se accederá con una Aplicación Cliente específica.

En la Figura 11.28 se representa el segmento de un escenario en el que la Aplicación Cliente está repartida entre el computador terminal del Sistema y la tarjeta inteligente. Es evidente que habrá muchos casos en los que la Persona Usuaria acceda directamente al Sistema sin necesidad de apoyarse en una tarjeta inteligente, pero, como ya hemos dicho, las situaciones que nos interesa analizar ahora son aquellas en las que sí es necesario el concurso de estos testigos de seguridad. En cualquier caso, la situación en la que no existiese una tarjeta inteligente para cooperar en la plasma-

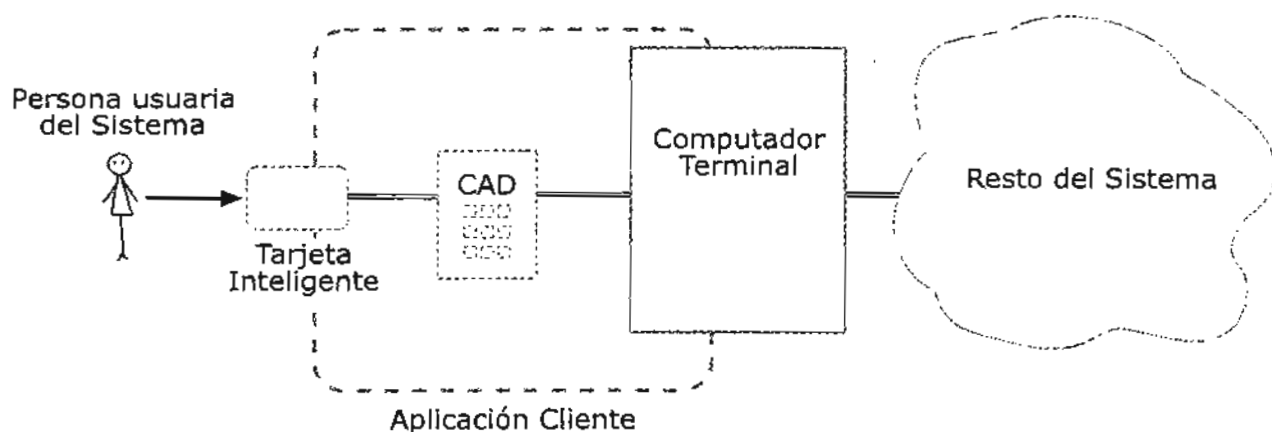


Figura 11.28. Acceso al Sistema mediante la Aplicación Cliente.

ción de la Aplicación Cliente sería un caso particular de la más general representada en la figura.

Como hemos visto anteriormente, en los distintos sistemas que hemos analizado para la provisión de servicios de voto telemático o comercio electrónico, la Persona Usuaria no accede siempre al mismo punto del Sistema global. Es decir, el computador terminal que se representa en la Figura 11.28 puede representar agentes telemáticos y puntos de acceso diversos, en los que residan entidades diversas. Por tanto, la que hemos llamado Aplicación Cliente es un caso genérico de las múltiples que pueden presentarse.

Dentro de una situación como esta podríamos a su vez distinguir tres casos:

- La Aplicación Cliente se ejecuta casi en su totalidad en el computador terminal del Sistema. La tarjeta inteligente le sirve a la aplicación: a1) como un testigo de seguridad que guarda informaciones particulares y privadas de la Persona Usuaria; y a2) como simple auxiliar para la realización de pocas y específicas operaciones criptográficas. Más concretamente, la clave privada de la Persona Usuaria está almacenada en la tarjeta y **no viaja nunca** fuera de ella, de forma que cualquier operación de cifrado o firma que deba realizarse utilizando dicha clave deberá ser llevada a cabo, mediante los necesarios algoritmos criptográficos, **dentro** de la tarjeta.
- La Aplicación Cliente se ejecuta mayoritariamente en el computador terminal pero también la tarjeta inteligente está involucrada directamente en la aplicación. Además de las tareas señaladas en el caso anterior, la tarjeta contiene datos propios de la aplicación, realiza autenticaciones de la entidad residente en el computador terminal, procesa o modifica los datos contenidos en su memoria en función de las APDUs que reciba y realiza internamente una parte importante de las operaciones criptográficas requeridas.
- La Aplicación Cliente se ejecuta mayoritariamente en la tarjeta inteligente de forma que el computador terminal se comporta como un simple auxiliar de la tarjeta, sirviendo para: c1) prestar sus terminales de entrada-salida para posibilitar el diálogo y la interacción con la Persona Usuaria; y c2) albergar la entidad comunicante que gobierne el protocolo telemático seguro mediante el cual se relaciona con los restantes agentes del sistema.

En esta clasificación no se ha querido tener en cuenta el caso en el que la tarjeta inteligente actúa como un simple testigo de seguridad que guarda datos particulares de

la Persona Usaria pero que no realiza operación criptográfica alguna. Además, esta división que acabamos de exponer no es absolutamente rigurosa ni conlleva que ante cualquier escenario que pudiera presentarse fuese inmediato averiguar a cuál de estos tres grupos pertenece. Posiblemente no sea este el caso y puede que el escenario en cuestión sea un híbrido entre algunos de los tres aquí señalados.

Para lo que sí nos va a servir esta clasificación es para permitirnos hacer una discusión acerca de los métodos necesarios para desarrollar la Aplicación Cliente en el acceso a los servicios telemáticos que sirven de soporte a la Sociedad de la Información.

a) La tarjeta como simple auxiliar

En este caso, la Aplicación Cliente reside en el computador terminal y la mayor parte de las tareas que lleva a cabo las realiza con recursos propios de esa máquina. Como tiene que ser capaz de soportar un protocolo seguro para comunicarse con el resto de los agentes telemáticos del Sistema, tendrá necesidad de realizar diversas operaciones criptográficas, para lo cual contará con una placa que contenga un módulo criptográfico, o bien con una librería contenida en un fichero lógico.

Una opción es desarrollar el programa que gobierna la aplicación en un lenguaje de alto nivel (también existe, en teoría, la posibilidad de codificarlo en lenguaje de ensamble, pero esto resulta mucho más tedioso) y, posteriormente, compilarlo y cargar el ejecutable en el computador terminal. Otra alternativa puede ser diseñar la aplicación para que resida en un determinado servidor y recuperarla en el momento oportuno mediante un proceso de telecarga usando, por ejemplo, tecnología Java. En este último caso sólo sería necesario que en el computador terminal residiese de forma fija un pequeño programa que fuese capaz de conectar con el servidor y descargar la Aplicación Cliente¹⁵.

La tarjeta inteligente que se requeriría para un escenario de este tipo podría ser una tarjeta convencional (de las que hemos estado denominando tradicionales) en la que el proceso de personalización consistiese básicamente en introducir en ella unos cuantos datos y parámetros particulares de la persona titular de la tarjeta. También podría ser una Tarjeta Java, pero sería más cara y necesitaría trabajos de desarrollo adicionales.

Entre los datos guardados en la tarjeta cabría incluir, al menos, un par de claves (pública y privada) que le permitan actuar como entidad comunicante dentro del Sistema. (Como ya hemos apuntado en otras ocasiones, si la tarjeta lo permite, sería conveniente que la Persona Usaria generase por su cuenta un nuevo par de claves y enviase la nueva clave pública a la Autoridad de Certificación adecuada para que generase el correspondiente certificado.)

En este caso, los datos almacenados en el proceso de personalización no cambiarán dinámicamente de forma significativa durante la ejecución de la Aplicación Cliente, por lo que la utilización que ésta haga de la tarjeta se limitará a lo que posibilite un sencillo API (*Application Programming Interface*) que proporcione el fabricante. En cuanto a las operaciones criptográficas se refiere, una buena solución es que este API sea compatible con la PKCS#11 de RSA (*Cryptographic Token Interface Standard*) que especifica un API normalizado para acceso a dispositivos (tarjetas inteligentes, por ejemplo) que guardan información criptográfica y llevan a cabo operaciones criptográficas.

b) Una tarjeta especialmente diseñada (e involucrada en la aplicación)

Un paso adelante en este asignar atribuciones a la tarjeta inteligente se presenta cuando, al diseñar el Sistema completo en el que se lleva a cabo la aplicación de que se trate, se llega a la conclusión de que la seguridad del Sistema global y la garantía del anonimato en algunas de las operaciones en las que participa la Persona Usuaria del Sistema mejorarían notablemente si algunas de las tareas que tiene que realizar la Aplicación Cliente se llevasen a cabo dentro de la tarjeta.

De esta forma, se puede conseguir que algunos de los datos resultantes de esas operaciones que se realizan en la tarjeta no viajen fuera de ésta, evitando así que pueda adquirir información sobre ellos el propio computador que actúa como terminal del Sistema ante el usuario final. En principio, cabe suponer que todos los agentes telemáticos que intervengan en el proceso sean honrados y que se diseñarán mecanismos técnicos y procedimentales para garantizarlo. Pero una idea útil en seguridad es que la mejor manera de fiarse de todo el mundo es no fiarse de nadie (o fiarse del menor número posible de cosas).

Piénsese además que, con frecuencia, las aplicaciones en las que se van a ver inmersos los ciudadanos, dentro de las comunicaciones presentes en la Sociedad de la Información, serán tales que la máquina ante la que tiene que autenticarse y operar la persona propietaria de la tarjeta pertenecerá a una organización sobre la que ella no tiene ningún tipo de control y, las más de las veces, ante la que tiene que establecer salvaguardas.

Sea como fuere, llegado el caso que estamos suponiendo, los diseñadores de la aplicación pueden contemplar la posibilidad de utilizar una tarjeta inteligente tradicional procurando sacarle el mayor partido posible a las funcionalidades que ofrece (por ejemplo, aprovechando inteligentemente la gestión del sistema de ficheros) y especificando algunas pocas funcionalidades complementarias [Dieg00] que deben ser programadas e incluidas en la memoria interna.

Habida cuenta que el microcomputador insertado en la tarjeta es una máquina de propósito general, esta tarea de programación es perfectamente posible. Pero, ¿quién tiene acceso a llevar a cabo esa programación? Por regla general, las tarjetas convencionales (conformes con la totalidad de la ISO 7816) que pueden adquirirse vienen ya con los programas grabados, y solamente podrán abordar esa tarea quienes tengan acceso a la estructura interna de su sistema operativo y al tipo de instrucciones soportadas por el microprocesador. Y eso es algo que, también por lo general, está fuera del alcance del equipo de diseñadores responsables de la aplicación global.

Así pues, tendrán que pedir a un fabricante de tarjetas (de primera o de segunda fuente) que programe en código nativo (Figura 11.29) esas nuevas tareas e instale esos pequeños añadidos en el programa previamente grabado en la memoria EEPROM de la tarjeta. Por eso decimos que en este caso se trataría de *una tarjeta especialmente diseñada para la aplicación* (aunque esta tarjeta también podría seguirse usando de forma convencional). Obviamente, cuanto más ligeros sean los cambios, más sencillo será realizar esa especialización.

Otra alternativa será usar una Java Card y diseñar las applets necesarias. De eso es de lo que hablaremos a continuación.

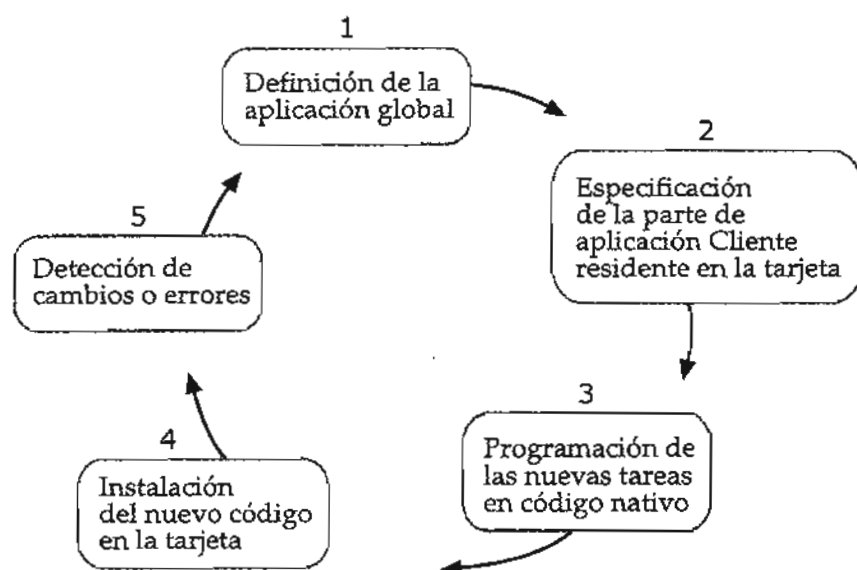


Figura 11.29. Proceso de instalación de nuevas funcionalidades en la tarjeta tradicional.

c) La Aplicación Cliente reside mayoritariamente en la tarjeta

Continuando con el razonamiento expuesto en los últimos párrafos, si lo que se desea es que, por las razones antes aducidas, la tarjeta se haga cargo de la mayoría de las tareas, la opción de implementarla directamente sobre una tarjeta tradicional sólo puede ser llevada a cabo por quienes tengan acceso a la configuración bajo la que está fabricada dicha tarjeta.

A veces, quienes diseñan una aplicación global medianamente evolucionada (por ejemplo, los sistemas que en epígrafes precedentes hemos denominado Sistema1, Sistema2 y SistemaA) deciden que la parte de la Aplicación Cliente que resida en la tarjeta inteligente se comunice con el computador terminal conforme a un protocolo condicionado por resultados intermedios, que use internamente determinadas funciones criptográficas y que sea capaz de manejar estructuras de datos sencillas.

Como hemos dicho, podría optarse por un proceso similar al representado en la Figura 11.29. Suponiendo que se encontrase alguna empresa fabricante dispuesta a hacerlo, esta opción sólo sería viable económicamente si el número de tarjetas modificadas es muy elevado. Además, el programa de la Aplicación Cliente deberá ser suficientemente estable y deberá estar exhaustivamente probado, porque cualquier detección de errores o necesidades de cambio conllevaría (Figura 11.29) iniciar de nuevo el proceso.

Por todo ello, en ese caso, lo razonable es pensar en Tarjetas Java y en programar una o varias applets residentes en la tarjeta.

Según hemos comentado en el apartado *Estructura y funcionamiento de las Tarjetas Java* (epígrafe 11.2), este tipo de dispositivos permite que la aplicación sea programada y compilada en un sistema de desarrollo convencional y posteriormente instalada dentro de la tarjeta para realizar las tareas previstas en cooperación con la máquina virtual y con todos los componentes que constituyen el entorno de ejecución Java Card (JCRE).

En la Figura 11.30 se representa el proceso mediante el cual una applet es generada e introducida en la tarjeta. Como resultado de la compilación del código fuente,

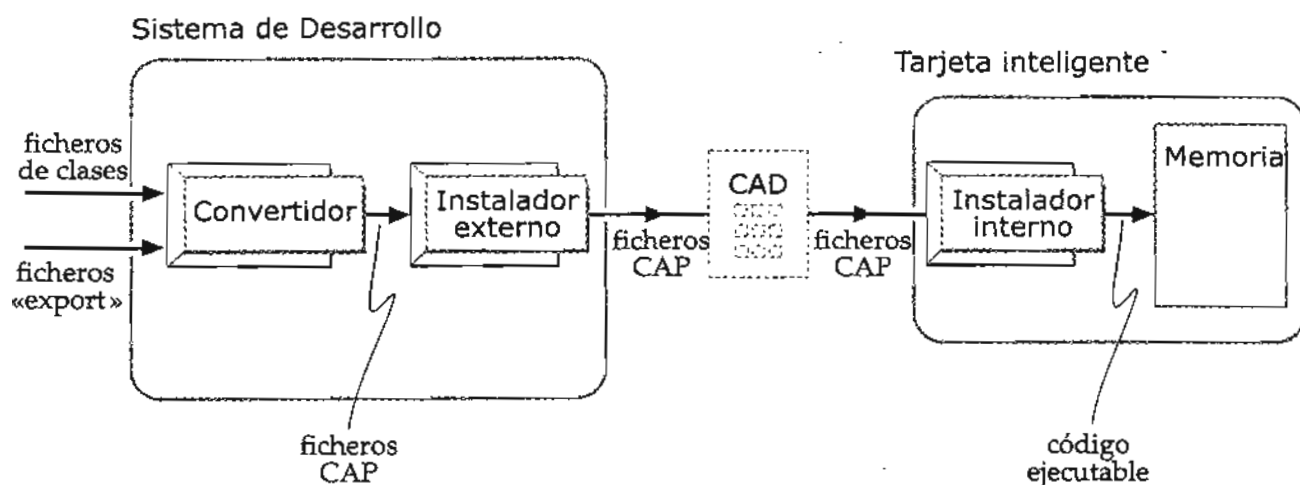


Figura 11.30. Proceso de introducción de una applet en la tarjeta Java.

se habrán generado los ficheros que contienen las distintas clases o módulos en que está descompuesta la aplicación. Para introducirlos en la tarjeta (debido a las particularidades de Java Card con respecto a los entornos de ejecución Java residentes en sistemas informáticos convencionales) es necesario transformarlos a través de un programa *Convertidor* (*Converter*) que genera unos ficheros denominados *CAP* (*converted applet*) que son los que se transfieren al programa instalador para que los almacene en la memoria de la tarjeta.

Para realizar esa transformación, el *Convertidor* no sólo parte de la información contenida en los ficheros de clases, sino también en unos ficheros, denominados ficheros *export*, que contienen la información de otras clases que la aplicación necesita incorporar para su correcta ejecución. Como se refleja en la Figura 11.30, la tarea de instalación se lleva a cabo mediante la cooperación de dos programas instaladores (uno residente en el sistema de desarrollo y el otro en la tarjeta) que se comunican a través del dispositivo de lectura-escritura en la tarjeta (*CAD*).

Aunque lo que antecede es una explicación muy resumida y algo imprecisa de todo el proceso que es necesario llevar a cabo para la generación de las applets necesarias (una descripción más pormenorizada puede encontrarse en las referencias indicadas en el epígrafe 11.2), sí nos sirve para hacernos una idea de las posibilidades y limitaciones que se nos presentan a la hora de abordar el diseño de la Aplicación Cliente y la posibilidad de que, en mayor o menor parte, resida en la tarjeta inteligente.

La gran diferencia entre el computador y otros automatismos consiste en que posibilita que el usuario determine su funcionamiento mediante el diseño y posterior almacenamiento en memoria de un programa. Las tarjetas tradicionales, aunque en realidad están constituidas por un microcomputador programable, en la práctica se presentan ante el diseñador de aplicaciones como un dispositivo bastante «cerrado» y difícil de programar. La Tarjeta Java, en cambio, se nos ofrece como una máquina «casi vacía» que nos permite inventar comportamientos y plasmarlos a través de programas.

11.8. PLATAFORMAS PARA LA DEMOCRACIA DIGITAL

La penetración de las nuevas tecnologías y del uso de los ordenadores se hace cada vez más patente en todos los ámbitos de la vida de los ciudadanos. Sin duda, la

creciente aparición de la Sociedad de la Información debe conllevar la utilización de estas nuevas tecnologías como herramientas que sirvan para impulsar el progreso de los derechos civiles, de la economía y de la sociedad.

La expansión del uso de Internet, tanto en su implantación como en las funcionalidades y servicios utilizados, ha dado lugar al surgimiento en todo el planeta de gran número de iniciativas y nuevas propuestas de infraestructuras telemáticas para ser utilizadas en la administración de nuestros sistemas políticos. No pocas veces estas nuevas propuestas no son sino adaptaciones de antiguas propuestas teóricas, impracticables en el pasado, pero que, apoyándose en las nuevas tecnologías, es posible considerarlas actualmente como viables.

Todas estas iniciativas y propuestas se consideran como elementos que contribuyen a la implantación de lo que se denomina *Democracia Electrónica* o *Democracia Digital* (aquí, por razones similares a las que nos han llevado a aceptar el término *voto telemático*, preferiremos utilizar la segunda de estas dos denominaciones). No obstante, cuando se hace referencia a *Democracia Digital*, nos encontramos con que bajo esta acepción conviven diversas concepciones e interpretaciones. Veamos algunas de ellas.

DIFERENTES TIPOS DE PLATAFORMAS

De forma general, podríamos clasificar los sistemas de *Democracia Digital* en distintas categorías dependiendo de las funcionalidades que sean capaces de soportar las plataformas que pretendan dar servicios de democracia. De menor a mayor rango de funcionalidades, esta clasificación de plataformas podría establecerse de la siguiente manera:

- a) Aquellas que utilizan el término *Democracia Digital* simplemente para referirse a procesos de «ventanilla electrónica», es decir, la sustitución del trámite de papeleo burocrático presencial por un formato telemático (en su concepción mínima, a veces se trata simplemente de páginas web de consulta). Aunque su incorporación representa una mejora indudable en la gestión pública y en la relación de los ciudadanos con la Administración, es evidente que este tipo de plataformas resultan, hoy en día, bastante insuficientes (aunque estadísticamente sean las que mayor presencia tengan).
- b) Otro tipo de plataformas son aquellas en las que se tiende a identificar *Democracia Digital* con la traslación al ciberespacio de los procesos de votación tradicionales. En ellos, los ciudadanos eligen representantes, o bien eligen entre opciones previamente especificadas, bajo el modelo de referéndum. La mayoría de las propuestas en torno a lo que hemos denominado *voto telemático* caen bajo esta acepción, identificando miméticamente *voto* con *democracia*. Aunque en la discusión que se presenta en los epígrafes 11.5 y 11.6 se repite varias veces que el voto telemático debe ofrecer mayores posibilidades que el voto convencional mediante papeletas, lo cierto es que en la descripción de sistemas que allí se realiza predomina esta concepción tradicional del voto.
- c) Una tercera categoría podría estar constituida por aquellas plataformas que, aprovechando las facilidades que dan los protocolos telemáticos, posibilitan el uso de procedimientos de votación que resultarían impracticables usando métodos convencionales. Por ejemplo:

- c1) *Opciones de condicionalidad.* La ciudadanía debería estar en disposición de responder condicionalmente a la consulta que se realiza. Por ejemplo, en preguntas múltiples, debería poderse responder: si ocurre (o gana) x, entonces voto por y; pero en ausencia de x, no apoyo y sino que propongo z. El uso de sistemas informáticos en el escrutinio podría permitir el cómputo de preguntas condicionadas tan complejas como se quisiese. En los sistemas convencionales, la ausencia de condicionalidad en la respuesta ha permitido a ciertos políticos plantear algunos referendos de forma tramposa.
 - c2) *Opciones de elección múltiple o reiterada.* Los sistemas telemáticos podrían permitir un sistema de consultas reiteradas escalonadas en el tiempo, en árbol, en las que, tras una primera votación y el consiguiente debate público, se fuesen planteando otras nuevas consultas en las que la ciudadanía fuera capaz de ir perfilando los detalles y eligiendo sucesivamente. Esto podría permitir dilucidar los elementos de consenso y centrarse en buscar soluciones a los problemas que susciten diferencias. Es evidente que un planteamiento de este tipo es inviable con un sistema convencional mediante papeletas por el coste y el trastorno que representa la movilización de tantas personas y de tantos recursos y locales, pero sería perfectamente realizable si existiese una infraestructura telemática para las votaciones (que podría utilizarse también para otras actividades de «ventanilla electrónica»).
 - c3) *Mezcla de listas cerradas y abiertas.* La lista cerrada tiene la ventaja de respetar la proporcionalidad en cuanto a opciones políticas genéricas, y la desventaja de que limita la capacidad de distinción entre sus miembros. La abierta permite, en cambio, elegir a los candidatos preferidos, si bien no se garantiza la proporcionalidad. El uso de sistemas informáticos para el recuento de los votos permitiría hacer practicable de forma sencilla la combinación de las ventajas de ambos procedimientos.
 - c4) *Opciones de interactividad.* La votación de los ciudadanos debiera basarse en una interactividad simétrica. Es decir, la ciudadanía debería estar en disposición de preguntar sobre las preguntas planteadas e incorporar modificaciones.
- d) Un cuarto grupo de plataformas son aquellas en las que las funcionalidades desempeñadas se sitúan bastante alejadas de los procesos de votación propiamente dichos. En estos casos se entiende la acepción Democracia Digital vinculada a las prácticas que permiten al ciudadano participar en los debates y expresar su opinión a través de plataformas telemáticas. Bajo este planteamiento, el énfasis se pone en los procesos de deliberación, discusión y contraste de la información objeto de debate.

Como compendio de todas ellas, la plataforma que soporte todas las funcionalidades descritas en los cuatro grupos anteriores será la que constituya la categoría más avanzada y completa de aquellas que pretendan proporcionar servicios de Democracia Digital. En este caso, se posibilitaría la comunicación interactiva no sólo entre los ciudadanos y las autoridades que gobiernan, sino entre los ciudadanos mismos, los cuales, tras poder discutir la información, pueden emitir su opinión y, llegado el caso, participar en el proceso de decisión mediante voto u otro tipo de mecanismos de consenso. Es por ello que todas las demás categorías pueden considerarse como subconjuntos de esta última.

IMPLANTACIÓN DE SISTEMAS PARA LA DEMOCRACIA DIGITAL

De lo dicho anteriormente cabe deducir que la implantación de la Democracia Digital no será algo que se resuelva de hoy para mañana ni como consecuencia de decisiones tajantes, sino que será el fruto de un proceso en el que de forma paulatina vayan asumiéndose iniciativas que, analizando los límites y oportunidades que los sistemas de información y las redes telemáticas ofrecen para el desarrollo de la democracia, sirvan para implantar sistemas que favorezcan la interacción, el debate y la más amplia participación de los ciudadanos en todos los aspectos de la vida pública.

En este sentido, es probable que la implantación de alguno de estos sistemas, como puede ser el caso de los sistemas de votación telemática, no se lleve a cabo de forma global y obligatoria en un instante dado, sino que lo esperable es que se vayan creando islas de participación ciudadana en las que la práctica regular del voto telemático evolucionado vaya facilitando la aceptación por parte de los ciudadanos de las ventajas que representa y reduciendo los temores en cuanto a su implantación. En cualquier caso, la implantación de sistemas de Democracia Digital deberá llevarse a cabo dejando la posibilidad de que los ciudadanos que lo prefieran puedan seguir usando métodos convencionales.

La introducción de estos sistemas dependerá de las condiciones políticas y sociales en las que se encuentre la colectividad de que se trate, evaluando en cada caso el balance entre los riesgos y los beneficios que tal implantación conlleve. Por ejemplo, la implantación de un sistema de urnas electrónicas tal como el llevado a cabo en Brasil, en el que se ha visto favorecida la participación de colectividades alejadas de los centros urbanos y con un deficiente nivel de alfabetización, posiblemente resultase inadecuado para ser implantado en otros países en los que los sistemas convencionales mediante papeletas ofrecen mayores garantías de supervisión y auditoría.

En la concepción que aquí estamos asumiendo de considerar la Democracia Digital como un componente más de las actividades que se lleven a cabo dentro de una Sociedad de la Información, las plataformas y los sistemas que diseñen deberán ser lo suficientemente flexibles como para permitir una configuración particularizada conforme a las necesidades de cada situación.

Las herramientas que posibiliten los procesos de participación ciudadana deberán proporcionar frecuentemente servicios de anonimato, lo que conlleva la utilización de mecanismos criptográficos robustos. En el caso de sistemas de votación telemática para la elección de representantes o para la celebración de referendos vinculantes, que pretendan sustituir a los actuales sistemas convencionales de votación, los requisitos de anonimato exigidos serán muy rigurosos (según hemos visto en epígrafes anteriores), pero en otras plataformas de participación la privacidad podrá garantizarse suficientemente con soluciones menos estrictas.

Al igual que en otros escenarios que hemos descrito en anteriores epígrafes, también para la participación de los ciudadanos en procesos de debate y decisión el uso de tarjetas inteligentes puede resultar muy beneficioso a la hora de diseñar protocolos y procedimientos que garanticen la privacidad y, en algunos casos, el anonimato. Aparte de las constatables ventajas de seguridad que proporciona, la tarjeta inteligente conlleva muchos matices simbólicos, ya que es un dispositivo personal y privado que aporta a su titular la sensación subjetiva de tranquilidad que supone que, cuando la retira del CAD en que está alojada, se lleva consigo todos los datos de seguridad personal que le pertenecen.

11.9. SEGURIDAD CÍVICA PARA LA SOCIEDAD DE LA INFORMACIÓN

Con el estudio de las características y posibilidades de aplicación de los servicios de anonimato que hemos estado abordando en el presente capítulo hemos cubierto el arco de los seis principales servicios de seguridad cuya existencia detectamos en el Capítulo 1. En relación con los servicios de anonimato hemos estado analizando lo imprescindible de su uso de cara a la implantación de las aplicaciones telemáticas avanzadas que constituirán una parte importante de los sistemas que soporten las funcionalidades presentes en la Sociedad de la Información.

LOS CRIMINALES ATACAN DE NUEVO

Todo nuestro enfoque ha estado dirigido, conforme a lo que hemos denominado *Seguridad Cívica*, al estudio de los protocolos y mecanismos de seguridad que sirven para protegerse de los riesgos que aparecen en las comunicaciones telemáticas *teniendo en cuenta las necesidades de la vida diaria de los ciudadanos*.

Conviene recordar esto de nuevo porque en torno a la implantación de los servicios de anonimato y de los mecanismos criptográficos en los que se apoyan (por ejemplo, las firmas opacas) existe también un runrún de alarmas y temores catastrofistas acerca del uso perverso que de estos servicios puedan hacer los criminales. Así, se habla del pago anónimo de rescates, del blanqueo de dinero y de otras zarandajas.

Ya se justificó al final del Capítulo 1 el poco interés que para las necesidades de la Seguridad Cívica representan los planteamientos que tienen en cuenta los comportamientos criminales al por mayor, así como la convicción de que para hacer frente a esos comportamientos criminales se puede recurrir a diseñar adecuadamente los esquemas organizativos bajo los que operan los protocolos de seguridad que protegen las comunicaciones.

El problema no está en las redes sino en los planteamientos sociales y políticos que consienten o dan pie a que esos comportamientos existan. No es de recibo poner en duda la aplicación de mecanismos que den soporte a servicios de anonimato que fortalezcan los derechos civiles aduciendo su posible utilización para fines ilícitos. Para que exista dinero negro no es necesario recurrir ni esperar a que se instalen servicios de pago anónimo que preserven la privacidad de las pequeñas compras de los ciudadanos normales. Si existe es porque se consiente o no se ataja adecuadamente.

No obstante, conviene recordar que existen propuestas intelectuales de configuraciones para la provisión de servicios de anonimato que incluyen protocolos específicos con el objetivo de establecer salvaguardas para solucionar, llegado el caso, el posible uso inadecuado de las facilidades ofrecidas.

LA FORTALEZA DE LOS SISTEMAS

Un asunto que también apuntamos en el Capítulo 1 y que ahora, con más conocimiento de causa, conviene resaltar es el relacionado con el nivel de seguridad que deben tener las comunicaciones seguras para conseguir la confianza de los ciudadanos. En la Figura 11.31 se representa de nuevo la figura 1.9 en la que se refleja cómo la confianza surge cuando las medidas de protección introducidas superan «el peso» que tienen los riesgos de vulneración.

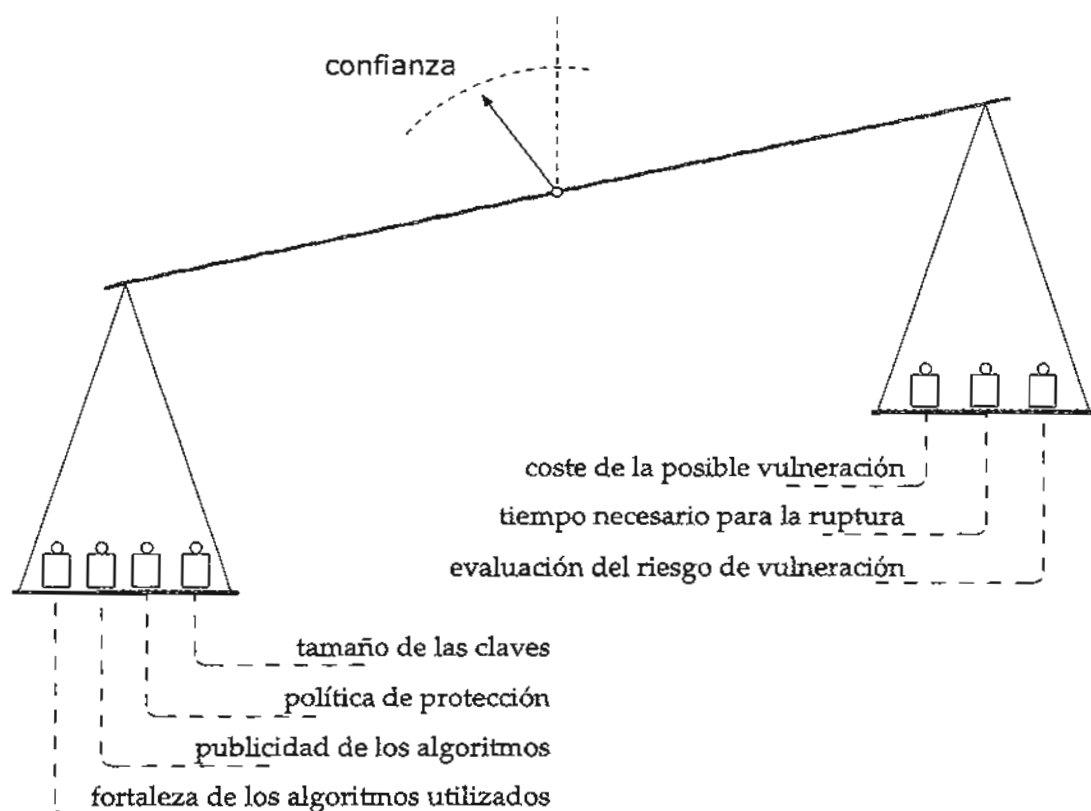


Figura 11.31. Confianza en los sistemas que interesan a los ciudadanos.

Así, por ejemplo, en un sistema de votación telemática que trate de sustituir a los sistemas actuales de votación mediante papeletas para la elección de representantes políticos, las garantías para asegurar que el voto es secreto deben ser elevadísimas, ya que trata de emular, e incluso superar, las garantías ofrecidas por los sistemas convencionales. Pero si de lo que se trata es de una plataforma cívica en la que se intercambian opiniones, y en algunos casos, además de la búsqueda de opciones consensuadas, se recurre a votar propuestas manteniendo el anonimato, las protecciones necesarias pueden ser más ligeras porque es bastante bajo el riesgo de que exista un atacante externo, armado con herramientas y procedimientos de ruptura robustos, interesado en vulnerar la seguridad de ese sistema. El escudo protector debe ser de tal grado que el coste y el esfuerzo necesario para romperlo sea superior al beneficio que pueda obtener el atacante.

De las descripciones que antes hemos hecho acerca de los mecanismos necesarios para conseguir servicios de seguridad robustos se deduce que la fortaleza de los protocolos y de los sistemas puede conseguirse con el concurso balanceado de tres formas de proceder:

- Con medidas organizativas y procedimentales que garantizan la protección de la información. Por ejemplo, en el caso del voto telemático, para la apertura de la Urna se podría implementar un «secreto compartido» procedimental consistente en que la tarjeta inteligente que sirve para provocar que la Urna entregue los votos esté guardada en una caja cerrada y lacrada que solamente pueda desprecintarse en presencia de un determinado número de vocales y de interventores. De forma más genérica, en el Capítulo 10, entre las regulaciones de una CPS, se recogen multitud de referencias a medidas de seguridad que se apoyan en procedimientos organizativos.

- b) Con protección mediante dispositivos resistentes contra manipulaciones (*tamper-resistant* o *tamper-proof*). Los programas que residan en los distintos agentes telemáticos que constituyen el sistema (por ejemplo, de voto telemático o de medios de pago) estarán almacenados en unidades de memoria cuyo contenido no se pueda alterar. Si previamente se ha auditado el contenido de esas memorias, se consigue una confianza bastante significativa en que el comportamiento de esos programas será el correcto y en que el sistema no puede actuar fraudulentamente. La tarjeta inteligente es, como tantas veces hemos insistido, un dispositivo cuya seguridad se apoya en gran medida en la característica de resistencia ante manipulaciones.
- c) Con la generación, por parte de los actores del sistema, de piezas de información criptográficamente robustas y seguras que se puedan presentar como prueba demostrable ante terceros (que pueden, en algunos casos, actuar como jueces o árbitros) en caso de litigio o disconformidad con los resultados deducidos.

Este último tipo de protecciones son las más contundentes y las que, en último extremo, proporcionan un nivel de seguridad más elevado, pero deben utilizarse de forma conjunta y coordinada con las referidas en los dos párrafos anteriores. En cada caso concreto, teniendo siempre en cuenta las necesidades de los ciudadanos y los riesgos que se corren, se elegirán, de entre los tres grupos mencionados, las medidas que inclinen la balanza, de forma proporcionada, para que el sistema pueda ser considerado como digno de confianza.

NECESIDAD DE UNA COOPERACIÓN MULTIDISCIPLINAR

Según anunciábamos al final del Capítulo 1, el objetivo principal de este libro es servir de ayuda a los estudiantes de ingeniería (y a los profesionales) interesados en el diseño, implementación y puesta a punto de sistemas que garanticen, para usos civiles, la seguridad en las redes telemáticas. En este y en anteriores capítulos se ha procurado dejar sentadas las bases para que se puedan adquirir los conocimientos y las habilidades necesarias para ello, aunque quede aún mucho camino por recorrer.

Dentro de la Sociedad de la Información es necesario proporcionar servicios de seguridad que sirvan para garantizar la privacidad y el respeto de los derechos civiles en las comunicaciones que se establezcan, ya sean éstas de carácter social (relacionadas con prácticas de Democracia Digital), comercial (dentro de las distintas facetas de lo que se denomina Comercio Electrónico) o estrictamente personal.

Ante la complejidad de estas demandas, un ingeniero no puede, con la sola ayuda de sus conocimientos tecnológicos, abordar en solitario la solución de los problemas planteados. Antes bien, como ya hemos dicho, será más que conveniente la colaboración de equipos multidisciplinarios que, de forma conjunta, averigüen las necesidades reales de los usuarios [Carr01] y sean capaces de determinar los requisitos que han de cumplir los sistemas que pretendan dar respuesta a ellas.

Será necesario, por tanto, que en el desarrollo de las aplicaciones telemáticas que den respuesta a la necesidades de la Sociedad de la Información, al mismo tiempo que se realicen los trabajos de ingeniería correspondientes, se hagan análisis sociológicos, políticos y jurídicos para determinar la viabilidad de los sistemas. La incorporación de servicios de seguridad en las redes telemáticas ha de servir para garantizar que los derechos y salvaguardas actualmente reconocidos en las comunicaciones con-

vencionales sean respetados también en la proyección y plasmación que éstos tienen en la Sociedad de la Información. Pero, siendo esto necesario, no es suficiente: los nuevos sistemas deberán proporcionar mejores protecciones y mayores cotas de seguridad y de privacidad en las comunicaciones.

No merece la pena desarrollar un sistema telemático técnicamente perfecto que incluya innovaciones notables y use las más avanzadas técnicas si el entorno social al que va dirigido, es decir, los ciudadanos para los cuales ha sido concebido, no confían en él o no responde a sus necesidades reales.

NOTAS AL CAPÍTULO 11

¹ Una propuesta interesante sobre esquemas que utilizan una TTP como ayuda en la generación y verificación de firmas opacas es la formulada en [FrYu94] acerca del opacado de «firmas débiles» (*weak signatures*). En este esquema, dado un mensaje m , la operación de generación de firma y es tan sencilla como realizar la operación $y = (a m + b) \bmod n$, donde a , b y n son parámetros predefinidos. Aunque, como decimos, los algoritmos son fáciles de entender, por cuestiones de espacio y de oportunidad no se han descrito en el texto, ni siquiera a título de ejemplo, porque sería necesario dedicar demasiado espacio para explicar las condiciones de existencia y la corrección de las operaciones que se realizan, tanto para la generación como para la verificación de las firmas. No obstante, el lector interesado puede, sin mucho esfuerzo, entender cumplidamente los planteamientos allí formulados.

² Los autores titulan este artículo como *Fair Blind Signatures*, que es tanto como decir firmas justas o firmas ecuanímes, en contraposición con las firmas opacas sin tercera parte, que serían, por tanto, según ellos, injustas o inadecuadas («en algunas circunstancias», cabe añadir). Más adelante, en el epígrafe 11.3, cuando se aborde el tema de los medios de pago, se comentará de nuevo el problema de los riesgos de las firmas no auditadas. Las realizaciones que se proponen en este artículo se apoyan en procedimientos criptográficos como «Cut-and-Choose» y «Oblivious Transfer» que, aunque no son especialmente complejos, no han sido explicados en el apartado correspondiente de este texto. Por ello, el lector interesado en analizar esos protocolos pormenorizadamente deberá perrecharse previamente del correspondiente bagaje criptográfico, que puede adquirir en un texto especializado en esa materia (por ejemplo, [Schn96]).

³ Las tarjetas de contactos tienen que insertarse en un dispositivo lectura-escritura, CAD, en una posición concreta para conseguir que los contactos coincidan exactamente con los equivalentes del lector. La tarjeta sin contactos no tiene esta restricción porque se comunica mediante una antena alojada en su interior, lo que evita el desgaste que se puede producir en éstos por el excesivo uso o a causa de operar en ambientes hostiles en los que pueda acumularse suciedad en los terminales. La tarjeta dispone de una pequeña pila y/o puede tomar energía rectificando la onda portadora que se modula para conseguir transmitir la información entre el dispositivo de lectura-escritura y la tarjeta. En realidad, la norma por la que se rige esta comunicación es la ISO 10536 (partes 1 a 3), aunque el funcionamiento global es en todo idéntico al que estamos comentando de las tarjetas de contactos conformes con la norma ISO 7816 (partes 1 a 7).

⁴ Para el adecuado entendimiento de las descripciones y comentarios contenidos en este apartado y el siguiente, damos por supuesto que el lector conoce con cierto detalle las características del lenguaje de programación Java y el esquema de ejecución de los programas en una Máquina Virtual Java. En lo que fijaremos nuestra atención en estos apartados es en la particularización de esas características generales al reducido entorno del sistema microcomputador existente en una tarjeta inteligente, con el objetivo último de resaltar las ventajas que ofrecen las Tarjetas Java (en comparación con las que hemos convenido en denominar «clásicas» o tradicionales) en lo que se refiere al desarrollo de aplicaciones telemáticas seguras.

⁵ Una descripción detallada de la estructura, funcionamiento y procedimientos de programación de las Java Cards puede encontrarse en [Chen00]. También puede encontrarse información actualizada en la sección dedicada a *Java Card* dentro del sitio web <http://java.sun.com> de la empresa Sun (que fueron los inventores de este tipo de tarjetas), y en otros foros de Internet dedicados a este asunto.

⁶ En [Carr02] se analiza con más detenimiento este problema y se hace referencia a algunos de estos estudios de investigadores sociales, entre los que cabe destacar las aportaciones de Manuel Castell (*The information Age. Economic, Society and Culture*. Vol I. Oxford. Blackwell Publishers, 1996-1997), Oscar Gandy (*Coming to terms with the panopticon sort*. In *Surveillance, Computers and Privacy*. University of Minnesota Press, 1996) y David Lyon (*Surveillance Society. Monitoring everyday life*. Open University Press. 2001).

⁷ Conviene citar, no obstante, que casi todas las propuestas se apoyan en los trabajos teóricos iniciales de D. Chaum que pusieron las bases para todos los desarrollos posteriores. Dos referencias interesantes para conocer estos planteamientos son [Cha85] y [ChPe93]. Para obtener una idea panorámica de las distintas posibilidades y alternativas que se pueden presentar en la provisión de este servicio, puede consultarse [BuPi89]. Muchas de las aplicaciones existentes, como es el caso de Mondex, no han hecho públicas la estructura y funcionamiento de los protocolos en que se basan y se limitan a declarar las calificaciones de seguridad que han obtenido de distintos organismos de evaluación. Otras, aunque se conoce con más detalle su comportamiento, están sujetas a restricciones de publicación debido a los derechos de patentes (como ya se comentó, Chaum posee varias de ellas).

⁸ Los que aquí hemos denominado *Sistema1* y *Sistema2* se corresponden con dos propuestas experimentales, en las que participó directamente el autor de este texto. El *Sistema1* se corresponde con [Dieg00] y con una versión simplificada y retocada que está recogida en [Carr02]. El *Sistema2* aquí descrito es una simplificación del descrito en [Gome00] modificado para adaptarlo al esquema *on-line* del *Sistema1*, y tiene partes comunes con alguno de los ejemplos descritos en [Schn96].

⁹ Realmente, las compras para las que interesa mantener el anonimato son las que conlleven cantidades de dinero pequeñas o medianas, que son las que sirven para trazar los hábitos y modos de vida de los ciudadanos (qué, dónde y cuándo). Precisamente, las compras importantes (un automóvil, una casa) no necesitan anonimato porque son transacciones de por sí bastante visibles. Es más, a los ciudadanos corrientes les interesa que esas transacciones **no sean anónimas**, sino, por contra, que exista la máxima transparencia en su adquisición y titularidad (y en el pago de los correspondientes impuestos).

¹⁰ Una descripción resumida de las experiencias de voto electrónico más significativas que se han llevado a cabo hasta la fecha puede encontrarse en [GoCa03]. Asimismo, en este artículo se hace una clasificación de este tipo de sistemas y se analizan las propuestas más relevantes que se apoyan en criptografía avanzada y en redes telemáticas (lo que daremos en llamar *voto telemático*). (Puede consultarse en www.vototelematico.diatel.es.)

¹¹ Este método de trabajo que aquí se expone es una esquematización del seguido durante el desarrollo de un proyecto de investigación sobre voto telemático (el proyecto VOTES-CRIPT [CGMP02] [CGC03]) cuyo desarrollo se llevó a cabo mediante un equipo multidisciplinar compuesto por investigadores pertenecientes tanto al campo de la ingeniería telemática como al campo sociopolítico y jurídico. La tesis de partida de este proyecto es que la votación telemática es imprescindible abordarla desde equipos multidisciplinares, necesarios no sólo en la fase de diseño y desarrollo sino también en la fase de implementación y puesta a punto. Dentro de este proyecto se han desarrollado el *Sistema Votescript* de votación mediante cabina y el denominado *Sistema VERA* (*Votación Electrónica para los Residentes Ausentes*) vinculado a un proyecto conjunto de la Subdirección General de Política Interior del Ministerio del Interior de España con la FNMT (Fábrica Nacional de Moneda y Timbre). En base a las especificaciones de VERA, la FNMT-RCM desarrolló un sistema propio de votación, de cuyo prototipo se realizaron pruebas de campo en El Hoyo de Pinares (Ávila) en marzo de 2003. Este prototipo no incluía el proceso de verificación individual ni la presencia de comprobantes de votación. En la web del proyecto (www.vototelematico.diatel.es) puede encontrarse información sobre los artículos reseñados y sobre los trabajos sociológicos

llevados a cabo. Una información más detallada sobre estos últimos puede encontrarse en www.ucm/info/demodigi.

¹² Los que aquí hemos denominado SistemaA y SistemaB son una simplificación de dos sistemas de votación telemática diseñados dentro del proyecto VOTESCRIPT citado en la nota anterior. El SistemaA es una simplificación del denominado *Sistema Votescript* de votación mediante cabina y el SistemaB lo es de una versión menos segura del anterior que utiliza un computador convencional como punto de votación y acceso al Sistema.

¹³ Para implementar este *Sobre Seguro* puede utilizarse un mecanismo habitualmente denominado *canal seguro* que es una mejora del mecanismo denominado *sobre* o *envoltura digital* (*digital envelope*) que analizamos en el Capítulo 5. Consiste en: *a*) generar una cadena aleatoria, *b*) generar una clave de sesión, *c*) concatenar la cadena y el mensaje a ocultar, *d*) obtener el *hash* (MAC, conforme a la nomenclatura usada en el Capítulo 8) de la concatenación anterior, *e*) concatenar el MAC, la cadena aleatoria y el mensaje, *f*) cifrar lo anterior con un algoritmo simétrico haciendo uso de la clave de sesión generada, *g*) cifrar la clave de sesión con la clave pública del destinatario, y *h*) enviar al destinatario el resultado anterior junto con el criptograma resultante del cifrado simétrico.

Pueden utilizarse también otros mecanismos que eviten el cifrado simétrico procediendo de la siguiente forma: *a*) generar una cadena aleatoria, *b*) concatenar la cadena y el mensaje a ocultar, *c*) obtener el MAC de ese conjunto mediante una operación *hash*, *d*) concatenar el MAC, la cadena aleatoria y el mensaje, y *e*) cifrar todo lo anterior con la clave pública del destinatario. El objetivo de la cadena aleatoria es el de evitar los ataques denominados *de fuerza bruta* comentados en la nota a pie de página (se podría adivinar el voto por tanteo de todas las opciones posibles). Por otra parte, el MAC garantiza la integridad y evita que aparezca el siguiente ataque: Un atacante podría alterar los votos que se envían desde el Punto de Votación a la Urna (sin ningún provecho, sólo por fastidiar). La Urna no detectaría la falta de integridad en lo recibido, por lo que lo aceptaría. Al final de la votación, durante la fase de recuento, se detectaría que la pieza de información no corresponde a un voto válido y, por lo tanto, sería desechada, consiguiendo así el atacante invalidar votos.

¹⁴ Un esquema como este de la Figura 11.26, o bien una manipulación del esquema representado en la Figura 11.24, que permitiese establecer bases de datos con relaciones pormenorizadas de personas y de sus huellas identificativas (o palabras de paso) haría las delicias de los defensores de no-sé-bien-qué, que tanto abundan, y que tratan de protegernos contra nuestra voluntad a costa de pulverizar nuestra privacidad. La evolución de las tecnologías que posibilitan la implantación de controles biométricos en lugares públicos o de acceso semipúblico, que condicionan los movimientos de la ciudadanía en su conjunto, es algo ajeno al uso de las técnicas biométricas que aquí proponemos, aunque tengan en común una misma base teórica. El debate sobre las ventajas e inconvenientes de esas prácticas cae fuera del alcance de este texto.

¹⁵ Una solución de este tipo es la adoptada en el sistema VERA [CGC03] del cual el aquí denominado SistemaB es una simplificación didáctica, según se refiere en nota anterior. En [LCP99] se describe otro sistema en el que se utiliza una applet que se descarga de un servidor central y utiliza una tarjeta inteligente como testigo de seguridad y auxiliar en la realización de operaciones criptográficas sencillas. La ventaja de este tipo de soluciones es que el desarrollo de la aplicación, en cuanto a la utilización de la tarjeta inteligente se refiere, resulta bastante sencillo.

RESEÑA SOBRE AUTORIZACIONES

A lo largo de los distintos capítulos de este libro se ha hecho referencia reiterada a las normas generadas por los distintos organismos de normalización y, frecuentemente, se ha reproducido algún segmento de las definiciones ASN.1 en ellas contenidas. Asimismo, también se ha recomendado reiteradamente al lector que recurra directamente a las normas para conocer los detalles y los pormenores de las materias especificadas, ya que la referencia y las explicaciones que aquí se incluyen sólo pretenden cubrir el objetivo didáctico de ayudar a entender el alcance y el significado de lo descrito en ellas.

La Asociación Española de Normalización y Certificación, AENOR, ha tenido la gentileza de autorizar expresamente la publicación de esas referencias y solicita la inclusión del siguiente texto:

«Las normas ISO/IEC 9594-8:2001 e ISO/IEC 18014:2002 se han reproducido parcialmente con autorización de AENOR. Este material no puede ser vendido ni distribuido a terceros. Cualquier cesión o reproducción parcial o total de los términos incluidos en las normas, por cualquiera de los medios de difusión existentes, sin el consentimiento por escrito de AENOR, queda absolutamente prohibida.»

También el European Telecommunication Standards Institute, ETSI, ha autorizado la inclusión en este libro de algunas figuras y de parte de la especificación contenidas en la norma TS 101 733 v1.4.0 relativa a la firma electrónica, solicitando la inclusión del siguiente párrafo aclaratorio:

«© ETSI 2002. Further use, modification, redistribution is strictly prohibited. The standards are available from <http://pda.etsi.org/pda/> and <http://www.etsi.org/eds/>»

Por su parte, ISOC, en la declaración que publica acerca de los derechos de propiedad y de las condiciones de publicación de información contenida en sus normas, aclara que: *«To protect the integrity of RFC publication, all RFCs are published with an ISOC (Internet Society) copyright statement»*. Asimismo, añade que: *«This copyright notice was designed to ensure that Request For Comments(RFC) documents will have the widest possible distribution»*. En lo que se refiere a la copia y distribución de trozos de una RFC, la condición que exigen es que se cite adecuadamente la autoría y el origen de esa norma, cosa que hemos hecho en los capítulos precedentes cada vez que se ha utilizado uno de esos segmentos, al tiempo que agradecemos desde aquí a ISOC y a los distintos autores de las RFCs (que también hemos citado en cada caso) las facilidades que ofrecen para que las personas que se encuentran en su fase de aprendizaje tengan acceso al material contenido en dichos documentos.

BIBLIOGRAFÍA

- [ABA95] American Bar Association, *Digital Signature Guidelines: Legal Infrastructure for Authorities and Electronic Commerce*, 1995 (citado en [ChFo99]).
- [ArTu02] A. Aresenault and S. Turner, *Internet X.509 Public Key Infrastructure: Roadmap*. PKIX Working Group. July 2002.
- [Beth00]. T. Beth, *El estado de la criptografía de clave pública a la vista de las posibilidades de la Computación Cuántica*, en «Criptología y Seguridad de la Información», editoras: P. Caballero y H. Hernández. Ed. RaMa, 2000.
- [BuPi89] H. Bürk and A. Pfitzmann, *Digital Payment Systems Enabling Security and Unobservability*, Karlsruhe University, 1989.
- [Carr99] J. Carracedo and J. D. Carracedo, *Use of Security Protocols for Privacy and Anonymity Protection in the Internet Communications*. En *Exploring Cyber Society*, Armitage, J. & Roberts J., editors. University of Northumbria at Newcastle Press, 1999.
- [Carr01] J. Carracedo y J. D. Carracedo, *Telemática y Sociología. Apuntes para una investigación multidisciplinar: Tarjetas de Crédito Anónimas y Democracia*. I Congreso Iberoamericano de Telemática. Cartagena de Indias (Colombia), julio 2001.
- [Carr02] J. Carracedo y J. D. Carracedo, *Tarjetas de Crédito Anónimas para garantizar la privacidad de los ciudadanos*. I Congreso Iberoamericano de Seguridad Informática. Morelia (México), febrero 2002.
- [Cast96] M. Castells, *The information Age. Economic, Society and Culture*. Vol I, II & III. Oxford. Blackwell Publishers, 1996-1997.
- [CGC03] J. Carracedo, A. Gómez y J. D. Carracedo, *Sistema VOTESCRIPT: Una propuesta innovadora desarrollada para resolver los problemas clásicos de la votación electrónica*. II Congreso Iberoamericano de Seguridad Informática. Ciudad de México, octubre 2003.
- [CGMP02] J. Carracedo, A. Gómez, J. Moreno, E. Pérez y J. D. Carracedo, *Votación electrónica basada en criptografía avanzada (Proyecto VOTESCRIPT)*. II Congreso Iberoamericano de Telemática. Mérida (Venezuela), septiembre 2002.
- [Cha83] D. Chaum, *Blind Signature Systems*, Proceedings of Crypto'83, Plenum p. 153, 1983.
- [Cha85] D. Chaum, *Security Without Identification. Transaction system to make Big Brother obsolete*, Communication of the ACM, v. 28, n. 10, Oct 1985, pp. 1030-44, 1985.
- [Chen00] Z. Chen, *Java Card Technology for Smart Cards*. Addison-Wesley 2000.
- [ChFo99] S. Chokhani and W. Ford, *Internet X.509 Public Key Infrastructure. Certificate Policy and Certification Practices Framework*, RFC 2527, March 1999.
- [ChPe93] D. Chaum and T. Pedersen, *Transferred Cash Grows in Size*, CWL, Netherlands, 1993.
- [DaRi99] J. Daemen and V. Rijmen, *The Rijndael Block Cipher. AES Proposal*. Document version 2, Sept. 99. Documento explicativo de 45 páginas que se puede encontrar en la página web dedicada a este algoritmo.
- [DBP96] H. Dobbertin, A. Bosselaers and B. Preneel, *RIPEMD-160: A Strengthened Version of RIPEMD*. Proceedings, Third International Workshop on Fast Software Encryption. Springer-Verlag, 1996.
- [Denn94] D. E. Denning, *The Key Escrow Encryption Technology*. Recogido en Hoffmam, E. J., editor, «Building in Big Brother, The Cryptographic Policy Debate». Springer-Verlag, 1995.
- [Dieg00] J. de Diego, *Servicios de anonimato para redes seguras. Tarjeta de Pago Anónima*. Proyecto Fin de Carrera. Autor: Javier de Diego; tutor: J. Carracedo. EUIT de Telecomunicación, Universidad Politécnica de Madrid, 2000.

- [DiHe76] W. Diffie and H. Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory, vol. 22, Nov. 1976.
- [DiLa98] W. Diffie and S. Landau, *Privacy on the Line*. The MIT Press, 1988.
- [DoD83] *Trusted Computer System Evaluation Criteria (TCSEC)*, DoD 5200.28-STD.
- [FaHu02] S. Farrell and R. Housley, *An Internet Attribute Certificate Profile for Authorization*. RFC 3281, April 2002.
- [FeKu92] D. F. Ferraiolo and D. R. Kuhn, *Role-Based Access Controls*, Proceedings of the 15th NIST-NSA National Computer Security Conference, Baltimore, October 1992.
- [FeMa02] J. L. Ferrer y A. Martínez, *El acuse de recibo (o no repudio de destino)*. Revista SIC, septiembre 2001.
- [FGHM00] A. Fúster, D. de la Guía, L. Hernández, F. Montoya y J. Muñoz, *Técnicas Criptográficas de protección de datos*. Ed. RA-Ma, 2000.
- [FKC03] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli, *Role-Based Access Controls*. Artech House, Computer Security Series, 2003.
- [FoBa97] W. Ford and M. S. Baum, *Secure Electronic Commerce*. Prentice-Hall, 1997.
- [Ford94] W. Ford, *Computer Communications Security*. Prentice-Hall, 1994.
- [FOO93] T. Fujioka, K. Okamoto and A. Otha, *A Practical Secret Voting Scheme for Large Scale Elections*, Advances in Cryptology, AUSCRYPT'92, Lecture Notes in Computer Science 718. Springer-Verlag, Berlin, pp. 244-251 (1993).
- [FrYu94] M. Franklin and M. Yung, *The Blinding of Weak Signatures*, Eurocrypt'94 Proceedings, 1994.
- [GoCa03] A. Gómez y J. Carracedo, *Del voto electrónico al voto telemático: clasificación y valoración de las propuestas existentes*. VI Congreso de Ciencia Política y de la Administración. Barcelona, septiembre de 2003 (puede consultarse en www.vototelematico.diatel.upm.es).
- [Gome00] P. Gómez, *Servicios de Anonimato para redes seguras. Tarjetas de Crédito Anónimas*. Proyecto Fin de Carrera. Autor: Pablo Gómez Espasandín, tutor: J. Carracedo. EUIT de Telecomunicación, Universidad Politécnica de Madrid. Octubre de 2000.
- [Hoff95] L. J. Hoffman, editor (45 contributors), *Building in the Big Brother. The Cryptographic Policy Debate*. Springer-Verlag, 1995.
- [HPFS02] R. Housley, R., W. Ford and D. Solo, *Internet X.509 Public Key Infrastructure. Certificate and CRL Profile*. PKIX Working Group. April 2002. Primero fue un documento de trabajo y posteriormente se catalogó como RFC 3280.
- [Jac01] E. Jacob, *Una Metodología para el estudio del cumplimiento de Políticas de Seguridad en Sistemas de Información*. Tesis Doctoral, Universidad del País Vasco, 2001.
- [Jara01] M. Jara, *Aplicación de una Metodología de Análisis y Gestión de Riesgos*. Proyecto Fin de Carrera. Autora: M. Jara; tutor: J. Carracedo. EUIT de Telecomunicación, Universidad Politécnica de Madrid, 2001.
- [KnDo69] Z. Knuth and E. Donlad, *Seminumerical Algorithms*, vol. 2, «The Art of Computer Programming», Addison Wesley, Reading Mass, 1969.
- [Koch96] P. Kocher, *Timing Attack on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*. Proceedings, Cryoto '96, Springer-Verlag, 1996.
- [LaMa91] X. Lai and J. Massey, *A Proposal for a New Block Encryption Standard*. Advances in Cryptology - EURODRYPT'90 Proceeding. Springer-Verlag, 1991.
- [LCP99] L. López, J. Carracedo e I. Pau, *Proyecto TIPI: Certificación en Internet con uso de tarjeta inteligente*. Revista SIC: Seguridad, Informática y Comunicaciones. Nº 37, noviembre 1999.
- [LMP94] S. H. Low, N. F. Maxemchuck and S. Paul, *Anonymous Credit Cards*, Proceeding of the Second annual ACM Conference on Computer and Communication Security, ACM Press.
- [LoCa97] L. López y J. Carracedo, *Hierarchical Organization of Certification Authorities for Secure Environments*. The Internet Society 1997 Symposium on Network and Distributed Systems Security. IEEE Computer Soc. Press, Los Alamitos, CA. USA.
- [Lope98] L. López, *Organización y jerarquización de Autoridades de Certificación para la provisión de servicios de seguridad en Redes Telemáticas*. Tesis Doctoral. Universidad Politécnica de Madrid, 1998.

- [Mage97] MAGERIT, Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. Ministerio de Administraciones Públicas. Ministerio de la Presidencia. BOE. Madrid, 1997.
- [Mene93] A. Menezes, *Elliptic Curve Public Key Cryptosystems*. Boston: Kluwer Academic Publisher, 1993.
- [MPW92] C. Mitchels, F. Piper and P. Wild, *Digital Signatures*. In: Simmons, G, ed. *Contemporary Cryptology: The Science of Information Integrity*. IEEE Press, 1992.
- [Myer99] M. Myers, R. Ankney, A. Malpani, S. Galperin and C. Adams, *X.509 Internet Public Key Infrastructure. Online Certificate Status Protocol – OCSP*. RFC 2560, June 1999.
- [PPR03] D. Pinkas, N. Pope and J. Ross, *Policy Requirements for Time-Stamping Authorities*. Internet Draft. PKIX WG. April 2003.
- [Rami98] J. Ramió, *Aplicaciones Criptográficas*, Ed. EU Informática (Universidad Politécnica de Madrid), 1998.
- [RSA78] R. Rivest, A. Shamir and L. Adleman, *A Method for Obtaining Digital Signatures and Public Key Cryptosystems*. Communications of The ACM, vol. 21, 1978.
- [SaPi94] R. S. Sandhu and P. Samarati, *Access Control: Principles and Practices*. IEE Communication Magazine, September 1994.
- [SCFY96] R. S. Sandhu, E. J. Coyne, H. L. Feinstein and C. E. Youman, *Role Based Access Control Models*. IEEE Computer 29, 2 (Feb. 1996).
- [Schn96] B. Schneier, *Applied Cryptography*, Second Edition. John Wiley and Sons, 1996.
- [Sham79] A. Shamir, *How to Share a Secret*, Communications of the ACM, v. 24, n. 11, Nov. 1979.
- [Shan48] C. Shannon, *A Mathematical Theory of Communication*, Bell System Technical Journal, 1948.
- [Shan49] C. Shannon, *Communication Theory of Secrecy Systems*, Bell System Technical Journal, 1949.
- [SPC95] M. Stadler, J. M. Piveteau and J. Camenish, *Fair Blind Signatures*. Advances in Cryptology, Eurocrypt'95, pp. 209-219, Springer-Verlag, 1995.
- [Stal99] W. Stalling, *Cryptography and Network Security, Principles and Practice*, Second Edition. Prentice Hall, 1999.
- [Tane89] A. Tanenbaum, *Computer Networks*, Prentice-Hall, 1989.
- [Tane97] A. Tanenbaum, *Redes de computadoras*, tercera edición. Prentice-Hall, 1997.
- [UGJF99] J. Unzila, I. Goirizelaia, E. Jacob y A. Ferro, *Visión general de las técnicas de marcas de agua (watermarking) y sus aplicaciones*. II Jornadas de Ingeniería Telemática. Libro de ponencias. Edit.: Universidad Carlos III, Madrid, 1999.
- [ViMa96] Visa and Mastercard, *Secure Electronic Transaction (SET) Specification*. <<http://www.visa.com>> y <<http://www.mastercard.com>>
- [Zime94] P. Zimmerman, *Pretty Good Privacy (PGP) Public Encryption for the Masses. PGP User's Guide Version 2.6.1.*, Ago., 1994.

ÍNDICE ANALÍTICO

- ACI, 374
- ACL, 376
- AES, 103
- Algoritmos
 - A5, 121
 - complejidad de, 82
 - DES, 94
 - Diffie-Hellman, 161, 163
 - Hash, 328
 - IDEA, 98
 - RC2, 101
 - RC4, 123
 - RC5, 102
 - Rijndael, 105
 - RSA, 153
- Análisis de riesgos, 36
- Árbol de información del directorio (DIT), 265
- Aritmética modular, 68, 70
- ASN.1, 205
- Ataques, 6
- Autenticación
 - de entidades, 312
 - fuerte, 316
 - por desafío, 315
 - simple, 313
 - del votante, 503
- Autoridad
 - de certificación (CA), 41, 186
 - de sellado de tiempo (TSA), 41, 303
 - registro, RA, 41, 251
- BER (ASN1), 217
- Bit String (ASN1), 209
- CA, 41, 186
- Caballo de Troya, 9
- CAD, 474
- Cajas-S, 96
- Cálculo de inversos, 72
- Camino de certificación, 191
- Campos de Galois, 68, 77
- Cantidad de información, 62
- Capacidades (control de acceso), 379
- CBF, 91
- CER (ASN1), 219
- Certificado de atributos, 299, 388
- CFB, 91
- Choice (ASN1), 214
- Cifrado múltiple, 113
- Cifrador(es)
 - de bloque, 90
 - de flujo, 115
 - de Geffe, 120
 - de Vernam, 116
- Clave de sesión, 151, 175
- CMS (Cryptographic Message Syntax), 355
- Complejidad de algoritmos, 82
- Computación cuántica, 104
- Confianza, 29, 31, 536
- Confusión y difusión, 67
- Conjunto de estipulaciones, 422
- Cortafuegos, 51
- CPS, 282, 415
- Criptografía, 20
 - clásica, 85
 - de curvas elípticas, 165
 - para el anonimato, 458
- Criptosistemas, 21
 - A5, 121
 - de clave pública, 23, 135
 - secreta, 21, 87
 - de secreto perfecto, 66
 - DES, 94
 - IDEA, 98
 - RC2, 101
 - RC4, 123
 - RC5, 102
 - Rijndael, 105
 - RSA, 153
 - Skipjack, 102
- CRLs, 239, 292
- DAC, 383
- Default (ASN1), 214
- Democracia digital, 531
- Denegación del servicio, 8
- DER (ASN1), 218
- DES, 94
- Dinero digital, 481, 486, 488
- Directorio X.500, 194, 261
- Distancia de unicidad, 63, 67
- Distribución de claves
 - públicas, 181
 - simétricas, 171
- DIT (árbol de información del directorio), 265

- Divulgación del contenido, 7
- Dominio de seguridad, 32
- DPD, 368
- DPV, 368
- DUA (agente de usuario de directorio), 263
- ECB, 90
- ECC, 90
- EESSI, 346
- Entropía, 63
- Envoltura digital, 151
- ES (Electronic Signature), 350
- Esteganografía, 28, 123
- Estipulaciones para la certificación, 422
- Etiquetas (control de acceso), 381
- ETSI, 59, 346
- Euler, 74, 75
- Extensiones
 - de la CRL, 240
 - del certificado
 - de atributos, 396
 - X.509, 197, 201, 227, 240
- Fermat, 74
- Firma
 - digital, 146, 330
 - electrónica, 331, 337
 - ES, 350
 - ES-C, 235
 - ES-T, 351
 - especificación formal, 355
 - formatos, 346
 - validación, 365
 - opaca o firma a ciegas, 458
 - RSA, 148
- Función indicadora de Euler, 74
- Geffe (Cifrador de Geffe), 115
- GF (qn), 77
- IDEA, 98
- Identificadores de objeto, 210
- IETF, 58
- Información e incertidumbre, 62
- Infraestructura
 - de certificación, 42, 247, 251
 - MISSI, 275
 - modelos flexibles, 279
 - organización de CAs, 189
 - PEM, 269
 - SET, 276
 - de clave pública (PKI), 42, 247
 - de gestión de privilegios (PMI), 299
- de seguridad, 42, 245, 248
- Inversos, 72
- IPRA, 270
- ISO, 57
- ITU-T, 57
- Java Card, 477
- JCVM, 477
- Kerberos, 322
- Key Escrow, 176
- LDAP, 265
- LFSR, 118, 120
- Listas
 - de certificados revocados (CRLs), 239, 292
 - de control de acceso (ACLs), 376
- MAC
 - (Mandatory Access Control), 384
 - (Message Authentication Code), 318, 319
- Magerit, 36
- Marcas de agua, 128
- MD-2, MD-4, MD-5, 328
- Mecanismo
 - de firma digital, 146
 - de seguridad, 11
- Medios de pago, 488
- Mensaje firmado, 458
- MISSI, 275
- Modificación de mensajes, 8
- Modo
 - CBF, 91
 - CFB, 91
 - ECB, 90
 - ECC, 90
 - OFB, 93
- No repudio, 15, 400
 - de depósito, 404
 - de entrega, 406
 - de envío, 404
 - de origen, 403
- Nombres de las entidades, 196, 200, 268, 289
- Normas, 55
- Object Identifier, 210
- OCSP, 290, 296
- Octet String (ASN1), 209
- OFB, 93
- Optional (ASN1), 214

Organismos de normalización, 55
Organización de CAs, 189, 254

PEM 269

PER (ASN1), 219

PKI, 42, 247

PMI, 299

Política de seguridad 27, 413

CPS, 282, 415

de certificación, 282, 415

de control de acceso, 382

de firma, 440, 443

de sellado de tiempo, 435, 437

en las extensiones del certificado, 201

estipulaciones para la certificación, 282, 288

políticas permitidas, 281

protocolo de, 11

restricciones, 286, 288

Prácticas de sellado de tiempo, 438

Protocolo de seguridad, 11

Puerta falsa, 9

Puntos de distribución de CRLs, 293

RA, 41, 251

Rastreabilidad del dinero digital, 494

RBAC, 383, 286

Recuperación de claves, 176

Redundancia, 64

Repetición del contenido, 7

Repositorio, 251, 261

Requisitos de usuario, 35, 497

Revocación de certificados, 237

Riesgos, 36

Rijndael, 105

RIPEMD, 329

Secreto

compartido, 469

dividido, 467

Secuencia cifrante, 117

Seguridad Cívica, 26, 86, 129, 535

Sequence (ASN1), 212

Servicios de seguridad, 11, 41, 249

Anonimato, 18, 457

Autenticación 13, 311

Control de acceso, 17, 331

Integridad, 25

No Repudio, 15, 400

Ubicación de los servicios, 47

SET, 276

SET (ASN1), 213

SHA, 329

Signed Data, 358

Sobre digital, 151

Suplantación de personalidad, 7

Tarjeta(s)

de pago anónima, 485, 489

inteligentes, 43, 472, 522

Java, 477

Teorema

de Euler, 74

de Fermat, 74

Trampilla, 9

Triple cifrado, 114

TSA, 41, 303

TTP, 39, 249

Ubicación de los servicios de seguridad, 47

Vernam, cifrador de, 116

Voto

electrónico, 495

telemático, 495

autenticación y autorización del votante,
503

determinación de requisitos, 497

emisión del voto, 507

interventores, 516

recuento de votos, 510

verificación

global, 516

individual, 512

X.509, 194

dentro de línea (in-line), 41

en línea (on-line), 40