UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T X.843

(10/2000)

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

SERIE X: REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS

Seguridad

Tecnología de la información – Técnicas de seguridad – Especificación de servicios de tercera parte confiable para soportar la aplicación de firmas digitales

Recomendación UIT-T X.843

(Anteriormente Recomendación del CCITT)

RECOMENDACIONES UIT-T DE LA SERIE X

REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS

REDES PÚBLICAS DE DATOS	
Servicios y facilidades	X.1–X.19
Interfaces	X.20–X.49
Transmisión, señalización y conmutación	X.50–X.89
Aspectos de redes	X.90–X.149
Mantenimiento	X.150–X.179
Disposiciones administrativas	X.180–X.199
INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Modelo y notación	X.200–X.209
Definiciones de los servicios	X.210–X.219
Especificaciones de los protocolos en modo conexión	X.220–X.229
Especificaciones de los protocolos en modo sin conexión	X.230–X.239
Formularios para declaraciones de conformidad de implementación de protocolo	X.240–X.259
Identificación de protocolos	X.260–X.269
Protocolos de seguridad	X.270–X.279
Objetos gestionados de capa	X.280–X.289
Pruebas de conformidad	X.290–X.299
INTERFUNCIONAMIENTO ENTRE REDES	
Generalidades	X.300–X.349
Sistemas de transmisión de datos por satélite	X.350–X.369
Redes basadas en el protocolo Internet	X.370–X.399
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400–X.499
DIRECTORIO	X.500–X.599
GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS	
Gestión de redes	X.600–X.629
Eficacia	X.630–X.639
Calidad de servicio	X.640–X.649
Denominación, direccionamiento y registro	X.650–X.679
Notación de sintaxis abstracta uno	X.680–X.699
GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Marco y arquitectura de la gestión de sistemas	X.700–X.709
Servicio y protocolo de comunicación de gestión	X.710–X.719
Estructura de la información de gestión	X.720–X.729
Funciones de gestión y funciones de arquitectura de gestión distribuida abierta	X.730–X.799
SEGURIDAD	**X.800–X.849**
APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Compromiso, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.899
PROCESAMIENTO DISTRIBUIDO ABIERTO	X.900–X.999

Para más información, véase la Lista de Recomendaciones del UIT-T.

NORMA INTERNACIONAL ISO/CEI 15945

RECOMENDACIÓN UIT-T X.843

TECNOLOGÍA DE LA INFORMACIÓN – TÉCNICAS DE SEGURIDAD – ESPECIFICACIÓN DE SERVICIOS DE TERCERA PARTE CONFIABLE PARA SOPORTAR LA APLICACIÓN DE FIRMAS DIGITALES

Resumen

Esta Recomendación | Norma Internacional define los servicios requeridos para soportar la aplicación de firmas digitales para el no repudio de la creación de un documento. Puesto que ello implica la integridad del documento y la autenticidad del creador, los servicios descritos pueden combinarse también para implementar los servicios de integridad y autenticidad.

Orígenes

La Recomendación UIT-T X.843, preparada por la Comisión de Estudio 7 (1997-2000) del UIT-T, fue aprobada por la Asamblea Mundial de Normalización de las Telecomunicaciones (Montreal, 27 de septiembre-6 de octubre de 2000). Se publica también un texto idéntico como Norma Internacional ISO/CEI 15945.

PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución 1 de la AMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

ÍNDICE

Introducción

Actualmente, el desarrollo de la tecnología de la información, así como el de la infraestructura mundial de comunicaciones, brinda la posibilidad de implementar el comercio electrónico con dimensiones económicamente pertinentes. Las firmas digitales constituyen una técnica importante para añadir seguridad a estas aplicaciones comerciales y a otros campos de aplicación que tienen necesidad de efectuar transacciones electrónicas válidas legalmente.

Las firmas digitales son adecuadas para asegurar la integridad de los datos transmitidos y la autenticación de los participantes en las transacciones. Pueden suministrar una representación análoga de la firma manuscrita para pedidos, ofertas y contratos digitales. La propiedad más importante de las firmas digitales en este contexto es que un persona que ha firmado un documento no puede tener éxito en la denegación de este hecho. Esta propiedad se denomina "no repudio de la creación" de un documento.

En varios países y en contextos internacionales, se está impulsando la legislación relativa a las firmas digitales a fin de que soporte el desarrollo del comercio electrónico y otros campos de aplicación que precisan de transacciones electrónicas legalmente válidas.

Existen diversas normas que especifican las firmas digitales así como su uso en diferentes aplicaciones como el no repudio o la autenticación. Se están introduciendo o planificando diversas aplicaciones comerciales y TTP que ofrecen servicios en conexión con firmas digitales Para la utilización efectiva a nivel mundial, desde el punto de vista legal y económico, de estas firmas digitales debe existir la interoperabilidad de cada una de estas TTP con los demás y con las aplicaciones comerciales.

El objetivo de esta Recomendación | Norma Internacional es definir los servicios requeridos para soportar la aplicación de firmas digitales para el no repudio de creación. Puesto que el uso de mecanismos de firmas digitales para el no repudio de creación de un documento implica la integridad del documento y la autenticidad del creador, se pueden también combinar los servicios descritos en esta Recomendación | Norma Internacional para implementar los servicios de integridad y autenticidad. Esto se lleva a cabo de modo que se promueva la interoperabilidad entre las TTP y las aplicaciones comerciales

NOTA – No existe un motivo intrínseco para que cada TTP planificada para que soporte la aplicación de firmas digitales deba ofrecer todos estos servicios. Es posible que algunas TTP que ofrecen diferentes servicios cooperen en el soporte de la utilización de firmas digitales. Pero, desde el punto de vista de las aplicaciones comerciales potenciales, puede requerirse la gama completa de los servicios, haciéndose la interoperabilidad incluso más importante en este escenario. Ello constituye una justificación adicional de la recopilación conjunta de todos estos servicios en un documento.

NORMA INTERNACIONAL

RECOMENDACIÓN UIT-T

TECNOLOGÍA DE LA INFORMACIÓN – TÉCNICAS DE SEGURIDAD – ESPECIFICACIÓN DE SERVICIOS DE TERCERA PARTE CONFIABLE PARA SOPORTAR LA APLICACIÓN DE FIRMAS DIGITALES

1 Alcance

En esta Recomendación | Norma Internacional se definen los servicios de tercera parte confiable (TTP) necesarios para soportar la aplicación de firmas digitales para el no repudio de la creación de documentos.

En esta Recomendación | Norma Internacional se definen además las interfaces y protocolos que posibilitan la interoperabilidad entre entidades asociadas con esos servicios TTP.

Se necesitan definiciones de protocolos y servicios técnicos que permitan la implementación de servicios TTP y las aplicaciones comerciales correspondientes.

Esta Recomendación | Norma Internacional se refiere a:

- la implementación e interoperabilidad;
- las especificaciones de servicios; y
- los requisitos técnicos.

Esta Recomendación | Norma Internacional no describe la gestión de las TTP ni se refiere a temas tales como los de organización, funcionamiento o personal. Esas materias se tratan principalmente en UIT-T X.842 | ISO/CEI TR 14516, *Information technology – Security techniques – Guidelines on the use and management of Trusted Third Party services*.

NOTA 1 – Puesto que la interoperabilidad es el tema principal de la presente Recomendación | Norma Internacional, se mantienen las restricciones siguientes:

i) En esta Recomendación | Norma Internacional sólo se tienen en cuenta aquellos servicios que pueden ser ofrecidos por una TTP a entidades terminales o a otra TTP.

ii) Sólo se tienen en cuenta aquellos servicios que pueden ser solicitados y/o entregados mediante mensajes digitales normalizables.

iii) Sólo se especifican en detalle aquellos servicios para los que se puedan acordar mensajes normalizados ampliamente aceptados en el momento en que se publique esta Recomendación | Norma Internacional.

En otros documentos se especificarán otros servicios, cuando se disponga para ellos de mensajes normalizados ampliamente aceptados. Los servicios de indicación de tiempo, en particular, se definirán en un documento aparte.

NOTA 2 – Las estructuras de datos y mensajes de esta Recomendación | Norma Internacional se especificarán de acuerdo con los documentos RFC 2510 y RFC 2511 (para los servicios de gestión de certificados) y RFC 2560 (para servicios OCSP). El formato de petición de certificado permite además la interoperabilidad con PKCS#10. Véanse en el anexo C referencias a los documentos mencionados en esta nota.

NOTA 3 – Se están llevando a cabo otros trabajos de normalización de servicios TTP en entornos y aplicaciones específicos, como SET o EDIFCAT. Esos servicios quedan fuera del alcance de la presente Recomendación | Norma Internacional.

NOTA 4 – Esta Recomendación | Norma Internacional define especificaciones técnicas de servicios. Las especificaciones son independientes de las políticas, las normas legales específicas y los modelos organizativos (que podrían definir, por ejemplo, cómo se comparten obligaciones y responsabilidades entre las autoridades de certificación y las autoridades de registro). Naturalmente, la normativa de las TTP que ofrecen los servicios descritos en esta Recomendación | Norma Internacional habrá de especificar la manera según la cual las TTP cumplirán las normas legales y los demás aspectos mencionados más arriba. Dicha normativa ha de especificar, sobre todo, cómo se determina la validez de los certificados y las firmas digitales.

2 Referencias normativas

Los siguientes documentos normativos contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación | Norma Internacional. Para referencias fechadas, no se aplican las subsiguientes modificaciones a cualquiera de estas publicaciones o revisiones de las mismas. Sin embargo, se preconiza que los participantes en acuerdos basados en la presente Recomendación | Norma Internacional investiguen la posibilidad de aplicar las ediciones más recientes de los documentos normativos citados a continuación. Para las referencias no fechadas, se aplica la última edición del documento normativo a que se hace referencia. Los miembros de la CEI y de la ISO mantienen registros de las Normas Internacionales actualmente vigentes. La Oficina de Normalización de las Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones UIT-T actualmente vigente.

2.1 Recomendaciones | Normas Internacionales idénticas

- Recomendación UIT-T X.501 (1997) | ISO/CEI 9594-2:1998, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Modelos.*
- Recomendación UIT-T X.509 (2000) | ISO/CEI 9594-8:2001, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Marcos para certificados de clave pública y de atributos.*
- Recomendación UIT-T X.520 (1997) | ISO/CEI 9594-6:1998, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Tipos de atributos seleccionados.*
- Recomendación UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*
- Recomendación UIT-T X.681 (1997) | ISO/CEI 8824-2:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de objetos de información.*
- Recomendación UIT-T X.682 (1997) | ISO/CEI 8824-3:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de constricciones.*
- Recomendación UIT-T X.683 (1997) | ISO/CEI 8824-4:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Parametrización de especificaciones de la notación de sintaxis abstracta uno.*
- Recomendación UIT-T X.690 (1997) | ISO/CEI 8825-1:1998, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de las reglas de codificación básica, de las reglas de codificación canónica y de las reglas de codificación distinguida.*
- Recomendación UIT-T X.810 (1995) | ISO/CEI 10181-1:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad para sistemas abiertos: Visión general.*
- Recomendación UIT-T X.813 (1996) | ISO/CEI 10181-4:1997, *Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad en sistemas abiertos: Marco de no rechazo.*

2.2 Otras referencias

- ISO/CEI 9796-2:1997, *Information technology – Security techniques – Digital signature schemes giving message recovery – Part 2: Mechanisms using a hash-function*.
- ISO/CEI 9796-3:2000, *Information technology – Security techniques – Digital signature schemes giving message recovery – Part 3: Discrete logarithm based mechanisms*.
- ISO/CEI 10118-1:1994, *Information technology – Security techniques – Hash-functions – Part 1: General*.
- ISO/CEI 10118-2:1994, *Information technology – Security techniques – Hash-functions – Part 2: Hash-functions using an n-bit block cipher algorithm*.
- ISO/CEI 10118-3:1998, *Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions*.
- ISO/CEI 11770-1:1996, *Information technology – Security techniques – Key management – Part 1: Framework*.
- ISO/CEI 11770-2:1996, *Information technology – Security techniques – Key management – Part 2: Mechanisms using symmetric techniques*.

- ISO/CEI 11770-3:1999, *Information technology – Security techniques – Key management – Part 3: Mechanisms using asymmetric techniques*.
- ISO/CEI 13888-1:1997, *Information technology – Security techniques – Non-repudiation – Part 1: General*.
- ISO/CEI 13888-2:1998, *Information technology – Security techniques – Non-repudiation – Part 2: Mechanisms using symmetric techniques*.
- ISO/CEI 13888-3:1997, *Information technology – Security techniques – Non-repudiation – Part 3: Mechanisms using asymmetric techniques*.
- ISO/CEI 14888-1:1998, *Information technology – Security techniques – Digital signatures with appendix – Part 1: General*.
- ISO/CEI 14888-2:1999, *Information technology – Security techniques – Digital signatures with appendix – Part 2: Identity-based mechanisms*.
- ISO/CEI 14888-3:1998, *Information technology – Security techniques – Digital signatures with appendix – Part 3: Certificate-based mechanisms*.
- ISO/CEI 15946-2 (por publicar), *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 2: Digital signatures*.

3 Definiciones

Se aplica el término siguiente definido en ISO/CEI 11770-1:

gestión de claves

Se aplica el término siguiente definido en ISO/CEI 10181-1:

tercera parte confiable (TTP, *trusted third party*)

Se aplican los términos siguientes definidos en la Rec. UIT-T X.509 | ISO/CEI 9594-8:

certificado CA
autoridad de certificación (CA, *certification authority*)
certificado de claves públicas

NOTA – En esta Recomendación | Norma Internacional se utiliza también el término más breve "certificado" para indicar "certificado de claves públicas".

política de certificado
lista de revocación de certificados (CRL, *certificate revocation list*)
trayecto de certificación (*certification path*)

Se aplican los términos siguientes definidos en ISO/CEI 10118-1:

función de troceo
código de troceo (valor de troceo)

Se aplica el término siguiente definido en ISO/CEI 14888-1:

parámetro de dominio (*domain parameter*)

A efectos de esta Recomendación | Norma Internacional, se aplican las siguientes definiciones.

3.1 servicios de gestión de certificados: Todos los servicios necesarios para el mantenimiento del ciclo de vida de los certificados, incluidos el registro, la certificación, la distribución y la revocación de certificados.

3.2 servicio de certificación: Servicio de creación y asignación de certificados realizado por una CA y descrito en ISO/CEI 9594-8: 1995.

3.3 firma digital: Transformación criptográfica de una unidad de datos que permite a un receptor probar el origen e integridad de la unidad de datos y proteger al emisor y al receptor de la unidad de datos contra la falsificación por terceros, y al emisor contra la falsificación por el receptor.

NOTA – Las firmas digitales pueden ser utilizadas por entidades terminales (véase a continuación) para los fines de autenticación, integridad de los datos y no repudio de la creación de datos. El uso del no repudio de la creación de datos es el más importante para la vinculación legal de firmas digitales. La definición anterior se ha tomado de ISO/CEI 9798-1.

3.4 autoridad de certificación de confianza directa: Una CA de confianza directa es una CA cuya clave pública ha sido obtenida y está siendo almacenada por una entidad terminal de un modo seguro, de confianza, y cuya clave pública es aceptada por esa entidad terminal en el contexto de una o varias aplicaciones.

3.5 clave de autoridad de certificación de confianza directa: Una clave de CA de confianza directa es una clave pública de una CA de confianza directa. Ha sido obtenida y está siendo almacenada por una entidad terminal de un modo seguro, de confianza. Se utiliza para verificar los certificados sin que sea ella misma verificada por medio de un certificado creado por otra CA.

NOTA – Si, por ejemplo, las CA de varias organizaciones se certifican mutuamente unas a otras de manera recíproca (cruzada) (véase el anexo A), la CA de confianza directa de una entidad puede ser la CA de la organización de la entidad. Las CA de confianza directa y las claves de CA de confianza directa pueden variar de una entidad a otra. Una entidad puede contemplar varias CA como CA de confianza directa.

3.6 servicio de directorio: Servicio de búsqueda y recuperación de información de un catálogo de objetos perfectamente definidos, que puede contener información relativa a certificados, números de teléfono, direcciones, etc. Un ejemplo es el servicio de directorio conforme con la Rec. UIT-T X.500 | ISO/CEI 9594-1.

3.7 servicio de distribución de claves: Servicio de distribución de claves de forma segura a entidades autorizadas realizado por el Centro de distribución de claves y descrito en ISO/CEI 11770-1.

3.8 no repudio de la creación: Protección contra la negación falsa de una entidad de haber creado el contenido de un mensaje (es decir, de ser responsable del contenido de un mensaje)

3.9 entorno de seguridad personal (PSE, *personal security environment*): Almacenamiento local seguro de una clave privada de entidad, la clave de CA de confianza directa y posiblemente de otros datos. Dependiendo de la política de seguridad de la entidad o de los requisitos del sistema, puede tratarse, por ejemplo, de un fichero protegido criptográficamente o de un testigo de soporte físico resistente a la manipulación.

3.10 servicio de personalización: Servicio de almacenamiento de información criptográfica (en especial, claves privadas) en un PSE.

NOTA – Las medidas de seguridad y de organización de este servicio no caen dentro del alcance de esta Recomendación | Norma Internacional. Véanse las directrices para el uso y la gestión de la tercera parte confiable (TTP) (*Guidelines on the use and management of trusted third party*) en la Rec. UIT-T X.842 | Informe técnico de ISO/CEI 14516.

3.11 directorio de claves pública (PKD, *public key directory*): Directorio que contiene un conjunto, o subconjunto, perfectamente definido de certificados de claves públicas. Este directorio puede contener certificados procedentes de diferentes autoridades de certificación.

3.12 infraestructura de claves públicas (PKI, *public key infrastructure*): Sistema formado por tercera parte confiable (TTP), junto con los servicios que hacen posible el soporte de la aplicación de firmas digitales (incluidas la generación y validación), y por las personas o componentes técnicos que utilizan estos servicios.

NOTA – Algunas veces, las personas y los componentes técnicos que forman parte de una PKI utilizando los servicios de TTP, pero que no son ellos mismos TTP, son designados como entidades terminales. Una tarjeta inteligente que puede utilizarse como dispositivo de almacenamiento y procesamiento constituye un ejemplo de equipo técnico utilizado por una entidad terminal.

3.13 autoridad de registro (RA, *registration authority*): Autoridad titulada y fiduciaria (de confianza) para prestar el servicio descrito después.

3.14 servicio de registro: Servicio de identificación de entidades y de registro de las mismas en una lista que permita la asignación segura de certificados a estas entidades.

3.15 servicio de indicación de tiempo: Servicio que atestigua la existencia de datos electrónicos en un instante de tiempo preciso.

NOTA – Los servicios de indicación de tiempo son útiles y probablemente indispensables para soportar la validación a largo plazo de las firmas. Será definidos en un documento separado.

4 Abreviaturas

A los efectos de esta Recomendación | Norma Internacional se aplican las siguientes siglas.

CA Autoridad de certificación (c*ertification authority*)

CRL Lista de revocación de certificados (c*ertificate revocation list*)

EE Entidad terminal (*end entity*)

OSCP Protocolo en línea del estado del certificado (*on-line certificate status protocol*)

PKD Directorio de claves públicas (*public key directory*)

PKI Infraestructura de claves públicas (*public key infrastructure*)

PSE Entorno de seguridad personal (*personal security environment*)

RA Autoridad de registro (*registration authority*)

TTP Tercera parte confiable (*trusted third party*)

5 Clasificación descriptiva de los servicios

En esta cláusula se describen los servicios que pueden utilizarse dentro del contexto de los servicios TTP para firmas digitales. Aquí se da una descripción de alto nivel de tales servicios que es independiente de los formatos de datos, algoritmos, lenguajes de descripción, etc.

En las cláusulas siguientes se da una especificación detallada de algunos de estos servicios.

No es necesario que cada planificación de TTP para soportar la aplicación de firmas digitales ofrezca todos estos servicios. Puede ocurrir que varios servicios TTP que ofrecen servicios diferentes cooperen en el soporte de la utilización de firmas digitales.

NOTA 1 – Puesto que la interoperabilidad es el objetivo principal de esta Recomendación | Norma Internacional, aquí solo se describen aquellos servicios que son ofrecidos por una TTP a entidades terminales o a otra TTP. Además, solamente se cubren aquellos servicios que pueden ser solicitados y/o prestados mediante mensajes digitales normalizables. (Sin embargo, esto no implica que los mensajes normalizados sean de hecho definidos para todos los servicios mencionados en esta Recomendación | Norma Internacional).

En los ejemplos siguientes se muestran servicios que **no** están cubiertos:

1) Registro cronológico de los eventos de seguridad importantes. Con respecto a una PKI de firma digital, este servicio es un servicio interno de las TTP pero no es ofrecido a las entidades.

2) Servicios criptográficos generales (por ejemplo, servicio de criptación). Los procesos como la criptación forman parte de algunos servicios pero no constituyen un servicio autónomo en el contexto de firmas digitales.

3) Archivo y recuperación de claves Éste puede ser un servicio interno de claves de CA de confianza directa. En general, no será realizado para claves de firmas digitales de entidades terminales.

NOTA 2 – Los servicios de indicación de tiempo se definirán en un documento separado.

5.1 Servicios de gestión de certificado

Esta subcláusula contiene la descripción de los siguientes servicios que forman parte del ciclo de vida de los certificados:

- registro;
- certificación de claves públicas;
- revocación de certificado;
- actualización de certificados; y
- actualización de claves.

En la cláusula 7 se especifica con detalle el flujo de mensajes en línea de estos servicios (salvo la determinación del estado del certificado) y en la cláusula 8 se da la especificación ASN.1 de las estructuras de datos necesarias para estos mensajes. En la cláusula 9 se recogen las especificaciones análogas para la determinación en línea del estado del certificado (véase 5.1.3.3, método segundo).

Los protocolos de acceso al directorio utilizados para confeccionar certificados y listas CRL disponibles públicamente no se especifican aquí, puesto que las especificaciones de estos protocolos ya están recogidas en la Rec. UIT-T X.511 | ISO/CEI 9594-3 y Rec. UIT-T X.519 | ISO/CEI 9594-5.

NOTA – Otros protocolos de acceso al directorio son el LDAP (RFC 1777, 2555 y 2587-LDAPv2) o el acceso WEB (RFC 2585).

En la figura 1 se presenta una visión general de la arquitectura de una PKI con algunos ejemplos de servicios.

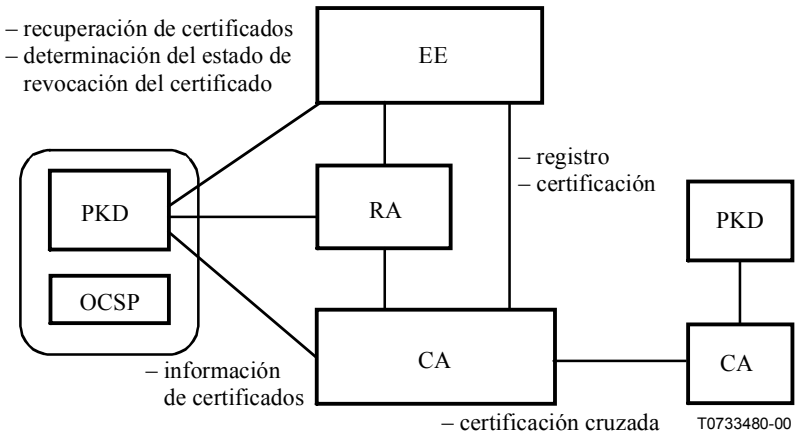

Figura 1 – Visión general de los servicios de gestión de certificados

5.1.1 Registro

La seguridad de una infraestructura de claves públicas depende de la identificación y registro adecuados de las entidades.

Para todas las entidades terminales, una RA (que puede ser también la CA) ha de verificar la identidad de la entidad terminal del modo adecuado, y ha de asignar inequívocamente un nombre exclusivo a cada entidad terminal dentro de su propio dominio. La política de certificados determina la clase de medios que han de utilizarse para la identificación (por ejemplo, documentos ID) y si la entidad ha de estar presente personalmente. Pueden ser también necesarios los alias de los nombres de las entidades terminales. Las entidades terminales que son componentes técnicos se registran de acuerdo con la política de seguridad de la aplicación, por ejemplo, la gestión de red. En aplicaciones en las que es importante la distinción entre entidades terminales humanas o no humanas, debe recogerse esta distinción clara en el certificado. Por ejemplo, un "nombre" puede ser "dispositivo de tipo X", número "Y" ubicado en "Z", pudiéndose efectuar la distinción de acuerdo con la política o indicar la diferencia mediante una extensión.

Puede utilizarse un formulario de registro para recopilar los datos pertinentes, por ejemplo, nombre, unidad empresarial, dirección empresarial y dirección de entrega del material en clave.

EJEMPLO: Un escenario posible dentro de una compañía es aquel en el que cada empleado deba estar presente personalmente en la oficina de personal pertinente, mostrando su tarjeta de identidad válida. El funcionario encargado del servicio de personal da fe de su identidad y envía un formulario de registro firmado a la CA.

5.1.2 Certificación de claves públicas

La CA es responsable del proceso de certificación y de vinculación del nombre (o un seudónimo) de la entidad con claves públicas. En la Rec. UIT-T X.509 | ISO/CEI 9594-8 se describe un formato de certificado.

Esta Recomendación | Norma Internacional requiere que la prueba de posesión de la clave privada que corresponde a la clave publica incluida en el certificado se lleve a cabo en el curso del servicio de certificación. Dependiendo de la política adoptada, una CA puede garantizar otras propiedades de la clave pública de la entidad terminal, por ejemplo, mediante la inclusión de uno de los dos servicios descritos en 5.3.2 y 5.3.3, o ambos.

5.1.3 Revocación de certificados

5.1.3.1 Consideraciones generales

Un tema importante en relación con los certificados de claves públicas es que, por diversas circunstancias, puede ser necesario *revocarlos* antes de la expiración del plazo programado. La revocación puede ser efectuada por la CA emisora o por otra autoridad, según la política adoptada.

La revocación es obligatoria si está comprometida una clave privada. Otros motivos de revocación pueden ser el cambio de nombre, la terminación del empleo, etc.

La política de certificados debe establecer todos los motivos de revocación. Algunos de estos motivos pueden encontrarse en la Rec. UIT-T X.509 | ISO/CEI 9594-8. La política debe especificar quién puede solicitar revocaciones, así como si puede utilizarse un testigo específico para identificar las entidades que están autorizadas a revocar certificados, tal como las contraseñas de revocación de una vez.

5.1.3.2 Métodos de revocación

Para revocar certificados pueden utilizarse dos métodos diferentes:

1) Emisión de una lista de revocación de certificados (CRL).

   Se emite periódicamente una CRL que está firmada por una CA y que identifica todos los certificados que son revocados.

   La revocación conduce a una inserción inmediata en la CRL, que incluye también la hora de la revocación del certificado correspondiente. Adicionalmente, debe eliminarse el certificado del PKD.

   NOTA – Dependiendo de la política, quizás convenga mantener el certificado para comprobar la validez de las firmas efectuadas antes de la fecha de la revocación (por ejemplo, si el motivo de la revocación el compromiso de una clave).

   Además de la publicación periódica, la CRL actualizada puede publicarse inmediatamente.

   ISO/CEI 11770-1 identifica dos indicaciones de tiempo de revocación diferentes:

   – el tiempo de compromiso conocido o sospechado, y

   – el tiempo en el cual la CA fue avisada del compromiso por la entidad.

   Dependiendo de la política, la inserción del certificado en la CRL puede ser suprimida cuando el certificado caduca (compárese con 5.1.3.3 Determinación del estado de revocación del certificado).

2) Almacenamiento del estado del certificado en una base de datos interna de confianza de la CA y ofrecimiento en línea a las entidades de la información de estado del certificado (compárese con 5.1.3.3 Determinación del estado de revocación de un certificado).

Ambos métodos se pueden combinar.

Si una clave está comprometida, el procedimiento normalizado consiste en revocar el certificado correspondiente, reinicializar la entidad terminal, generar un nuevo par de claves pública/privada y certificar la nueva clave pública con los atributos anteriores.

Dependiendo de la política, un certificado puede:

– ser revocado permanentemente.

– ser suspendido. La inserción CRL incluye una bandera que indica el estado "en retención".

La política de la CA deberá especificar lo que significa el estado "en retención" con respecto al nivel de confianza con que representa a una entidad y sobre todo el modo en que una entidad debe tratar esta situación. Por ejemplo, un certificado puede ser suspendido como resultado de una petición de revocación no autorizada. Algunas políticas no permiten en absoluto el estado "en retención".

EJEMPLO: Una política puede establecer lo siguiente: Cuando se realiza una autenticación, deberá ser rechazada mientras el certificado está suspendido. Cuando una evidencia es verificable utilizando un certificado suspendido, el resultado de la verificación será negativo si se necesita inmediatamente un resultado pero puede también ser interpretado como de una validez condicional. Cuando la suspensión es seguida por una revocación, la evidencia se hace no válida y la fecha de revocación será la fecha del inicio de la suspensión (es decir, no la fecha del final de la suspensión).

5.1.3.3 Determinación del estado de revocación de un certificado

Este servicio permite a una entidad determinar si un certificado está revocado. Esto puede efectuarse por diferentes métodos correspondientes a los métodos de revocación descritos en 5.1.3.2:

1) Método 1: Comprobación del PKD y de la CRL.

2) Método 2: Solicitar en línea el estado de una TTP que goza de confianza para este fin. La respuesta de la TTP deberá transportarse de manera auténtica a la entidad.

5.1.3.4 Revocación de un certificado de CA

Si está comprometida una clave privada de CA, aparece un escenario especial. Ocurre lo siguiente:

– El certificado de la clave pública correspondiente a la clave comprometida de CA debe ser revocado.

  La firma que figura en los certificados calculados con la clave comprometida de la CA deja de ser garantía de fiabilidad. Sin embargo, es posible garantizar la validez de esos certificados con otros medios (por ejemplo, si los certificados son almacenados de manera fiable por una TTP que presta un servicio OCSP). En cualquier caso, las entidades deben obtener un certificado nuevo y una nueva clave de CA de confianza directa.

En determinadas situaciones, que dependen de la política de certificados y de la arquitectura del sistema, deben generarse nuevos pares de claves para las entidades.

Es importante señalar que, bajo este escenario, las firmas que fueron emitidas antes de que tuviera lugar el compromiso, y que son aseguradas por medios adicionales (por ejemplo, que son dotadas de una indicación de tiempo por una autoridad de indicación de tiempo cuyo certificado es todavía válido) puede aún ser consideradas válidas dependiendo de la política, mientras que las firmas emitidas después del momento en que ocurrió determinado compromiso, o que no han sido aseguradas por medios adicionales, serán consideradas no válidas.

– Si la clave pública correspondiente a la clave privada comprometida de la CA es certificada de modo cruzado con otras CA, se enviará un mensaje de alerta a las otras CA. Este mensaje de alerta les notificará la revocación del certificado de CAs de certificación cruzada.

5.1.4 Actualización de un certificado

Cuando caduca, un certificado puede ser actualizado mediante la expedición de un nuevo certificado para la clave pública de la entidad que ya estaba contenida en el certificado antiguo.

Este método deberá utilizarse en caso de que:

– el par de claves de la entidad está comprometido; o

– el estado de la criptografía indica que el algoritmo de claves públicas junto con los parámetros del par de claves puede garantizar la seguridad de la firmas generadas con este par de claves durante el periodo de validez del nuevo certificado; o

– el nuevo certificado tendrá diferencias substanciales con el certificado antiguo en cuanto a política, extensiones o atributos.

La política de certificados de la CA puede especificar que para la expedición de un nuevo certificado resulta aceptable un procedimiento de registro simplificado.

EJEMPLO: Si el certificado antiguo no ha sido revocado, este hecho puede aceptarse como suficiente para expedir un nuevo certificado.

El cambio de atributos no críticos en un certificado durante su periodo de validez, tal como el nombre de afiliación (debido, por ejemplo, a un cambio a otro departamento), puede conducir también a la necesidad de una nueva certificación de los atributos modificados con las mismas claves que antes. No obstante, en este caso deberá revocarse el certificado anterior.

5.1.5 Actualización de claves

La propia entidad o la TTP generan un par de claves y se expide un certificado de la clave pública de este nuevo par.

Este método deberá adoptarse en caso de expiración de un certificado si no es aceptable la actualización del certificado por uno de los motivos dados en 5.1.4. Dependiendo de la política de la CA, el método puede utilizarse también en otras situaciones.

La política de certificados de la CA puede especificar que en la expedición de un certificado de una clave actualizada se acepte un procedimiento de registro simplificado.

5.2 Servicios de gestión de claves

En ISO/CEI 11770 (todas las partes) se recogen descripciones generales de los servicios de gestión de claves.

Esta subcláusula contiene solamente una descripción de los servicios de gestión de claves que pueden ofrecerse como parte de los servicios relacionados con el ciclo de vida de los certificados (compárese con 5.1). La especificación detallada del flujo de mensajes en línea de los servicios de la cláusula 7, y la especificación ASN.1 de las estructuras de datos de la cláusula 8 abarcan los servicios de gestión de claves en la medida en que ellos generan mensajes en línea entre la CA, RA y/o EE.

5.2.1 Generación de claves

En el contexto de las firmas digitales, las TTP pueden generar pares de claves pública/privada para entidades terminales, si esta operación no es efectuada por las propias entidades. Aunque el servicio puede ser ofrecido por TTP independientes, se supone más adelante que es realizada por una CA o una RA en respuesta a una petición de certificación, o por una entidad terminal antes de una petición de certificación.

5.2.2 Distribución de claves

5.2.2.1 Distribución de claves privadas

En la descripción del servicio "distribución de claves", se pueden distinguir diferentes modos de transmisión de las claves (en línea o fuera de línea) y diferentes componentes (TTP o entidad) generadores de claves. En el caso de generación de claves centralizada, la TTP es responsable de la transmisión segura del certificado de la clave privada y la clave pública de la entidad. Además debe garantizarse que se enviará una clave privada de entidad de manera confidencial. Esto puede realizarse mediante la criptación de esta clave con una clave de transporte especial (simétrica) que solo es conocida por la TTP y la entidad correspondiente. Alternativamente, la clave privada puede transmitirse utilizando facilidades de soporte físico seguras y apropiadas como, por ejemplo, las tarjetas inteligentes. La transmisión de la clave privada no es necesaria si la entidad es capaz de generar su propio par de claves asimétricas. En este caso la TTP solamente tiene que realizar algunas pruebas de plausibilidad (por ejemplo, comprobar si la entidad es capaz de firmar un mensaje con una clave privada correspondiente a la clave pública), certificar la clave pública de la entidad y poner este certificado a disposición.

5.2.2.2 Distribución de claves públicas

Las claves públicas deben ponerse a disposición de las entidades de un modo que garantice su autenticidad. En el caso de claves públicas certificadas, la distribución de claves se realiza mediante la distribución del certificado, y la autenticidad es garantizada por la firma de la CA que ha creado el certificado.

En el caso de una clave de CA de confianza directa se utilizarán otros medios de distribución segura. Si las claves privadas de una entidad se distribuyen a otra entidad mediante un testigo de soporte físico seguro, este testigo puede también utilizarse para entregar la clave de CA. En otros casos se requiere un proceso adicional. En ISO/CEI 11770-3, subcláusula 8.1 "Public key distribution without a trusted third party" (distribución de claves públicas sin un tercero de confianza) pueden encontrarse métodos para llevar acabo esta operación.

5.2.3 Personalización

El almacenamiento de claves privadas y de datos adicionales se puede facilitar por medio de un testigo físico. En esta situación, la CA, RA o entidades terminales deben soportar la personalización de un testigo. Por ejemplo, la personalización de las tarjetas inteligentes puede incluir el establecimiento de procedimientos (como la creación del sistema de ficheros), la selección de un PIN (número de identificación personal) aleatorio o contraseña, y el envío y almacenamiento de todos los datos pertinentes dentro de la tarjeta inteligente.

5.3 Otros servicios

5.3.1 Certificación cruzada

La certificación cruzada es un servicio ofrecido para la verificación de firmas de entidades terminales con certificados procedentes de una CA por entidades terminales con certificados procedentes de otra CA. Por ejemplo, una CA1 expide un certificado para una CA2 en otra PKI con el efecto de que las entidades que dan su confianza a la CA1 pueden verificar certificados de entidades en la otra PKI a través de un trayecto de certificación que incluye este nuevo certificado.

En la cláusula 7 se da una especificación detallada del flujo de mensajes en línea de este servicio y en la cláusula 8 la especificación ASN.1 de las estructuras de datos necesarias para estos mensajes.

5.3.2 Validación de parámetros de dominio

La validación de parámetros de dominio es la validación de un conjunto propuesto de parámetros de dominio para garantizar que cada parámetro del conjunto cumple con todos los atributos que se le reclaman.

EJEMPLOS:

a) puede requerirse que un parámetro válido sea un número primo: para efectuar esta validación se ejecuta una prueba (quizás probabilística) de la condición de número primo para asegurarse de que el número reclamado es realmente primo;

b) puede requerirse que un parámetro válido guarde una relación aritmética con algunos otros parámetros: para validar esta situación, se prueba la relación aritmética con el objetivo de garantizar que se cumple;

c) puede comprobarse un caso débil específico (por ejemplo, en una lista exclusiva) para asegurar que no se aplica al conjunto en cuestión; o

d) puede requerirse que se genere un parámetro mediante el uso de una semilla en una función de troceo sembrada canónica: para validar esto, se introduce la semilla en la función de troceo sembrada canónica a fin de garantizar que genera realmente el parámetro.

El generador de un conjunto de parámetros de dominio debe garantizar que ellos superan la validación de parámetros de dominio. Si cualquier otra entidad necesita efectuar la validación de los parámetros de dominio, ello dependerá de la relación de confianza entre el generador y la entidad. Si se utiliza un conjunto de parámetros de dominio no válidos se pueden producir resultados imprevistos, incluida la perdida de la seguridad perseguida. Como los parámetros de dominio son normalmente públicos, es preferible que la validación se pueda efectuar de manera autónoma (es decir, sin necesidad de que el generador de los parámetros de dominio responda a preguntas), lo que constituye el caso más frecuente.

Usualmente una CA generará y validará un conjunto de parámetros de dominio que pueden ser entonces implícitamente aceptados como fiables por todos los miembros de la PKI.

5.3.3 Validación de claves públicas

La validación de claves públicas es la validación de una clave pública para garantizar que está conforme con los requisitos aritméticos de tal clave, es decir que la clave pública pretendida es plausible. La validación de claves públicas supone que los parámetros de dominio han sido validados previamente.

EJEMPLOS:

a) puede ser necesario que un parámetro válido se encuentre dentro de en una gama específica de valores: para validar esto se prueba el parámetro requerido para garantizar que se encuentra dentro de la gama correcta;

b) puede ser necesario que un parámetro válido tenga un orden específico (en general un orden primo grande): para validar esta situación, se prueba el parámetro requerido para garantizar que tiene el orden correcto; o

c) puede ser necesario que un parámetro válido guarde una relación aritmética específica con algunos otros parámetros, y para validar esta situación se prueba la relación aritmética.

Si se utiliza una clave pública no válida se pueden producir resultados imprevistos, incluida la perdida de la seguridad perseguida del propietario de la clave privada asociada, de los receptores de documentos firmados con aquella clave o de ambos. Una tercera parte confiable, tal como una CA, puede efectuar la validación de claves públicas para asegurar todas las entidades de su dominio.

5.3.4 Validación de certificados

Si una entidad terminal que desea confiar en una firma digital de otra entidad terminal no es capaz de verificar el correspondiente certificado puede pedir a una TTP que lo haga.

La validación de certificados se refiere a la validez de un solo certificado. Un solo certificado puede ser suministrado en la petición, o este certificado único puede ser suministrado y seguido por una secuencia de certificados (que no forman necesariamente un trayecto de certificación).

Al probar la validación de una certificado se deben cumplir las dos condiciones siguientes:

a) Para una política de certificados, solamente puede ser válido un certificado. Es por tanto necesario conocer que política de seguridad se aplica. La política de certificados define, entre otras cosas, la regla de construcción de un trayecto de certificación válido. Así pues, la tarea es determinar si es posible construir un trayecto de certificación válido. Si los certificados suministrados no son suficientes, el servicio puede intentar reunir por sí mismo los certificados perdidos. Puede haber diferentes modos de identificar la

política de seguridad, pero deberá haber un puntero que apunte a la política correcta. A este fin, se puede utilizar un OID (identificador de objeto) o un URL. Con el fin de garantizar que el puntero sea correcto, debe añadirse un valor de código de troceo de la política. Una forma simple y degenerada de una política es el certificado único autofirmado. En este caso, es necesario apuntar a este certificado autofirmado proporcionando el nombre de la CA y el número de serie del certificado.

b) También es importante saber en qué fecha debe realizarse la prueba de validez. Esto tiene una importancia especial para garantizar que el certificado no ha sido revocado antes de dicha fecha. Dependiendo de la política de certificados, habrá de comprobarse la validez no sólo del propio certificado sino la de todos los certificados del trayecto de certificación en momentos que pueden diferir de unos a otros y depender de la hora de la firma. El conocimiento de la fecha de la firma es por tanto de particular importancia.

Se concluye que los parámetros de entrada de este servicio serían:

- un identificador del certificado cuya validez ha de comprobarse, seguido de cero o más certificados;
- cero, uno o más CRL (listas de revocación de certificados);
- el identificador de una política de seguridad frente a la cual debe probarse el certificado, es decir, un puntero tal como un OID o un URL seguido de un troceo de la política de seguridad, o el identificador de un certificado autofirmado (en un caso degenerado);
- el instante en que deberá efectuarse la prueba.

Debe enumerarse el parámetro de salida.

5.3.5 Servicio de archivo

Puesto que se pueden utilizar firmas digitales para los contratos que deben gozar de validez durante un tiempo largo, deberá ser posible determinar la validez de una firma incluso mucho después de la expiración del certificado correspondiente. A este fin han adoptarse disposiciones que excedan los procedimientos normales de la CA, dado que el periodo de validez del certificado es el intervalo de tiempo durante el cual la CA normalmente garantiza que mantendrá la información relativa al estado del certificado. La política de una PKI conforme deberá definir estas disposiciones. Son disposiciones posibles el uso de servicios de indicación de tiempo o de servicios de archivo.

Los servicios de archivo pueden ser resultar importantes en la resolución de conflictos relacionados con las firmas digitales. Por ejemplo, los certificados, listas CRL y otros datos pueden ser archivados por TTP dedicadas, ya que los servicios usuales de directorio solamente pueden contener certificados y CRL que no hayan expirado. Para conseguir esto, puede registrarse el historial del ciclo de vida de los certificados y las CRL, de modo que pueda reconstruirse su marco temporal de validez. La política de certificados puede requerir que la propia CA archive todos los certificados y CRL que hayan sido expedidos hasta entonces.

6 Perfil mínimo de los certificados y las CRL

6.1 Perfil mínimo de los certificados

Los certificados deberán cumplir con el formato X.509 especificado en la Rec. UIT-T X.509 | ISO/CEI 9594-8. Las entidades que verifican los certificados han de ser capaces de procesar este formato.

Aquí no se dará un perfil completo de los certificados, pero se implementarán los siguientes campos de certificado por razones de seguridad y/o interoperabilidad (los nombres de los campos de certificado pueden verse en la Rec. UIT-T X.509| ISO/CEI 9594-8):

a) Se utilizará por lo menos la versión v3.

b) Las CA utilizarán las extensiones siguientes, y las aplicaciones deberán ser capaces de procesarlas:

- constricciones básicas;
- utilización de claves;
- identificador de clave de autoridad (excepción posible: certificados autofirmados para claves de CA de confianza directa);
- políticas de certificados.

c) Las aplicaciones deberán ser capaces de procesar la extensión de nombre alternativo del sujeto. Las CA utilizarán esta extensión si el campo del sujeto de un certificado esta vacío.

d) Las aplicaciones deberán ser capaces de procesar las constricciones de nombre, constricciones de política y extensiones de utilización de claves ampliadas. Las constricciones de nombre se utilizarán solamente en los certificados de CA.

6.2 Perfil mínimo de las CRL

Las listas CRL deberán cumplir con el formato X.509 que se especifica en la Rec. UIT-T X.509 | ISO/CEI 9594-8. Las entidades que utilizan listas CRL deberán poder procesar este formato.

Aquí no se va a dar un perfil completo de las CRL, pero deberán implementarse los campos de certificado siguientes por razones de seguridad e/o interoperabilidad (los nombres de los campos de certificado pueden verse en la Rec. UIT-T X.509 | ISO/CEI 9594-8):

a) Se utilizará por lo menos la versión 2, definida en la Rec. UIT-T X.509 | ISO/CEI 9594-8.

b) Las listas CRL deberán contener el campo de actualización siguiente, la extensión del número de la CRL y el campo *keyIdentifier* (identificador de clave) de la extensión identificadora de la clave de autoridad. Las aplicaciones deberán poder procesar estos campos.

7 Mensajes de gestión de certificados

En esta cláusula se definen los mensajes de todos los servicios relativos a la gestión de certificados. Las estructuras de datos deberán especificarse de conformidad con los documentos RFC 2510 y RFC 2511. Los mensajes se especifican detalladamente en la cláusula 8 utilizando la ASN.1. La sintaxis ASN.1 se define en las Recomendaciones UIT-T X.680-X.683 | ISO/CEI 8824 partes 1-4, y las reglas de codificación se describen en la Rec. UIT-T X.690 | ISO/CEI 8825-1.

NOTA 1 – Obsérvese que los protocolos en línea no son el único medio de implementación de estos servicios. Para todos los servicios hay métodos fuera de línea con los que se puede conseguir el mismo resultado, y esta Recomendación | Norma Internacional no obliga a la utilización de protocolos en línea. Por ejemplo, cuando se utilizan testigos de soporte físico, muchos de los servicios se pueden obtener como parte de la entrega del testigo físico.

NOTA 2 – El código ASN.1 es equivalente al de los documentos RFC arriba mencionados, si bien la sintaxis parece diferente en algunas partes.

Básicamente, cada tipo de intercambio de información TTP se compone de un mensaje de petición enviado hacia adelante por el iniciador y de un mensaje de respuesta enviado hacia atrás por el respondedor, tal como se muestra en la figura 2. En caso de problemas se puede sustituir la respuesta por un mensaje de error.

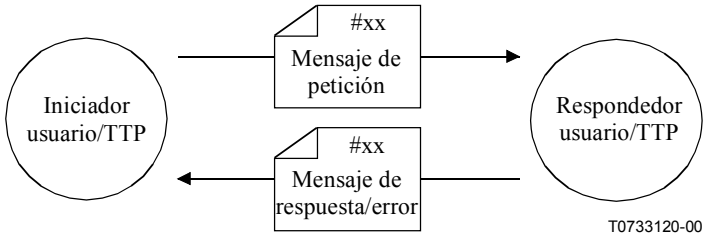

Figura 2 – Intercambio de información

En la figura 3 se muestra el flujo de información básico en un entorno TTP y todas las entidades involucradas.

Se pueden distinguir las siguientes clases de tipos de intercambio de información TTP:

- tipos de intercambio de información iniciado en una TTP con destino a otras TTP: clase "TT";
- tipos de intercambio de información iniciado en una TTP con destino a entidades terminales, y viceversa: clase "TU";
- tipos de intercambio de información iniciado en una entidad terminal con destino a otras entidades terminales: clase "UU".

Los tipos de intercambio de información de clase "UU" caen fuera del alcance de esta Recomendación | Norma Internacional.

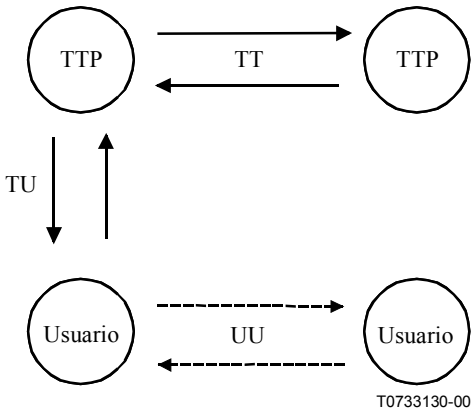

Figura 3 – Flujo de información

7.1 Visión general de los servicios y mensajes de la gestión de certificados

Hay que señalar que no todos los servicios de gestión PKI dan como resultado la creación de un mensaje PKI.

Los servicios que incluyen entidades y CAs descritas a continuación pueden incluir además una RA.

Los nombres de mensajes impresos en **negritas** se refieren a las definiciones de 8.1.2.

7.1.1 Inicialización

7.1.1.1 Inicialización de una CA

Antes de expedir ningún certificado una CA recientemente establecida (que planifica emitir las listas CRL) deberá producir listas de revocación iniciales, por ejemplo, versiones "vacías" de cada una de las CRL que han de producirse periódicamente.

Si una CA recientemente creada quiere servir como CA de confianza directa para algunas entidades terminales, deberá producir al menos un "autocertificado" que sea un certificado para las claves públicas de las CA firmado con la clave propia de la CA. Pueden necesitarse diferentes certificados autofirmados que incluyan diferentes constricciones de denominación para soportar diferentes aplicaciones. Con el fin de producir el autocertificado de CA útil para la entidades terminales que no adquieran el autocertificado por métodos "fuera de banda", la CA debe también producir un código de troceo para cada certificado autofirmado. Las entidades terminales que adquieran este código de troceo y un identificador del autocertificado correspondiente de modo seguro a través de algún método "fuera de banda", pueden entonces verificar este autocertificado, y en consecuencia los otros atributos allí contenidos.

Desde la perspectiva de los protocolos de gestión de PKI, la inicialización de una CA que no es una CA de confianza directa es igual que la inicialización de una entidad terminal. La única diferencia estriba en que la CA debe también producir una lista de revocación inicial.

7.1.1.2 Inicialización de CA a CA

Antes de que las entidades de una CA puedan utilizar las claves públicas de una entidad certificada por otra CA, debe establecerse una relación de confianza (fiducia) entre estas CA. Esto se puede conseguir mediante una certificación cruzada entres estas CA o a través de un trayecto de certificación fiable. Esta función puede ser soportada por mecanismos de intercambio distintos de los utilizados para las interacciones de las entidades con la CA (por ejemplo, intercambio físico entre gestores de CA), con lo cual no es preciso que sean soportados necesariamente por protocolos de intercambio en línea.

La certificación cruzada comprende el uso de tres mensajes:

- Mensaje de petición de certificación cruzada (**CrossCertReq**).
- Mensaje de respuesta de certificación cruzada (**CrossCertRep**).
- Mensaje de confirmación PKI (**PKIConfirm**).

La CA peticionaria es la CA que vendrá a ser el sujeto del certificado cruzado; la CA respondedora será el emisor del certificado cruzado.

7.1.1.3 Inicialización de una entidad

Es el proceso por el cual una entidad terminal se da a conocer a una CA o RA, antes de que la CA expida un certificado o certificados para dicha entidad terminal. El resultado final de este proceso (cuando tiene éxito) es que la CA expide un certificado de una clave pública de la entidad terminal, y devuelve dicho certificado a la entidad terminal y/o entrega el certificado en un depósito público. Este proceso puede comprender, y así sucede por regla general, múltiples "pasos", que pueden incluir una inicialización del equipo de la entidad terminal. Por ejemplo, el equipo de la entidad terminal debe ser inicializado de manera segura con la clave pública de una CA, para ser utilizado en la validación de los trayectos de certificado. Además, la entidad terminal necesita normalmente ser inicializada con su propio par (o pares) de claves propio.

La identidad de la entidad será verificada por la RA. Este procedimiento de generación puede comprender comunicaciones fuera de línea e incluso no electrónicas (por ejemplo, cuando se utiliza el servicio postal) y no necesita ser soportada por ningún protocolo de intercambio en línea. En 7.2.1 se describen varios esquemas de autenticación para el registro inicial de una entidad terminal.

Antes de utilizar los servicios ofrecidos por una TTP, una entidad puede tener necesidad de obtener información acerca de las funciones soportadas por la TTP junto con las claves necesarias para comunicar de modo seguro con el mismo y las claves públicas de la TTP precisas para verificar los certificados.

Puede utilizarse el diálogo GenMsg PKI para solicitar y suministrar esta información PKI. En este caso la petición será el mensaje **GenMsg**, la respuesta el mensaje **GenRep** y la respuesta de error el mensaje **ErrorMsg**. Estos mensajes se protegen por medio de un MAC basado en información secreta compartida (es decir, **PasswordBasedMAC**) o en cualquier otro medio autenticado (si la entidad terminal dispone de un certificado existente).

- Petición de información PKI (mensaje **GenMsg**).
- Respuesta de información PKI (mensaje **GenRep**).

Este par de mensajes está destinado a las peticiones generales de información PKI y puede utilizarse también después de la inicialización. Puede además ser utilizado por otras TTP para adquirir información relativa al estado actual de una CA. La CA deberá responder a la petición suministrando la información requerida, o enviando un error si no puede proporcionar una parte de la información.

Cuando se efectúa el proceso de registro/certificación, una entidad puede indicar que su petición es la primera petición de certificación utilizando los siguientes mensajes, en lugar de los mensajes **CertReq** y **CertRep** descritos más adelante:

- Petición de registro/certificación inicial (**InitReq**).
- Respuesta de registro/certificación inicial (**InitRep**).

7.1.2 Generación de claves

El par de claves pública/privada utilizado en un sistema asimétrico puede ser generado tanto por la entidad como por una TTP. El par de claves debe generarse de modo que pueda confiarse en que su generación sea la adecuada de conformidad con el algoritmo utilizado y que se mantenga la integridad de ambas claves y la confidencialidad de la clave privada.

a) Generación por la entidad

Si la entidad genera el par de claves pública/privada, la clave pública se pasa a la TTP como parte del proceso de registro/certificación. Se incluye en un campo de la plantilla de certificado contenida en el **CertReqMessage**.

b) Generación por la TTP

Si la TTP genera el par de claves pública/privada, ese par se transfiere a la entidad como parte del proceso de registro/certificación (véase 7.1.1.3). La clave privada se transfiere mediante entrega física PSE o dentro del campo clave privada de **CertKeyPair** dentro de **CertRep**. La clave pública se transfiere dentro de su certificado; el campo certificado de **CertKeyPair** dentro de **CertRep** está contenido en **CertRepContent**.

c) Actualización de claves

Cuando una pareja de claves debe caducar la entidad terminal correspondiente puede solicitar una actualización de claves, esto es, puede pedir que la TTP emita un nuevo certificado para una nueva pareja de claves. La petición se realiza mediante un mensaje de petición de actualización de claves. Si la entidad terminal ya posee una pareja de claves de firma (con un certificado de verificación correspondiente), este mensaje se protegerá normalmente por la firma digital de la entidad. La TTP devuelve el nuevo certificado (si la petición tiene éxito) en un mensaje de respuesta de actualización de claves.

- Petición de certificación de una actualización de claves (**KeyUpdReq**).
- Respuesta de certificación de una actualización de claves (**KeyUpdRep**).

La nueva pareja de claves puede ser generada por la entidad o por la TTP. En el segundo caso la clave es generada y cursada en estos mensajes como parte del proceso de registro/certificación tal como se ha descrito anteriormente.

NOTA – Cuando la entidad desea ampliar la vida de una clave certificada existente, deberá entonces actualizarse el certificado.

d) Notificación de actualización de claves de CA

La claves de CA (al igual que el resto de las claves) tienen un tiempo de vida finito y deberán actualizarse periódicamente.

La CA expide certificados especiales (véase 7.2.3) para ayudar a las entidades terminales existentes que mantienen el certificado de CA autofirmado antiguo en la ejecución de una transición segura al nuevo certificado de CA autofirmado, y para ayudar a las nuevas entidades terminales que mantendrán el nuevo certificado de CA autofirmado en la adquisición del antiguo de manera segura para la verificación de los datos existentes.

Cuando una CA actualiza su propia pareja de claves pública/privada se puede utilizar el mensaje siguiente para notificar este evento a las entidades:

- Notificación de actualización de claves de CA (**CAKeyUpdAnn**).

7.1.3 Certificación de claves

Una entidad terminal inicializada puede solicitar un certificado en cualquier momento (como parte de un procedimiento de actualización o para cualquier otro fin). Esta petición se efectuará mediante el mensaje petición de certificación. Si la entidad terminal ya posee una pareja de claves de firma (con un certificado de verificación correspondiente), este mensaje será protegido generalmente por la firma digital de la entidad. La CA devuelve el nuevo certificado (si la petición tiene éxito) en un mensaje **CertRep**.

Una entidad registra con una TTP y solicita un certificado utilizando los siguientes mensajes:

- Petición de registro/certificación (**CertReq**).
- Respuesta de registro/certificación (**CertRep**).

Se puede utilizar opcionalmente una petición/respuesta de registro/certificación inicial (**InitReq**/**InitRep**).

Cuando la entidad identifica la necesidad de una nueva pareja de claves y el certificado asociado, se pueden cursar opcionalmente las peticiones de certificación/registro utilizando los siguientes mensajes:

- Petición de registro/certificación después de la actualización de claves (**KeyUpdReq**).
- Respuesta de registro/certificación después de la actualización de claves (**KeyUpdRep**).

La entidad incluye en la petición toda la información que haya de colocarse en el certificado, incluido su nombre o un seudónimo. Cierta información de la respuesta, como el tiempo de caducidad, puede diferir de la solicitada por la entidad.

Con el fin de completar el proceso de registro/certificación puede ser necesario efectuar más intercambios para atestiguar la validez del nombre de la entidad y de otra información que haya de ser incluida en el certificado o esté involucrada por el mismo.

Si el par de claves es generado por la entidad, puede entonces transferirse la clave pública a la TTP en la petición. Si el par de claves es generado por la TTP puede transferirse en la respuesta a la entidad la clave privada criptada.

Si el par de claves es generado por la entidad para la generación y verificación de firmas digitales, puede entonces incluirse una firma en la petición para comprobar la propiedad de la clave privada.

Si es necesario, la entidad puede enviar el siguiente mensaje para demostrar la aceptación del certificado.

- Confirmar (**PKIConfirm**).

7.1.4 Notificación de certificados

Después de completarse el proceso de registro/certificación, incluida si fuera necesario la recepción de un mensaje **PKIConfirm**, la TTP prepara el certificado y lo pone a disposición de otras partes. Esto puede alcanzarse mediante varios mecanismos que incluyen:

a) la colocación del certificado en un depósito, tal como un servicio de directorio o un servidor Web;

b) la transferencia del certificado a otras entidades de las cuales se conoce que pueden solicitar el certificado (por ejemplo, miembros de una comunidad de usuarios conocida).

Alternativamente, la entidad puede poner su certificado a disposición de otras partes incluyéndolo con cualquier información protegida.

Una TTP puede enviar certificados a entidades utilizando el siguiente mensaje:

– Notificación de certificado **CertAnn**.

Los intercambios requeridos para colocar los certificados en un depósito dependerán de la forma del depósito utilizado

Si a una parte que solicita un certificado aún no le ha sido proporcionado éste mediante un mensaje de notificación de certificado o junto con los datos protegidos, el certificado puede entonces obtenerse de un depósito, tal como un servicio de directorio o un servidos Web.

Los intercambios necesarios para obtener un certificado a partir de un depósito dependerán de la forma del depósito utilizado.

7.1.5 Distribución de claves

En la distribución de claves, la clave privada debe tratarse de modo diferente a la clave pública.

Clave privada: Para la clave privada, ha de hacerse una diferenciación en cuanto a quién genera el par de claves. Si una entidad genera su propia pareja, no es necesaria la distribución de la clave privada. Si el par de claves no es generado por la entidad (es decir, es generada por la TTP), la clave privada debe distribuirse a la entidad de un modo seguro.

NOTA – En 8.4.4 se presenta un método de distribución de claves privadas. En ISO/CEI 11770 (todas las partes) se recoge información general sobre la distribución de claves privadas y se introducen diferentes métodos para la distribución de claves.

Clave pública: Las claves públicas de entidades se distribuyen junto con los certificados a ellas asignados. Son mensajes asociados los relativos a la notificación y la recuperación de certificados.

7.1.6 Revocación de claves/certificados

La revocación de certificados debe efectuarse siempre que se tenga alguna sospecha de que hay algún error ene las claves (la clave privada está comprometida), o si se producen cambios relativos al certificado asociado con la clave pública (cambio de nombre, finalización de su empleo en una organización, etc.). Si una entidad terminal (o cualquier otra entidad autorizada) desea revocar la clave pública y el certificado asociado con ella, envía una petición **RevReq** a la TTP, la cual contesta con una **RevRep**.

Si la TTP ha revocado, o está a punto de hacerlo, el certificado solicitado, puede emitir una notificación de este evento (**RevAnn**). En particular, este puede ser el caso en que la petición de revocación no ha sido emitida por la entidad.

En combinación con las actividades de revocación, cambian las inserciones en la CRL. Para notificar estos cambios, la TTP emite una notificación de la nueva CRL (**CRLAnn**). Esta información debe enviarse a todas las entidades afectadas.

7.2 Hipótesis y restricciones aplicables a algunos de los servicios

7.2.1 Registro/certificación inicial

Se pueden utilizar muchos esquemas para conseguir el registro y la certificación iniciales de entidades terminales. Ningún método es adecuado a todas las situaciones debido a la gama de políticas que una CA puede implementar y a la variedad de tipos de entidades terminales que se pueden dar.

Puede, sin embargo, hacerse una clasificación de los esquemas de registro/certificación soportados por esta Recomendación | Norma Internacional. Esta clasificación se aplica a la situación en que la entidad terminal en cuestión no ha tenido un contacto anterior con la PKI. Si la entidad terminal ya posee claves certificadas, se pueden entonces introducir algunas simplificaciones/alternativas.

Para diferenciar los distintos esquemas se pueden aplicar los siguientes criterios:

a) Autenticación por la RA

Durante el registro/certificación inicial, la entidad terminal se entiende con la RA.

Puede estar separada o combinada con la CA. La RA puede tener necesidad de autenticar al peticionario en algún momento antes de la expedición de un certificado. Esta autenticación puede producirse antes del intercambio de ningún mensaje con la RA o después de que se hayan intercambiado algunos mensajes con la RA. Ello se puede realizar mediante un contacto directo (por ejemplo, la presentación de una tarjeta ID, un permiso de conducir, un pasaporte) o utilizando otros medios fuera de banda (por ejemplo, procedimiento de retrollamada, reconocimiento de la voz).

b) Información inicial distribuida por medios fuera de banda

La información pública y/o secreta ha de distribuirse por medios fuera de banda antes de que se pueda utilizar ningún protocolo. La información pública puede estar formada por el nombre de una RA (o de una CA), su ubicación y su valor de clave pública y algoritmo, o un puntero hacia el certificado autofirmado y un troceo del mismo. La información secreta puede estar constituida por un identificador de la entidad terminal y una clave de autenticación inicial. La clave de autenticación inicial puede utilizarse entonces para proteger los mensajes PKI pertinentes.

La información pública puede ser compartida libremente por todos los usuarios de una comunidad, y la garantía de su integridad es suficiente. A título de ejemplo se puede recoger esta información en un CD-ROM.

A la entidad terminal o a la organización que representa a la entidad terminal se le dará la información secreta como resultado de la autenticación inicial (véase anteriormente). La parte secreta de la información deberá ser protegida contra su revelación.

c) Autenticación del origen de un mensaje de entidad terminal

Los mensajes en línea producidos por la entidad terminal que requieren ser certificados y enviados por esta a la RA o CA, pueden ser autenticados o no. Se puede asegurar un procedimiento de registro/certificación inicial cuando los mensajes procedentes de la entidad terminal son autenticados a través de algún medio fuera de banda (por ejemplo, una visita posterior).

d) Ubicación de la generación de claves

La generación de pares de claves puede tener tres ubicaciones diferentes: la entidad terminal, una RA o una CA.

NOTA – Esto no excluye un servicio de generación de claves independiente. El par de claves real puede haberse generado en cualquier punto y ser transportado a la entidad terminal, la RA o la CA utilizando un protocolo (propietario o estándar) de petición/respuesta de generación de claves, cuyo estudio cae fuera del alcance de esta Recomendación | Norma Internacional.

e) Confirmación eficaz de la certificación

Después de la creación de un certificado inicial para una entidad terminal, puede conseguirse una seguridad adicional haciendo que la entidad terminal confirme explícitamente la recepción exitosa del mensaje que contiene el certificado (o que indica la creación de éste). Naturalmente, debe protegerse este mensaje (mediante una clave de autenticación inicial o por otros medios). Esto proporciona dos posibilidades más: confirmado o no confirmado.

El criterio anterior permite un gran número de esquemas de registro/certificación iniciales. Aquí se describen algunos de ellos.

7.2.1.1 Esquema centralizado

En términos de los criterios anteriores, este esquema tiene lugar cuando:

- se produce una autenticación inicial en la RA antes de ningún intercambio;
- la información secreta es distribuida por la RA por métodos fuera de banda durante la autenticación inicial;
- no se solicita ningún mensaje en línea de la entidad terminal;
- se produce la generación de parejas de claves en la CA o la RA certificante;
- no se requiere mensaje de confirmación.

En términos de flujo de mensajes, puesto que la generación de parejas de claves se produce en la CA o RA certificante, debe devolverse el PSE a la entidad terminal. El único mensaje requerido es enviado desde la CA a la entidad terminal y contiene el PSE completo para la entidad terminal. La información secreta distribuida anteriormente por métodos fuera de banda permite a la entidad terminal autenticar el mensaje recibido y decriptar los valores criptados. Una alternativa consiste en la entrega física de un testigo PSE.

7.2.1.2 Esquema de autenticación previa

En términos de los criterios anteriores, este esquema tiene lugar cuando:

- se produce la autenticación inicial en la RA antes de ningún intercambio;
- la información secreta es distribuida por la RA por métodos fuera de banda durante la autenticación inicial;
- los mensajes procedentes de la entidad terminal son autenticados utilizando esta información secreta;
- la generación de pares de claves tiene lugar en la entidad terminal;
- se requiere mensaje de confirmación.

En términos de flujo de mensajes, este esquema es el siguiente:

Entidad terminal		**RA/CA**
	Autenticación de la entidad terminal a la RA.	
	Distribución fuera de banda de la clave de autenticación inicial (IAK) y del valor de referencia (RA/CA → EE) por la RA.	
Generación de claves Creación de petición de certificación Protección de petición con IAK		
	→→petición de certificación→→	
		Verificar petición Procesar petición Crear certificado Crear respuesta
	←←respuesta de certificación←←	
Trocear respuesta Crear confirmación		
	→→mensaje de confirmación→→	Verificar confirmación

(Cuando la verificación del mensaje de confirmación falla, la RA/CA deberá revocar si es necesario el certificado recientemente enviado.)

7.2.1.3 Esquema de autenticación posterior

En términos del criterio anterior, este esquema tiene lugar cuando:

- se produce una autenticación inicial en la RA después de los intercambios;
- la información pública acerca de RA es distribuida por métodos fuera de banda (es decir, no se distribuye ninguna información secreta);
- los mensajes procedentes de la entidad terminal no son autenticados;
- la generación de parejas de claves se produce en la entidad terminal;
- no se requiere mensaje de confirmación.

En términos de flujos de mensajes, el esquema es el siguiente:

Entidad terminal		RA/CA
	Distribución fuera de banda de información pública (RA/CA → EE)	
Generación de claves Creación de petición de certificación Ninguna protección de la petición		
	→→petición de certificación→→	
		Verificar petición Procesar petición Crear respuesta
	←←respuesta de certificación←←	
Trocear respuesta Verificar el origen de la respuesta Extraer el número de registro		
	Presentación del número de registro y autenticación de la entidad terminal a la RA utilizando métodos fuera de banda.	
		Crear certificado

La entidad terminal puede en cualquier momento rellenar una petición tan pronto como posea la información correcta acerca de la RA (en particular, su clave pública). La RA asigna una número de registro exclusivo a cualquier petición de certificación no autenticada que se reciba. Puesto que la respuesta de certificación emitida por la RA está firmada, la entidad terminal puede verificar su origen y extraer el número de registro. La entidad terminal puede a continuación presentar este número a la RA y autenticarlo ella misma utilizando algunos de los métodos fuera de banda.

Las ventajas de este esquema son las siguientes:

- no se requiere ningún contacto anticipado con la RA;
- no se manipula nunca ninguna información secreta.

7.2.2 Prueba de posesión (POP, *proof of possession*) de clave privada

Para evitar algunas agresiones y permitir que una CA/RA compruebe adecuadamente la validez de la vinculación entre una entidad terminal y un par de claves de firma, las operaciones de gestión PKI aquí especificadas posibilitan que una entidad terminal pruebe que está en posesión de (o sea, que puede de utilizar) la clave de firma privada correspondiente a la clave de firma pública para la cual se solicita el certificado. Una CA/RA determinada puede elegir libremente el modo de imponer la POP [por ejemplo, medios procedimentales fuera de banda versus mensajes ("dentro de banda") de protocolo de gestión de certificados] en sus intercambios de certificación (lo que quiere decir que puede constituir un tema de política). Sin embargo, la conformidad de las CA/RA deberá activar la POP por algunos medios.

Esta Recomendación | Norma Internacional tiene en cuenta explícitamente los casos en que una entidad terminal suministra la prueba correspondiente a una RA y la RA atestigua seguidamente a la CA que la prueba requerida ha sido recibida y validada. Por ejemplo, una entidad terminal que desee conseguir un certificado para una clave de firma puede enviar la firma adecuada a la RA, la cual sencillamente notifica a la CA pertinente que la entidad terminal ha suministrado la prueba requerida. Tal situación puede, por supuesto, ser desautorizada por algunas políticas (por ejemplo, las CA pueden ser las únicas entidades autorizadas para verificar la POP durante la certificación).

Puesto que el par de claves de la entidad es generado para fines de firma digital, la entidad terminal puede firmar un valor adecuado para probar la posesión de la clave privada. El valor debe escogerse de modo que se impida que los atacantes tengan éxito utilizando las firmas antiguas de la entidad terminal.

7.2.3 Actualización de claves CA de confianza directa

El procedimiento descrito aquí se basa en que la CA protege su nueva clave pública utilizando su clave privada anterior, y viceversa. Por ello, cuando una CA actualiza su par de claves deberá generar dos valores de atributo **cACertificate** extra si los certificados se ponen a disposición utilizando una directorio X.500 (de un total de cuatro: **OldWithOld**; **OldWithNew**; **NewWithOld**; y **NewWithNew**).

Cuando una CA cambia su par de claves, las entidades que han adquirido la clave pública de CA antigua por un método "fuera de banda" son las más afectadas. Son estas entidades terminales las que necesitan acceder a la nueva clave pública de CA protegida con la clave privada de CA antigua. No obstante, sólo precisarán realizar lo anterior durante un periodo limitado (hasta que hayan adquirido la nueva clave pública de CA a través del mecanismo "fuera de banda"). Esto normalmente se conseguirá de manera fácil cuando expiran estos certificados de entidades terminales.

La estructura de datos empleada para proteger las claves públicas de CA nueva y antigua es un certificado estándar (el cual puede contener también extensiones). No se necesitan nuevas estructuras de datos.

NOTA 1 – Este esquema no utiliza ninguna de las extensiones X.509 v3 o v4 ya que está concebido para trabajar incluso con certificados de la versión 1. Puede aprovecharse la presencia de la extensión **KeyIdentifier** para mejorar la eficacia.

NOTA 2 – Si bien el esquema puede generalizarse para cubrir los casos en que la CA actualiza su par de claves, más de una vez a lo largo del periodo de validez de uno de sus certificados de entidades terminales el valor de esta generalización es dudoso. El no disponer de esta generalización significa sencillamente que el periodo de validez de un par de claves CA debe ser mayor que el periodo de validez de cualquier certificado emitido por dicha CA utilizando aquel par de claves.

NOTA 3 – Este esquema obliga a las entidades terminales a adquirir la nueva clave pública de CA cuando caduque el último certificado de su propiedad que haya sido firmado con la clave privada CA antigua (vía los métodos "fuera de banda"). Los servicios de actualización de claves y/o certificados que tienen lugar ocurren en otros momentos no precisan necesariamente de esta acción (dependiendo del equipo de la entidad terminal).

CA – acciones

Para cambiar la clave de la CA, ésta efectúa las acciones siguientes:

1) Generar un nuevo par de claves.

2) Crear un certificado que contenga la clave pública de CA antigua firmada y la nueva clave privada (certificado "antigua con nueva").

   Esto permite a las entidades terminales cuyo PSE contenga la clave pública nueva de la CA verificar la clave CA antigua y los certificados firmados con esta clave.

3) Crear un certificado que contenga la clave pública de CA nueva firmada y la clave privada antigua (certificado "nueva con antigua").

   Esto permite a las entidades terminales cuyo PSE contenga la clave pública antigua de la CA verificar la clave CA nueva y los certificados firmados con esta clave.

4) Crear un certificado que contenga la clave pública de CA nueva firmada y la clave privada nueva (certificado "nueva con nueva").

   Esto permite a las entidades terminales cuyo PSE contenga la clave pública antigua de la CA importar este certificado autofirmado nuevo.

5) Publicar estos nuevos certificados a través del directorio y/o por otros medios (posiblemente utilizando un mensaje **CAKeyUpdAnn**).

6) Exportar la clave pública de CA nueva de modo que las entidades terminales puedan adquirirla por mecanismos "fuera de banda" (si es necesario).

La clave privada CA antigua ya no se requiere. La clave pública de CA antigua permanecerá sin embargo en uso durante algún tiempo. La clave pública de CA antigua no se requerirá por más tiempo (salvo en caso de no repudio) cuando todas las entidades terminales de esta CA hayan adquirido con garantía la nueva clave pública de CA.

El certificado "antigua con nueva" tendrá un periodo de validez que arranca en el momento de la generación del par de claves antiguo y el finaliza en la fecha de caducidad de la clave pública antigua.

El certificado "nueva con antigua" tendrá un periodo de validez que arranca en el momento de la generación del par de claves nuevo y termina en el momento en el cual todas las entidades terminales de esta CA posean con garantía la clave pública de CA nueva (como máximo, en la fecha de expiración de la clave pública antigua).

El certificado "nueva con nueva" tendrá un periodo de validez que arranca en el momento de la generación del par de claves nuevo y termina en el momento en el cual la CA actualizará de nuevo su par de claves.

El procedimiento de publicación de estos certificados se utiliza también para las pruebas de revocación puesto que la CA puede haber firmado la CRL utilizando una clave privada más reciente que la que se encuentra dentro del PSE de la entidad.

7.2.4 Certificación cruzada

La subcláusula que sigue describe un posible esquema de certificación cruzada. No obstante, se pueden utilizar otros esquemas dependiendo de las políticas de los TTP que figuren en la certificación cruzada. Véase, en el anexo A, un análisis de los escenarios posibles.

7.2.4.1 Esquema petición-respuesta unidireccional

El esquema de certificación cruzada es normalmente un servicio unidireccional; esto es, cuando tiene éxito, el resultado de este servicio es la creación de un certificado cruzado nuevo. Si lo que se necesita es que se creen certificados cruzados en "ambas direcciones", entonces la autoridad de certificación CA debe iniciar a su vez un servicio de certificación cruzada (o utilizar otro esquema).

Este esquema es adecuado cuando las dos CA en cuestión pueden ya verificar cada una las firmas de la otra (tienen algunos puntos comunes de confianza) o cuando hay una verificación fuera de banda del origen de la petición de certificación.

Descripción detallada

La certificación cruzada es iniciada por la CA1. La CA1 identifica la CA (o sea, CA2) con la que desea cruzar certificaciones y el equipo de CA1 genera un código de autorización. La CA1 transfiere este código de autorización por métodos fuera de banda a la CA2. La CA2 ingresa el código de autorización para iniciar el intercambio en línea.

El código de autorización se utiliza para la autenticación y la integridad. Esto se lleva a cabo generando una clave simétrica basada en el código de autorización y utilizando una clave simétrica para la generación de los códigos de autenticación de mensajes (MAC, *message authentication codes*) en todos los mensajes intercambiados.

La CA2 inicia el intercambio generando un número aleatorio (número aleatorio del peticionario). La CA2 envía entonces a la CA1 el mensaje de petición de certificación cruzada (**CrossCertReq**). Los campos de este mensaje se protegen contra su modificación con un MAC basado en el código de autorización.

Tras la recepción del mensaje de petición de certificación cruzada, la CA1 comprueba la versión del protocolo, guarda el número aleatorio del peticionario, genera su propio número aleatorio (número aleatorio del respondedor) y valida el MAC. Genera luego (y archiva, si se desea) un nuevo certificado de peticionario que contiene la clave pública de la CA2 y es firmado con la clave privada de firma de la CA1. La CA1 responde con el mensaje respuesta de certificación cruzada (**CrossCertRes**). Los campos de este mensaje se protegen contra su modificación con un MAC basado en el código de autorización.

Tras la recepción del mensaje respuesta de certificación cruzada, la CA2 comprueba que su propio tiempo de sistema está próximo al tiempo de sistema de la CA1, comprueba los números aleatorios recibidos y valida el MAC. La CA2 responde con el mensaje **PKIConfirm**. Los campos de este mensaje se protegen contra su modificación con un MAC basado en el código de autorización. La CA2 escribe el certificado del peticionario al depósito.

Tras la recepción del mensaje **PKIConfirm**, CA1 comprueba los números aleatorios y valida el MAC.

NOTA 1 – El mensaje petición de certificación cruzada debe contener una petición de certificación "completa", esto es, todos los campos (que incluyen, por ejemplo, una extensión **BasicConstraints**) deben ser especificados por la CA2.

NOTA 2 – El mensaje respuesta de certificación cruzada debe contener el certificado de verificación de la CA1, si está presente, la CA2 debe entonces verificar este certificado (por ejemplo, mediante el mecanismo "fuera de banda").

8 Estructuras de datos para los mensajes de gestión de certificados

Esta cláusula contiene descripciones de las estructuras de datos requeridas para los mensajes de gestión de certificados. Las estructuras de datos se especificarán de conformidad con los documentos RFC 2510 y RFC 2511. La presente Recomendación | Norma Internacional utiliza sintaxis ASN.1 e importa las definiciones ASN.1 de la Rec. UIT-T X.509 | ISO/CEI 9594-8.

8.1 Mensaje global

Todos los mensajes utilizados en esta Recomendación | Norma Internacional para la gestión de certificados emplean la estructura siguiente:

```
PKIMessage ::= SEQUENCE {
    header          PKIHeader,
    body            PKIBody,
    protection      [0] PKIProtection   OPTIONAL,
    extraCerts      [1] SEQUENCE SIZE (1..MAX) OF Certificate OPTIONAL
}
```

En encabezamiento, cuerpo y protección del mensaje PKI se examinarán con detalle en las subcláusulas a continuación.

El campo de certificados extra **extraCerts** puede contener certificados que sean útiles al receptor. Por ejemplo, estos certificados extra pueden ser utilizados por una TTP para entregarlos a una entidad terminal que necesita verificar su propio certificado nuevo (si la TTP que expidió el certificado de la entidad terminal no es una CA de confianza directa para la entidad terminal). Debe también señalarse que este campo no contiene necesariamente un trayecto de certificación. El receptor puede tener que clasificar, seleccionar, o someter a cualquier otro proceso los certificados extra con el fin de utilizarlos.

8.1.1 Encabezamiento de un mensaje PKI

Todos los mensajes TTP necesitan alguna información de encabezamiento que permita la identificación de las transacciones. Alguna información de este tipo puede encontrarse en un sobre específico del transporte; sin embargo, si el mensaje está protegido esta información también está protegida (es decir, no se establece ninguna hipótesis acerca de la seguridad del transporte).

Para contener esta información se utiliza la siguiente estructura de datos:

```
PKIHeader ::= SEQUENCE {
                    pvno            INTEGER     { version2 (1) },
                    sender          GeneralName,
                    recipient       GeneralName,
                    messageTime     [0] GeneralizedTime         OPTIONAL,
                    protectionAlg   [1] AlgorithmIdentifier     OPTIONAL,
                    senderKID       [2] KeyIdentifier           OPTIONAL,
                    recipKID        [3] KeyIdentifier           OPTIONAL,
                    transactionID   [4] OCTET STRING            OPTIONAL,
                    senderNonce     [5] OCTET STRING            OPTIONAL,
                    recipNonce      [6] OCTET STRING            OPTIONAL,
                    freeText        [7] PKIFreeText             OPTIONAL,
                    generalInfo     [8] SEQUENCE SIZE (1..MAX) OF InfoTypeAndValue OPTIONAL
                            -- this may be used to convey context-specific information
}
```

El campo **pvno** es fijo para esta versión del documento TTP. El campo **sender** contiene el nombre del emisor del **PKIMessage**. Este nombre (junto con **senderKID**, si ha sido suministrado) debe poderse utilizar para verificar la protección del mensaje.

Si no se conoce nada acerca del emisor en la entidad emisora (por ejemplo, DN, nombre del correo electrónico, dirección IP, etc.), entonces el campo "sender" ("emisor") deberá contener un valor "NULL"; esto es, la SEQUENCE OF relativa a los nombres distinguidos es de longitud cero. En tal caso el campo **senderKID** deberá mantener un identificador (es decir, un número de referencia) que indica al receptor la información secreta compartida apropiada para su utilización en la verificación del mensaje.

El campo **recipient** contiene el nombre del receptor del **PKIMessage**. Este nombre (junto con **recipKID**, si ha sido suministrado) debe poderse utilizar para verificar la protección del mensaje. El campo **messageTime** contiene la hora en la cual el emisor creó el mensaje (utilizado cuando el emisor cree que el transporte será "adecuado", es decir, que la hora será todavía importante tras la recepción). Esto puede ser útil para que las entidades terminales puedan corregir sus horas locales para que sean coherentes con la hora de un sistema central. El campo **protectionAlg** especifica el algoritmo utilizado para proteger el mensaje. Si no se suministran bits de protección (**PKIProtection** es opcional), deberá omitirse este campo; si se suministran bits de protección deberá suministrarse este campo.

Los campos **senderKID** y **recipKID** son aplicables para indicar cuales son las claves que se han empleado para proteger el mensaje. El campo **transactionID** dentro del encabezamiento del mensaje es aplicable al receptor de un mensaje de respuesta, el cual lo puede correlacionar con una petición anteriormente enviada. Por ejemplo, en el caso de una RA puede haber muchas peticiones pendientes en un momento dado.

Los campos **senderNonce** y **recipNonce** protegen al **PKIMessage** contra la reanudación de ataques. Los campos **Nonces** son utilizados para la protección de reanudación, **senderNonce** es insertado por el creador de este mensaje; **recipNonce** es un **nonce** insertado previamente en un mensaje relacionado por el receptor deseado de este mensaje. El campo **freeText** puede utilizarse para enviar al receptor un mensaje legible por el ser humano. La estructura de este campo es:

```
PKIFreeText ::= SEQUENCE SIZE (1..MAX) OF UTF8String
        -- text encoded as UTF-8 String (NOTE – Each UTF8String should
        -- include an RFC 1766 language tag to indicate the language
        -- of the contained text)
```

8.1.2 Cuerpo de un mensaje PKI

```
PKIBody ::= CHOICE {      -- message-specific body elements
    ir          [0]  CertReqMessages,        --Initialization Request (InitReq)
    ip          [1]  CertRepMessage,         --Initialization Response (InitRep)
    cr          [2]  CertReqMessages,        --Certification Request (CertReq)
    cp          [3]  CertRepMessage,         --Certification Response (CertRep)
    p10cr       [4]  CertificationRequest,   --PKCS #10 Cert.- Request (alternative form for CertRep)
                     -- the certification request as defined in PKCS #10 for compatibility
    kur         [7]  CertReqMessages,        --Key Update Request (KeyUpdReq)
    kup         [8]  CertRepMessage,         --Key Update Response (KeyUpdRep)
    rr          [11] RevReqContent,          --Revocation Request (RevReq)
    rp          [12] RevRepContent,          --Revocation Response (RevRep)
    ccr         [13] CertReqMessages,        --Cross-Cert. Request (CrossCertReq)
    ccp         [14] CertRepMessage,         --Cross-Cert. Response (CrossCertRep)
    ckuann      [15] CAKeyUpdAnnContent,     --CA Key Update Ann. (CAKeyUpdAnn)
    cann        [16] CertAnnContent,         --Certificate Ann. (CertAnn)
    rann        [17] RevAnnContent,          --Revocation Ann. (RevAnn)
    crlann      [18] CRLAnnContent,          --CRL Announcement (CRLAnn)
    conf        [19] PKIConfirmContent,      --PKI Confirmation (PKIConfirm)
    nested      [20] NestedMessageContent,   --Nested Message
    genm        [21] GenMsgContent,          --General Message (GenMsg)
    genp        [22] GenRepContent,          --General Response (GenRep)
    error       [23] ErrorMsgContent,        --Error Message (ErrorMsg)
}
```

En las subcláusulas 8.3 y 8.4 más adelante se describen los tipos específicos.

8.1.3 Protección de un mensaje PKI

Algunos mensajes PKI se protegerán en cuanto a su integridad. (Hay que señalar que si se utiliza un algoritmo asimétrico para proteger un mensaje y el componente público pertinente ya ha sido certificado, el origen del mensaje también puede ser autenticado. Por otra parte, si el componente público no ha sido certificado el origen del mensaje no puede ser autenticado automáticamente, pero puede ser autenticado por métodos fuera de banda).

Cuando se aplica la protección se utiliza la estructura siguiente:

```
PKIProtection ::= BIT STRING
```

La entrada para el cálculo de PKIProtection es la codificación DER (reglas de codificación distinguida) de la estructura de datos siguiente:

```
ProtectedPart ::= SEQUENCE {
    header      PKIHeader,
    body        PKIBody
}
```

Puede haber casos en los que no se utilizará de manera deliberada la BIT STRING (CADENA DE BITS) PKIProtection para proteger un mensaje (es decir, se omite este campo OPTIONAL) porque se aplicará en su lugar otra protección externa a esta especificación. Tal elección se permite explícitamente en esta Recomendación | Norma Internacional.

Hay que señalar, no obstante, que muchos de tales mecanismos externos requieren que la entidad terminal posea ya un certificado de claves públicas, y/o un número distinguido único, y/u otra información relativa a la infraestructura. Por tanto, estos mecanismos no pueden ser apropiados para el registro inicial o cualquier otro proceso con características de "secuencia de arranque". Para estos casos puede ser necesario utilizar el parámetro PKIProtection.

Dependiendo de las circunstancias, los bits de PKIProtection pueden contener un código de autenticación de mensaje (MAC) o una firma. Pueden darse los siguientes casos:

Caso 1: Información secreta compartida

En este caso el emisor y el receptor comparten información secreta (establecida por métodos fuera de banda o a partir de un servicio de gestión TTP anterior). PKIProtection contendrá un valor MAC.

El protectionAlg será el siguiente:

```
PasswordBasedMac ::= OBJECT IDENTIFIER    --{1 2 840 113533 7 66 13}
PBMParameter ::= SEQUENCE {
    salt            OCTET STRING,
    owf             AlgorithmIdentifier,
    -- AlgorithmIdentifier  for a One-Way Function (SHA-1 recommended)
    iterationCount          INTEGER,
    -- number of times the OWF is applied
    mac             AlgorithmIdentifier
    -- the MAC AlgorithmIdentifier (e.g. DES-MAC, Triple-DES-MAC as in PKCS #11,
}   -- or HMAC as in RFC2104, RFC2202)
```

En el **protectionAlg** anterior el valor salt se añade a la entrada secreta compartida. La OWF se aplica entonces **iterationCount** veces, donde el secreto tratado con salt es la entrada de la primera iteración y, para cada iteración sucesiva, la entrada se fija para que sea la salida de la iteración anterior. La salida de la iteración final (denominada "BASEKEY" para facilitar la referencia, con un tamaño de "H") es la que se utiliza para formar la clave simétrica. Si el algoritmo MAC requiere una clave de K bits y K <= H, entonces se utilizan los K bits más significativos de BASEKEY. Si K > H, entonces se utilizan todos los bits de BASEKEY para los H bits más significativos de la clave, OWF("1" || BASEKEY) se utiliza para los H bits siguientes más significativos de la clave, OWF("2" || BASEKEY) se utiliza para los H bits siguientes más significativos de la clave, y así sucesivamente, hasta que se hayan obtenidos todos los K bits. [Aquí "N" es el byte en código ASCII del número N y "||" representa concatenación.]

Caso 2: Pares de claves Diffie-Hellman

Cuando el emisor y el receptor poseen certificados Diffie-Hellman con parámetros DH compatibles, para proteger el mensaje la entidad terminal debe generar una clave simétrica basada en el valor de su clave DH privada y en la clave pública DH del receptor del mensaje. PKIProtection contendrá un valor MAC tratado con esta clave simétrica derivada.

```
DHBasedMac ::= OBJECT IDENTIFIER --{1 2 840 113533 7 66 30}
  DHBMParameter ::= SEQUENCE {
    owf             AlgorithmIdentifier,
-- AlgorithmIdentifier for a One-Way Function (SHA-1 recommended)
    mac             AlgorithmIdentifier
    -- the MAC AlgorithmIdentifier  (e.g. DES-MAC, Triple-DES-MAC as in PKCS#11,
}   -- or HMAC as in RFC2104, RFC2202)
```

En el protectionAlg anterior, OWF se aplica al resultado del cálculo Diffie-Hellman. La salida OWF (denominada "BASEKEY" para facilitar la referencia, con un tamaño de "H") es la que se utiliza para formar la clave simétrica. Si el algoritmo MAC requiere una clave de K bits y K <= H, entonces se utilizan los K bits más significativos de BASEKEY. Si K > H, entonces se utilizan todos los bit de BASEKEY para los H bits más significativos de la clave, OWF("1" || BASEKEY) se utiliza para los H bits siguientes más significativos de la clave, OWF("2" || BASEKEY) se utiliza para los H bits siguientes más significativos de la clave, y así sucesivamente, hasta que se hayan obtenidos todos los K bits. [Aquí "N" es el byte en código ASCII del número N y "||" representa concatenación.]

Caso 3: Firma digital

Cuando el emisor posee un par de claves de firma ello puede simplificar la firma del mensaje. **PKIProtection** contendrá el valor firma y el **protectionAlg** será un **AlgorithmIdentifier** para una firma digital (por ejemplo, **md5WithRSAEncryption** o **dsaWithSha-1**).

Caso 4: Protección múltiple

En los casos en que una entidad terminal envía un mensaje protegido a una RA, la RA puede reenviar el menaje a una CA, adjuntando su propia protección (que puede ser un MAC o una firma, dependiendo de la información y certificados compartidos entre la RA y la CA). Esto se realiza anidando el mensaje completo enviado por la entidad terminal dentro de un nuevo mensaje TTP. La estructura utilizada es la siguiente.

```
NestedMessageContent ::= PKIMessage
```

8.2 Estructuras de datos comunes

En esta subcláusula se definen algunas estructuras de datos que se utilizan en más de un mensaje PKI.

8.2.1.1 Contenido del certificado pedido

Diversos mensajes de gestión TTP establecen que el originador del mensaje indique algunos de los campos que han de estar presentes en el certificado. La estructura **CertTemplate** permite a una entidad terminal o RA que especifique cuántos campos desea acerca del certificado que requiere. **CertTemplate** es idéntica a un certificado pero con todos los campos opcionales.

Hay que señalar que, incluso si el originador especifica completamente el contenido de un certificado que solicita, una CA es libre de modificar los campos dentro del certificado realmente expedido.

En relación con la sintaxis de **CertTemplate,** véase 8.3.

8.2.2 Valores criptados

Cuando se envían valores criptados (restringidos en esta Recomendación | Norma Internacional a claves privadas o a certificados) en mensajes PKI, se aplica la estructura de datos **EncryptedValue**.

En relación con la sintaxis de **EncryptedValue**, véase 8.6.1.

El uso de esta estructura de datos requiere que el creador y el receptor deseado sean capaces, respectivamente, de efectuar las operaciones de criptación y decriptación. Generalmente, esto significará que el emisor y el receptor dispongan de una clave secreta compartida, o sean capaces de generarla.

Si el receptor del PKIMessage ya posee una clave privada utilizable para la decriptación, el campo **encSymmKey** puede contener una clave de sesión criptada utilizando la clave pública del receptor.

8.2.3 Información de fallo y códigos de estado de los mensajes PKI

Todos los mensajes de respuesta incluirán alguna información de estado. Se definen los siguientes valores:

```
PKIStatus ::= INTEGER {
    granted                 (0),
    -- you got exactly what you asked for
    grantedWithMods         (1),
    -- you got something like what you asked for; the
    -- requester is responsible for ascertaining the differences
    rejection               (2),
    -- you don't get it, more information elsewhere in the message
    waiting                 (3),
    -- the request body part has not yet been processed,
    -- expect to hear more later
    revocationWarning       (4),
    -- this message contains a warning that a revocation is
    -- imminent
    revocationNotification  (5),
    -- notification that a revocation has occurred
    keyUpdateWarning        (6)
    -- update already done for the oldCertId specified in
    -- the key update request message
}
```

Los respondedores pueden utilizar la siguiente sintaxis para proporcionar más información sobre los casos de fallo.

```
PKIFailureInfo ::= BIT STRING {
-- since a request can fail in more than one way!
-- More codes may be added in the future if/when required.
    badAlg                  (0),
    -- unrecognized or unsupported Algorithm Identifier
    badMessageCheck         (1),
    -- integrity check failed (e.g. signature did not verify)
    badRequest              (2),
    -- transaction not permitted or supported
```

```
    badTime                 (3),
    -- messageTime was not sufficiently close to the system time,
    -- as defined by local policy
    badCertId               (4),
    -- no certificate could be found matching the provided criteria
    badDataFormat           (5),
    -- the data submitted has the wrong format
    wrongAuthority          (6),
    -- the authority indicated in the request is different from the
    -- one creating the response token
    incorrectData           (7),
    -- the requester's data is incorrect (used for notary services)
    missingTimeStamp        (8)
    -- when the timestamp is missing but should be there (by policy)
}

PKIStatusInfo ::= SEQUENCE {
    status          PKIStatus,
    statusString    PKIFreeText     OPTIONAL,
    failInfo        PKIFailureInfo  OPTIONAL
}
```

8.2.4 Identificación de certificados

Para identificar certificados concretos se utiliza la estructura de datos **CertId**. En 8.3 se recoge la sintaxis de **CertId**.

8.2.5 Clave pública de CA de confianza directa "fuera de banda"

Cada CA de confianza directa debe tener capacidad para publicar su clave pública vigente a través de algunos métodos "fuera de banda". Si bien tales mecanismos caen fuera del alcance de esta Recomendación | Norma Internacional, se definen las estructuras de datos que los mismos pueden soportar.

En general se dispone de dos métodos: bien la CA publica directamente su certificado autofirmado; o bien se dispone de esta información a través de directorio (o equivalente) y la CA publica un código de troceo de este valor para permitir la verificación de su integridad antes de utilizarlo.

```
OOBCert ::= Certificate
```

Los campos dentro de este certificado están restringidos del siguiente modo:

- El certificado deberá estar autofirmado (es decir, la firma debe ser verificable utilizando el campo SubjectPublicKeyInfo).
- Los campos de sujeto y de expedidor serán idénticos.
- Si el campo de sujeto es **NULL** las extensiones **subjectAltNames** e **issuerAltNames** estarán presentes y tendrán exactamente el mismo valor.
- Los valores de las demás extensiones serán adecuados para un certificado autofirmado (por ejemplo, los identificadores de clave para el sujeto y el expedidor serán los mismos).

```
OOBCertHash ::= SEQUENCE {
    hashAlg     [0] AlgorithmIdentifier     OPTIONAL,
    certId      [1] CertId                  OPTIONAL,
    hashVal     BIT STRING
    -- hashVal is calculated over the self-signed
    -- certificate with the identifier certID.
}
```

La intención del código de troceo es que cualquiera que haya recibido de manera segura el código de troceo (vía los mecanismos "fuera de banda") pueda verificar un certificado autofirmado para la CA.

8.2.6 Publicación de información

Los peticionarios pueden indicar su deseo de que la PKI publique un certificado utilizando la estructura **PKIPublicationInfo**. Para la sintaxis de **PKIPublicationInfo** véase 8.3.

8.2.7 Estructuras de prueba de posesión

La prueba de posesión de la clave de firma privada se confirma mediante el uso de la estructura **POPOSigningKey**. La sintaxis de **POPOSigningKey** se encuentra en 8.3, pero se ha de señalar que **POPOSigningKeyInput** tiene las siguientes estipulaciones semánticas en esta especificación:

```
POPOSigningKeyInput ::= SEQUENCE {
        authInfo        CHOICE {
                sender              [0] GeneralName,
                -- from PKIHeader (used only if an authenticated identity
                -- has been established for the sender (e.g. a DN from a
                -- previously-issued and currently-valid certificate))
                publicKeyMAC    [1] PKMACValue
                -- used if no authenticated GeneralName currently exists for
                -- the sender; publicKeyMAC contains a password-based MAC
                -- (using the protectionAlg AlgorithmIdentifier from PKIHeader) on the
                -- DER-encoded value of publicKey
                },
        publicKey           SubjectPublicKeyInfo-- from CertTemplate
        }
```

8.3 Estructuras de datos específicas de los mensajes de petición de certificado del tipo CertReq

En esta subcláusula se describe el formato del mensaje petición de certificado (CRMF, *certificate request message format*). Esta sintaxis se utiliza para cursar la petición de un certificado a una autoridad de certificación (CA) [posiblemente a través de una autoridad de registro (RA)] a los efectos de producción de certificados de conformidad con X.509. La petición incluirá por lo general una clave pública y la información de registro asociada.

8.3.1 Visión de conjunto

La construcción de una petición de certificado comprende los siguientes pasos:

a) Se construye un valor de CertRequest. Este valor puede incluir la clave pública, el nombre completo de la entidad terminal (EE, *end-entity*) o una porción del mismo, otros campos de certificado requeridos e información de control adicional relativa al proceso de registro.

b) Puede calcularse un valor de prueba de posesión (de la clave privada que corresponde a la clave pública para la cual se pide el certificado) a través del valor de CerrtRequest.

c) Puede combinarse la información de registro adicional con el valor de prueba de posesión y la estructura de CertRequest para formar un CertReqMessage.

d) El CertReqMessage se comunica de manera segura a una CA (como se especifica en 8.1.3).

8.3.2 Sintaxis de CertReqMessage

El cuerpo del mensaje de petición de certificado se compone de la petición de certificado, un campo de prueba de posesión opcional y un campo de información de registro opcional.

```
CertReqMessages ::= SEQUENCE SIZE (1..MAX) OF CertReqMsg

CertReqMsg ::= SEQUENCE {
   certReq  CertRequest,
   pop      ProofOfPossession  OPTIONAL,
   regInfo  SEQUENCE SIZE(1..MAX) OF AttributeTypeAndValue OPTIONAL }
```

El campo prueba de posesión se utiliza para demostrar que la entidad asociada con el certificado se encuentra realmente en posesión de la clave privada correspondiente. Este campo puede calcularse a través del contenido del campo **certReq** y varía en estructura y contenido según el tipo de algoritmo de claves públicas y el modo de servicio.

El campo **regInfo** sólo debe contener información suplementaria relativa al contexto de la petición de certificación cuando tal información es requerida para cumplimentar una petición de certificación. Esta información puede incluir información de contacto del abonado, información de facturación u otra información auxiliar útil para la cumplimentación de la petición de certificación.

La información relacionada directamente con el contenido del certificado debe ser incluida en el contenido de **certReq**. Sin embargo, la inclusión de contenido de **certReq** adicional por parte de las RA puede invalidar el campo prueba de posesión. Por consiguiente, los datos destinados al contenido del certificado pueden ser proporcionados en **regInfo**.

8.3.3 Prueba de posesión (POP, *proof of possession*)

Para evitar ciertas agresiones y permitir que una CA/RA compruebe adecuadamente la validez de la vinculación entre una entidad terminal y un par de claves, los servicios de gestión PKI aquí especificados posibilitan que una entidad terminal pruebe que está en posesión de (es decir, que puede utilizar) la clave privada correspondiente a la clave pública para la cual se solicita un certificado. Una CA/RA determinada puede elegir libremente el modo de imponer la POP (por ejemplo, mecanismos procedimentales fuera de banda versus el mensaje CRMF dentro de banda) en sus intercambios de certificación (lo que quiere decir que puede constituir un tema político). En cualquier caso, las CA/RA impondrán la POP por algún medio.

Esta Recomendación | Norma Internacional tiene en cuenta los casos en los cuales la POP es validada por la CA, la RA, o por ambas. Algunas políticas pueden requerir que la CA verifique la POP durante la certificación, en cuyo caso la RA deberá reenviar los campos CertRequest y ProofOfPossession de la entidad terminal inalterados a la CA, pudiendo también, opcionalmente, verificar la POP. Si la política no requiere que la CA verifique la POP, entonces la RA debe reenviar la petición y la prueba de la entidad terminal inalteradas a la CA como se ha indicado anteriormente. Si esto no es posible (debido, por ejemplo, a que la RA verifica la POP por un método fuera de banda), la RA puede dar testimonio a la CA de que la prueba solicitada ha sido validada. Si la CA utiliza un método fuera de banda para verificar la POP (tal como la entrega física de claves privadas generadas por la CA), el campo ProofOfPossession no se utiliza.

8.3.3.1 Sintaxis de la prueba de posesión

```
ProofOfPossession ::= CHOICE {
    raVerified      [0] NULL,
    -- used if the RA has already verified that the requester is in
    -- possession of the private key
    signature       [1] POPOSigningKey,
}

POPOSigningKey ::= SEQUENCE {
    poposkInput             [0] POPOSigningKeyInput OPTIONAL,
    algorithmIdentifier     AlgorithmIdentifier,
    signature               BIT STRING }
    -- The signature (using "algorithmIdentifier") is on the
    -- DER-encoded value of poposkInput. NOTE –  If the CertReqMsg
    -- certReq CertTemplate contains the subject and publicKey values,
    -- then poposkInput shall be omitted and the signature shall be
    -- computed on the DER-encoded value of CertReqMsg certReq.  If
    -- the CertReqMsg certReq CertTemplate does not contain the public
    -- key and subject values, then poposkInput shall be present and
    -- shall be signed.  This strategy ensures that the public key is
    -- not present in both the poposkInput and CertReqMsg certReq CertTemplate fields.

POPOSigningKeyInput ::= SEQUENCE {
    authInfo        CHOICE {
        sender      [0] GeneralName,
        -- used only if an authenticated identity has been
        -- established for the sender (e.g. a DN from a
        -- previously-issued and currently-valid certificate)
        publicKeyMAC    PKMACValue },
        -- used if no authenticated GeneralName currently exists for
        -- the sender; publicKeyMAC contains a password-based MAC
        -- on the DER-encoded value of publicKey
    publicKey       SubjectPublicKeyInfo }  -- from CertTemplate

PKMACValue ::= SEQUENCE {
    algId  AlgorithmIdentifier,
    -- the algorithm value shall be PasswordBasedMac
    -- {1 2 840 113533 7 66 13}
    -- the parameter value is PBMParameter
    value  BIT STRING }
```

8.3.3.2 Utilización de MAC basado en contraseñas

Cuando se utiliza **publicKeyMAC** en **POPOSigningKeyInput** para probar la autenticidad de una petición, deberá aplicarse el algoritmo siguiente.

```
PBMParameter ::= SEQUENCE {
    salt            OCTET STRING,
    owf             AlgorithmIdentifier,
    -- AlgorithmIdentifier for a One-Way Function (SHA-1 recommended)
    iterationCount  INTEGER,
    -- number of times the OWF is applied
    mac             AlgorithmIdentifier
    -- the MAC AlgorithmIdentifier (e.g. DES-MAC, Triple-DES-MAC [PKCS#11],
}   -- or HMAC [RFC2104, RFC2202])
```

El proceso de utilización de **PBMParameter** para calcular **publicKeyMAC** y autenticar así el origen de una petición de certificación de clave pública consta de dos etapas. La primera etapa emplea información secreta compartida para producir una clave MAC. La segunda etapa somete a codificación MAC la clave pública en cuestión utilizando la clave MAC para producir un valor autenticado.

La inicialización de la primera etapa del algoritmo supone la existencia de una entrada secreta compartida distribuida según una relación de confianza entre la CA/RA y la entidad terminal. El valor salt se añade a la entrada secreta compartida y la función unidireccional (owf) se aplica iterationCount veces, donde secreto compartido es la entrada a la primera iteración y, para cada iteración sucesiva se fija la entrada de modo que sea la salida de la iteración anterior, produciendo una clave K.

En la etapa segunda, K y la clave pública son entradas a HMAC, tal como se documenta en RFC 2104, para producir un valor de **publicKeyMAC** del modo siguiente:

publicKeyMAC = Hash(K XOR opad, Hash(K XOR ipad, public key))

donde ipad y opad se definen en RFC 2104.

El AlgorithmIdentifier para owf será SHA-1 {1 3 14 3 2 26} y para mac será HMAC-SHA-1 {1 3 6 1 5 5 8 1 2}.

8.3.4 Sintaxis de CertRequest

La sintaxis de CertRequest se compone de un identificador de petición, una plantilla del contenido del certificado y una secuencia opcional de información de control.

```
CertRequest ::= SEQUENCE {
        certReqId     INTEGER,              -- ID for matching request and response
        certTemplate  CertTemplate,         -- Selected fields of certificate to be issued
        controls      Controls OPTIONAL }   -- Attributes affecting issuance

CertTemplate ::= SEQUENCE {
    version          [0] Version               OPTIONAL,
    serialNumber     [1] INTEGER               OPTIONAL,
    signingAlg       [2] AlgorithmIdentifier   OPTIONAL,
    issuer           [3] Name                  OPTIONAL,
    validity         [4] OptionalValidity      OPTIONAL,
    subject          [5] Name                  OPTIONAL,
    publicKey        [6] SubjectPublicKeyInfo  OPTIONAL,
    issuerUID        [7] UniqueIdentifier      OPTIONAL,
    subjectUID       [8] UniqueIdentifier      OPTIONAL,
    xtensions        [9] Extensions            OPTIONAL }

OptionalValidity ::= SEQUENCE {
    notBefore   [0] Time OPTIONAL,
    notAfter    [1] Time OPTIONAL } --at least one shall be present

Time ::= CHOICE {
    utcTime     TCTime,
    generalTime eneralizedTime }
```

8.3.5 Sintaxis de control

El generador de una CertRequest puede incluir uno o más valores de control pertenecientes al procesamiento de la petición.

```
Controls ::= SEQUENCE SIZE(1..MAX) OF AttributeTypeAndValue
```

Se definen los siguientes controles (se reconoce que esta lista puede ampliarse con el tiempo): **regToken**; **authenticator**; **pkiPublicationInfo**; **oldCertID**; **protocolEncrKey**.

8.3.5.1 Control del testigo de registro

Un control de **regToken** contiene información "antigua" (basada en un valor secreto o en uno conocido) destinada a ser utilizada por la CA para verificar la identidad del sujeto antes de la expedición de un certificado. Tras la recepción de una petición de certificación que contenga un valor de **regToken**, la CA receptora verifica la información para confirmar la identidad pretendida en la petición de certificación.

El valor de **regToken** puede ser generado por la CA y proporcionado fuera de banda al abonado, o puede por lo demás estar disponible tanto en la CA como en el abonado. La seguridad de cualquier intercambio fuera de banda debe ser proporcionada con el riesgo de que la CA acepte un valor interceptado de alguien distinto al abonado deseado.

El control de **regToken** por lo general sólo se utilizaría para la inicialización de una entidad terminal en la PKI, mientras que el control del autenticador (véase la subcláusula siguiente) se utilizaría normalmente para las peticiones de certificación inicial y siguientes.

En algunos casos de uso el valor de **regToken** puede ser una cadena de texto o una cantidad numérica, como un número aleatorio. El valor en este último caso puede en principio representarse, bien como una cantidad binaria o bien como una cadena de texto. Para garantizar una codificación uniforme de los valores con independencia de la naturaleza de la cantidad, es necesario que la codificación de **regToken** sea UTF8String en cualquier caso.

8.3.5.2 Control del autenticador

Un control del autenticador contiene información utilizada sobre una base progresiva para establecer una comprobación no criptográfica de identidad en comunicación con la CA. Ejemplos: nombre de soltera de la madre, cuatro últimos dígitos del número de la seguridad social, u otra información conocida compartida con la CA del abonado; un código de troceo de dicha información; u otra información producida con esta finalidad. El valor de un control del autenticador puede ser generado por el abonado o por la CA.

En algunos casos de uso el valor de **authenticator** puede ser una cadena de texto o una cantidad numérica, como un número aleatorio. El valor en este último caso puede en principio representarse, bien como una cantidad binaria o bien como una cadena de texto. Para garantizar una codificación uniforme de los valores con independencia de la naturaleza de la cantidad, es necesario que la codificación de **authenticator** sea UTF8String en cualquier caso.

8.3.5.3 Control de la información de publicación

El control de **pkiPublicationInfo** permite a los abonados controlar la publicación del certificado de la CA. Viene definido por la sintaxis siguiente:

```
PKIPublicationInfo ::= SEQUENCE {
   action     INTEGER {
              dontPublish (0),
              pleasePublish (1) },
   pubInfos  SEQUENCE SIZE (1..MAX) OF SinglePubInfo OPTIONAL }
    -- pubInfos SHALL not be present if action is "dontPublish"
    -- (if action is "pleasePublish" and pubInfos is omitted,
    -- "dontCare" is assumed)

SinglePubInfo ::= SEQUENCE {
   pubMethod    INTEGER {
      dontCare       (0),
      x500           (1),
      web            (2),
      ldap           (3) },
   pubLocation  GeneralName OPTIONAL }
```

Si se elige la opción **dontPublish**, el peticionario indica que la PKI no debe publicar el certificado (éste puede indicar que el peticionario intenta publicar el certificado por sí mismo/sí misma).

Si se elige el método **dontCare**, o si el control de **PKIPublicationInfo** es omitido desde la petición, el peticionario indica que la PKI puede publicar el certificado utilizando cualquier medio que elija.

Si el peticionario desea que el certificado aparezca en algunos emplazamientos al menos, pero desea posibilitar que la CA ponga el certificado a disposición en otros depósitos, se fijan dos valores de **SinglePubInfo** para **pubInfos**: uno con valor **x500**, **web** o **ldap** y otro con **dontCare**.

El campo **pubLocation**, si se suministra, indica la ubicación que el peticionario desearía para el certificado (obsérvese que la CHOICE dentro de **GeneralName** incluye un URL y una dirección IP, por ejemplo).

8.3.5.4 Control del ID de OldCert

Si está presente, el control de **OldCertID** especifica el certificado que ha de ser actualizado por la petición de certificación actual. La sintaxis de su valor es:

```
CertId ::= SEQUENCE {
    issuer          GeneralName,
    serialNumber    INTEGER
}
```

8.3.5.5 Control de clave de criptación de protocolo

Si está presente, el control de **protocolEncrKey** especifica una clave que utilizará la CA en la criptación de una respuesta a **CertReqMessages**. Este control puede aplicarse cuando una CA ha de enviar información al abonado que ha de criptarse. Tal información incluye una clave privada generada por la CA para uso del abonado.

La codificación de **protocolEncrKey** será **SubjectPublicKeyInfo**.

8.3.6 Identificadores de objeto

El OID **id-pkix** tiene el valor

```
id-pkix  OBJECT IDENTIFIER  ::= { iso(1) identified-organization(3)
dod(6) internet(1) security(5) mechanisms(5) pkix(7) }

-- arc for Internet X.509 PKI protocols and their components
id-pkip  OBJECT IDENTIFIER :: { id-pkix pkip(5) }

-- Registration Controls in CRMF
id-regCtrl  OBJECT IDENTIFIER ::= { id-pkip regCtrl(1) }
id-regCtrl-regToken                 OBJECT IDENTIFIER ::= { id-regCtrl 1 }
id-regCtrl-authenticator            OBJECT IDENTIFIER ::= { id-regCtrl 2 }
id-regCtrl-pkiPublicationInfo       OBJECT IDENTIFIER ::= { id-regCtrl 3 }
id-regCtrl-oldCertID                OBJECT IDENTIFIER ::= { id-regCtrl 5 }
id-regCtrl-protocolEncrKey          OBJECT IDENTIFIER ::= { id-regCtrl 6 }

-- Registration Info in CRMF
id-regInfo      OBJECT IDENTIFIER ::= { id-pkip id-regInfo(2) }
id-regInfo-asciiPairs   OBJECT IDENTIFIER ::= { id-regInfo 1 }
-- with syntax OCTET STRING
id-regInfo-certReq      OBJECT IDENTIFIER ::= { id-regInfo 2 }
-- with syntax CertRequest
```

8.4 Estructuras de datos específicas de otros mensajes

8.4.1 Petición de inicialización

Un mensaje de petición de inicialización contiene como el **PKIBody** una estructura de datos **CertReqMessages** que especifica los certificados solicitados. Generalmente, **SubjectPublicKeyInfo**, **KeyId** y **Validity** son los campos de plantilla que pueden ser suministrados para cada certificado solicitado. Este mensaje se destina a la inicialización primera de entidades en la PKI.

En 8.3 puede verse la sintaxis de **CertReqMessages**.

8.4.2 Respuesta de inicialización

Un mensaje de respuesta de inicialización contiene como el **PKIBody** una estructura de datos **CertRepMessage** que tiene para cada certificado solicitado un campo **PKIStatusInfo**, un certificado de sujeto y posiblemente una clave privada (normalmente criptada con una clave de sesión, que está ella misma criptada con la **protocolEncKey**).

En la subcláusula 8.4.4 puede verse la sintaxis de **CertRepMessage**.

8.4.3 Petición de registro/certificación

Un mensaje de petición de registro/certificación contiene como el **PKIBody** una estructura de datos **CertReqMessages** que especifica los certificados solicitados. Este mensaje se destina para uso de las entidades PKI existentes que desean obtener certificados adicionales.

En 8.3 puede verse la sintaxis de **CertReqMessages**.

Alternativamente, el **PKIBody** puede ser una **CertificationRequest** (esta estructura es especificada enteramente por la estructura ASN.1 **CertificationRequest** dada en PKCS#10). Esta estructura puede ser requerida para las peticiones de certificado de pares de claves de firma cuando se desea intercambiar servicios con sistemas de herencia, pero se desaconseja decididamente su utilización cuando no sea absolutamente necesaria.

8.4.4 Respuesta de registro/certificación

Un mensaje de respuesta de registro contiene como el **PKIBody** una estructura de datos **CertRepMessage** que tiene un valor de estado para cada certificado solicitado, y opcionalmente una clave pública de CA, información de fallo, una certificado de sujeto y una clave privada criptada.

```
CertRepMessage ::= SEQUENCE {
   caPubs        [1] SEQUENCE SIZE (1..MAX) OF Certificate OPTIONAL,
   response          SEQUENCE OF CertResponse
}

CertResponse ::= SEQUENCE {
   certReqId        INTEGER,
   -- to match this response with corresponding request (a value
   -- of –1 is to be used if certReqId is not specified in the
   -- corresponding request)
   status           PKIStatusInfo,
   certifiedKeyPair    CertifiedKeyPair   OPTIONAL,
   rspInfo             OCTET STRING     OPTIONAL
   -- analogous to the id-regInfo-asciiPairs OCTET STRING defined
   -- for regInfo in CertReqMsg
}

CertifiedKeyPair ::= SEQUENCE {
   certOrEncCert      CertOrEncCert,
   privateKey      [0] EncryptedValue    OPTIONAL,
publicationInfo [1] PKIPublicationInfo    OPTIONAL
}

CertOrEncCert ::= CHOICE {
   certificate       [0] Certificate,
   encryptedCert       [1] EncryptedValue
}
```

Solamente uno de los campos **failInfo** (en **PKIStatusInfo**) y **certificate** (en **CertifiedKeyPair**) puede estar presente en cada **CertResponse** (dependiendo del estado). Para algunos valores de estado (por ejemplo, en espera) no estará presente ninguno de los campos opcionales.

Dado un **EncryptedCert** y la clave de decriptación pertinente puede obtenerse el certificado. La finalidad es permitir que una CA devuelva el valor de un certificado, pero con la constricción de que solamente el receptor deseado pueda obtener el certificado real. La ventaja de este enfoque es que una CA puede reactuar con un certificado incluso en ausencia de una prueba de que el peticionario es la entidad terminal que puede utilizar la clave privada pertinente (se señala que la prueba no se obtiene hasta que el mensaje **PKIConfirm** es recibido por la CA). De este modo la CA no tendrá que revocar este certificado en el caso de que se encuentre algún error en la prueba de posesión.

8.4.5 Contenido de la petición de actualización de claves

Para las peticiones de actualización de claves se utiliza la sintaxis de **CertReqMessages**. Típicamente, **SubjectPublicKeyInfo**, **KeyId** y **Validity** son los campos de plantilla que pueden ser suministrados para cada clave que ha de actualizarse. Este mensaje está destinado a la petición de actualizaciones de certificados existentes (no revocados y no caducados).

En 8.3 se presenta la sintaxis de **CertReqMessages**.

8.4.6 Contenido de la respuesta de actualización de claves

Para las respuestas de actualización de claves se utiliza la sintaxis de **CertRepMessage**. La respuesta es idéntica a la respuesta de inicialización.

En 8.4.4 puede verse la sintaxis de **CertRepMessage**.

8.4.7 Contenido de la petición de revocación

Cuando se pide la revocación de un certificado (o de varios) se utiliza la siguiente estructura de datos. El nombre del peticionario está presente en la estructura **PKIHeader**.

```
RevReqContent ::= SEQUENCE OF RevDetails

RevDetails ::= SEQUENCE {
   certDetails      CertTemplate,
   -- allows requester to specify as much as they can about
   -- the cert. for which revocation is requested
   -- (e.g. for cases in which serialNumber is not available)
   revocationReason  ReasonFlags      OPTIONAL,
   -- the reason that revocation is requested
   badSinceDate      GeneralizedTime      OPTIONAL,
   -- indicates best knowledge of sender
   crlEntryDetails   Extensions       OPTIONAL
   -- requested crlEntryExtensions
}
```

8.4.8 Contenido de la respuesta de revocación

Es la respuesta al mensaje anterior. Si se produce, es enviada al peticionario de la revocación. (Puede enviarse un mensaje de notificación de revocación separado al sujeto del certificado para el cual se ha solicitado la revocación.)

```
RevRepContent ::= SEQUENCE {
   status      SEQUENCE SIZE (1..MAX) OF PKIStatusInfo,
   -- in same order as was sent in RevReqContent
   revCerts  [0] SEQUENCE SIZE (1..MAX) OF CertId OPTIONAL,
   -- IDs for which revocation was requested (same order as status)
   crls    [1] SEQUENCE SIZE (1..MAX) OF CertificateList  OPTIONAL
   -- the resulting CRLs (there may be more than one)
}
```

8.4.9 Contenido de la petición de certificación cruzada

Las peticiones de certificación cruzada utilizan la misma sintaxis (**CertReqMessages**) que las peticiones de certificación normales con la restricción de que el par de claves será generado por la CA peticionaria y la clave privada no se enviará a la CA respondedora.

En 8.3 se puede ver la sintaxis de **CertReqMessages**.

8.4.10 Contenido de la respuesta de certificación cruzada

Las respuestas de certificación cruzada utilizan la misma sintaxis (**CertRepMessage**) que las respuestas de certificación normales con la restricción de que no puede enviarse ninguna la clave privada.

En 8.4.4 se recoge la sintaxis de **CertRepMessage**.

8.4.11 Contenido de la notificación de actualización de claves

Cuando una CA actualiza su propio par de claves se puede aplicar la siguiente estructura de datos para notificar este evento.

```
CAKeyUpdAnnContent ::= SEQUENCE {
    oldWithNew     Certificate, -- old pub signed with new priv
    newWithOld     Certificate, -- new pub signed with old priv
    newWithNew     Certificate  -- new pub signed with new priv
}
```

8.4.12 Notificación de certificado

Esta estructura puede utilizarse para notificar la existencia de certificados.

Obsérvese que este mensaje se destina para su uso en aquellos casos (si procede) en que no existe un método preexistente para la publicación de certificados; no se aplicará cuando, por ejemplo, el método de publicación de certificados sea X.500.

```
CertAnnContent ::= Certificate
```

8.4.13 Notificación de revocación

Cuando una CA ha revocado, o está a punto de hacerlo, un certificado concreto puede expedir una notificación de este evento (posiblemente próximo).

```
RevAnnContent ::= SEQUENCE {
    status          PKIStatus,
    certId          CertId,
    willBeRevokedAt    GeneralizedTime,
    badSinceDate       GeneralizedTime,
    crlDetails         Extensions  OPTIONAL
    -- extra CRL details(e.g. CRL number, reason, location, etc.)
}
```

Una CA puede utilizar tal notificación para avisar (o notificar) a un sujeto de que su certificado está a punto de ser revocado (o lo ha sido ya). Se aplicaría generalmente cuando la petición de revocación no ha provenido del sujeto concernido.

El campo **willBeRevokedAt** contiene la hora en que se añadirá una nueva inserción a las CRL correspondientes.

8.4.14 Notificación de CRL

Cuando una CA emite una nueva CRL (o un conjunto de CRL) se puede utilizar la siguiente estructura de datos para notificar el evento.

```
CRLAnnContent ::= SEQUENCE OF CertificateList
```

8.4.15 Contenido de la confirmación de PKI

Esta estructura de datos se aplica en protocolos tridireccionales como el **PKIMessage** final. Su contenido es el mismo en todos los casos. Actualmente no hay contenido puesto que el **PKIHeader** transporta toda la información requerida.

```
CRLAnnContent ::= SEQUENCE OF CertificateList
```

8.4.16 Contenido del mensaje general PKI

```
InfoTypeAndValue ::= SEQUENCE {
    infoType          TYPE-IDENTIFIER.&id({InfoTable}),
    infoValue         TYPE-IDENTIFIER.&Type ({InfoTable}{@infoType}) OPTIONAL
}

    -- Example InfoTypeAndValue contents include, but are not limited to:
    --  { CAProtEncCert    = {id-it 1}, Certificate                    }
    --  { SignKeyPairTypes = {id-it 2}, SEQUENCE OF AlgorithmIdentifier }
    --  { EncKeyPairTypes  = {id-it 3}, SEQUENCE OF AlgorithmIdentifier }
    --  { PreferredSymmAlg = {id-it 4}, AlgorithmIdentifier         }
    --  { CAKeyUpdateInfo  = {id-it 5}, CAKeyUpdAnnContent          }
    --  { CurrentCRL       = {id-it 6}, CertificateList             }
    -- where {id-it} = {id-pkix 4} = {1 3 6 1 5 5 7 4}
    -- This construct may also be used to define new PKIX Certificate
    -- Management Protocol request and response messages, or general-
    -- purpose (e.g. announcement) messages for future needs or for
    -- specific environments.
```

```
GenMsgContent ::= SEQUENCE OF InfoTypeAndValue
   -- May be sent by EE, RA, or CA (depending on message content).
   -- The OPTIONAL infoValue parameter of InfoTypeAndValue will typically
   -- be omitted for some of the examples given above.  The receiver is
   -- free to ignore any contained OBJECT IDs that it does not recognize.
   -- If sent from EE to CA, the empty set indicates that the CA may send
   -- any/all information that it wishes.
```

8.4.17 Contenido de la respuesta general PKI

```
GenRepContent ::= SEQUENCE OF InfoTypeAndValue
-- The receiver is free to ignore any contained OBJECT IDs that it does
-- not recognize.
```

8.4.18 Contenido del mensaje de error

```
ErrorMsgContent ::= SEQUENCE {
   pKIStatusInfo        PKIStatusInfo,
   errorCode            INTEGER       OPTIONAL,
   -- implementation-specific error codes
   errorDetails         PKIFreeText   OPTIONAL
   -- implementation-specific error details
}
```

8.5 Protocolos de transporte

No está ordenado ningún protocolo de transporte específico para la transferencia de mensajes PKI entre entidades terminales, RA y CA. No es necesario aplicar mecanismos de seguridad específicos en este nivel si los mensajes PKI son protegidos adecuadamente (esto es, si se utiliza el parámetro PKIProtection opcional especificado para cada mensaje).

Los mensajes PKI pueden transportarse en ficheros que contengan solamente la codificación DER de un mensaje PKI, es decir, sin información de encabezamiento o terminación extraña en el fichero. Estos ficheros pueden utilizarse para transportar mensajes PKI utilizando, por ejemplo, FTP. Estos ficheros pueden también adjuntarse a correos electrónicos o transferirse vía HTTP (pueden definirse para este fin objetos MIME especiales).

8.6 Módulo ASN.1 completo

8.6.1 Módulo específico para el formato de mensaje de petición de certificación (CRMF, *certification request message format*)

```
CRMF DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- EXPORTS ALL; --
IMPORTS
-- Directory Information Framework (X.501)
        Name
          FROM InformationFramework {
            joint-iso-itu-t(2) ds(5) module(1) informationFramework(1) 3 }

    -- Directory Authentication Framework (X.509)
        AlgorithmIdentifier, Extensions, SubjectPublicKeyInfo, Time,
          Version
          FROM AuthenticationFramework {
            joint-iso-itu-t(2) ds(5) module(1) authenticationFramework(7) 3 }

    -- Directory Selected Attributes (X.520)
        UniqueIdentifier
          FROM SelectedAttributeTypes {
            joint-iso-itu-t(2) ds(5) module(1) selectedAttributeTypes(5) 3 }

    -- Certificate Extensions (X.509)
        GeneralName
          FROM CertificateExtensions {joint-iso-itu-t(2) ds(5)
            module(1) certificateExtensions(26) 0}
```

```
    -- Cryptographic Message Syntax
      EnvelopedData
        FROM CryptographicMessageSyntax { iso(1) member-body(2)
          us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16)
           modules(0) cms(1) };

CertReqMessages ::= SEQUENCE SIZE (1..MAX) OF CertReqMsg

CertReqMsg ::= SEQUENCE {
  certReq   CertRequest,
  pop       ProofOfPossession  OPTIONAL,
  -- content depends upon key type
  regInfo   SEQUENCE SIZE(1..MAX) OF AttributeTypeAndValue OPTIONAL }

CertRequest ::= SEQUENCE {
  certReqId    INTEGER,                 -- ID for matching request and response
certTemplate  CertTemplate,             -- Selected fields of certificate to be issued
  controls     Controls OPTIONAL }      -- Attributes affecting issuance

CertTemplate ::= SEQUENCE {
  version          [0] Version                 OPTIONAL,
  serialNumber     [1] INTEGER                 OPTIONAL,
  signingAlg       [2] AlgorithmIdentifier     OPTIONAL,
  issuer           [3] EXPLICIT Name           OPTIONAL,
  validity         [4] OptionalValidity        OPTIONAL,
  subject          [5] EXPLICIT Name           OPTIONAL,
  publicKey        [6] SubjectPublicKeyInfo    OPTIONAL,
  issuerUID        [7] UniqueIdentifier        OPTIONAL,
  subjectUID       [8] UniqueIdentifier        OPTIONAL,
  extensions       [9] Extensions              OPTIONAL }

OptionalValidity ::= SEQUENCE {
  notBefore        [0] EXPLICIT Time OPTIONAL,
  notAfter         [1] EXPLICIT Time OPTIONAL } --at least one SHALL be present

Controls  ::= SEQUENCE SIZE(1..MAX) OF AttributeTypeAndValue

AttributeTypeAndValue ::= SEQUENCE {
  type             TYPE-IDENTIFIER.&id ({CRMF-Table}),
  value            TYPE-IDENTIFIER.&Type ({CRMF-Table}{@type})
}

CRMF-Table TYPE-IDENTIFIER::={ ... }

ProofOfPossession ::= CHOICE {
  raVerified       [0] NULL,
  -- used if the RA has already verified that the requester is in
  -- possession of the private key
  signature        [1] POPOSigningKey }

POPOSigningKey ::= SEQUENCE {
  poposkInput          [0] POPOSigningKeyInput OPTIONAL,
  algorithmIdentifier   AlgorithmIdentifier,
  signature             BIT STRING }
  -- The signature (using "algorithmIdentifier") is on the
  -- DER-encoded value of poposkInput.  NOTE – If the CertReqMsg
  -- certReq CertTemplate contains the subject and publicKey values,
  -- then poposkInput SHALL be omitted and the signature SHALL be
  -- computed on the DER-encoded value of CertReqMsg certReq.  If
  -- the CertReqMsg certReq CertTemplate does not contain the public
  -- key and subject values, then poposkInput SHALL be present and
  -- SHALL be signed.  This strategy ensures that the public key is
  -- not present in both the poposkInput and CertReqMsg certReq
  -- CertTemplate fields.

POPOSigningKeyInput ::= SEQUENCE {
  authInfo         CHOICE {
     sender        [0] EXPLICIT GeneralName,
     -- used only if an authenticated identity has been
     -- established for the sender (e.g. a DN from a
     -- previously-issued and currently-valid certificate
```

```
      publicKeyMAC     PKMACValue },
      -- used if no authenticated GeneralName currently exists for
      -- the sender; publicKeyMAC contains a password-based MAC
      -- on the DER-encoded value of publicKey
   publicKey        SubjectPublicKeyInfo }  -- from CertTemplate

PKMACValue ::= SEQUENCE {
   algId  AlgorithmIdentifier,
   -- algorithm value shall be PasswordBasedMac {1 2 840 113533 7 66 13}
   -- parameter value is PBMParameter
   value  BIT STRING }

PBMParameter ::= SEQUENCE {
     salt            OCTET STRING,
     owf             AlgorithmIdentifier,
     -- AlgorithmIdentifier for a One-Way Function (SHA-1 recommended)
     iterationCount     INTEGER,
     -- number of times the OWF is applied
     mac             AlgorithmIdentifier
     -- the MAC AlgorithmIdentifier (e.g. DES-MAC, Triple-DES-MAC as in PKCS #11,
}   -- or HMAC as in RFC2104, RFC2202)

-- Object identifier assignments --

id-pkix  OBJECT IDENTIFIER  ::= { iso(1) identified-organization(3)
dod(6) internet(1) security(5) mechanisms(5) 7 }

-- arc for Internet X.509 PKI protocols and their components
id-pkip  OBJECT IDENTIFIER ::= { id-pkix 5 }

-- Registration Controls in CRMF
id-regCtrl OBJECT IDENTIFIER ::= { id-pkip 1 }

id-regCtrl-regToken OBJECT IDENTIFIER ::= { id-regCtrl 1 }
--with syntax:
RegToken ::= UTF8String

id-regCtrl-authenticator OBJECT IDENTIFIER ::= { id-regCtrl 2 }
--with syntax:
Authenticator ::= UTF8String

id-regCtrl-pkiPublicationInfo OBJECT IDENTIFIER ::= { id-regCtrl 3 }
--with syntax:
PKIPublicationInfo ::= SEQUENCE {
   action     INTEGER {
              dontPublish (0),
              pleasePublish (1) },
   pubInfos  SEQUENCE SIZE (1..MAX) OF SinglePubInfo OPTIONAL }
   -- pubInfos SHALL not be present if action is "dontPublish"
   -- (if action is "pleasePublish" and pubInfos is omitted,
   -- "dontCare" is assumed)

SinglePubInfo ::= SEQUENCE {
   pubMethod    INTEGER {
      dontCare    (0),
      x500        (1),
      web         (2),
      ldap        (3) },
   pubLocation  GeneralName OPTIONAL }

EncryptedKey ::= CHOICE {
   encryptedValue       EncryptedValue,
   envelopedData     [0] EnvelopedData }
   -- The encrypted private key SHALL be placed in the envelopedData
   -- encryptedContentInfo encryptedContent OCTET STRING.

EncryptedValue ::= SEQUENCE {
     intendedAlg   [0] AlgorithmIdentifier   OPTIONAL,
     symmAlg       [1] AlgorithmIdentifier   OPTIONAL,
     encSymmKey    [2] BIT STRING            OPTIONAL,
```

```
    keyAlg          [3] AlgorithmIdentifier    OPTIONAL,
    valueHint       [4] OCTET STRING           OPTIONAL,
    encValue            BIT STRING
  }

KeyGenParameters ::= OCTET STRING

id-regCtrl-oldCertID        OBJECT IDENTIFIER ::= { id-regCtrl 5 }
--with syntax:
OldCertId ::= CertId

CertId ::= SEQUENCE {
  issuer          GeneralName,
  serialNumber    INTEGER }

id-regCtrl-protocolEncrKey    OBJECT IDENTIFIER ::= { id-regCtrl 6 }
--with syntax:
ProtocolEncrKey ::= SubjectPublicKeyInfo

-- Registration Info in CRMF
id-regInfo OBJECT IDENTIFIER ::= { id-pkip 2 }

id-regInfo-utf8Pairs    OBJECT IDENTIFIER ::= { id-regInfo 1 }
--with syntax
UTF8Pairs ::= UTF8String

id-regInfo-certReq      OBJECT IDENTIFIER ::= { id-regInfo 2 }
--with syntax
CertReq ::= CertRequest

END
```

8.6.2 Módulo general

```
-- Note that additional syntax appears in the CRMF module above.
GeneralModule DEFINITIONS EXPLICIT TAGS ::=

  BEGIN

  -- EXPORTS ALL --

  IMPORTS

--  InformationFramework (X.501) --

  Attribute, Name
    FROM InformationFramework {
      joint-iso-itu-t ds(5) module(1) informationFramework(1) 3 }

-- Directory Authentication Framework (X.509) --

  AlgorithmIdentifier, Certificate, CertificateList, Extensions,
  SubjectPublicKeyInfo
    FROM AuthenticationFramework {
      joint-iso-itu-t ds(5) module(1) authenticationFramework(7) 3 }

-- Certificate Extensions (X.509)

  GeneralName, KeyIdentifier, ReasonFlags
    FROM CertificateExtensions {
      joint-iso-itu-t(2) ds(5) module(1) certificateExtensions(26) 0 }

-- X.843 ISO 15945 (CRMF) --

  CertTemplate, PKIPublicationInfo, EncryptedValue, CertId,
    CertReqMessages
        FROM CRMF;
```

```
-- CertificationRequest compatible to PKCS#10

  CertificationRequest ::= SEQUENCE {
    certificationRequestInfo  CertificationRequestInfo,
    signatureAlgorithm        AlgorithmIdentifier,
    signature                 BIT STRING
  }

  CertificationRequestInfo ::= SEQUENCE {
    version       INTEGER,
    subject       Name,
    subjectPKInfo  SubjectPublicKeyInfo,
    attributes    [0] IMPLICIT Attributes
  }

  Attributes ::= SET SIZE(0..MAX) OF Attribute

                    --  Locally defined OIDs  ---- Note that tagging is EXPLICIT in this module.

  PKIMessage ::= SEQUENCE {
    header           PKIHeader,
    body             PKIBody,
    protection       [0] PKIProtection OPTIONAL,
    extraCerts       [1] SEQUENCE SIZE (1..MAX) OF Certificate OPTIONAL
  }

  PKIHeader ::= SEQUENCE {
    pvno             INTEGER   { version1 (0) },
    sender           GeneralName,
    -- identifies the sender
    recipient        GeneralName,
    -- identifies the intended recipient
    messageTime   [0] GeneralizedTime      OPTIONAL,
    -- time of production of this message (used when sender
    -- believes that the transport will be "suitable"; i.e.
    -- that the time will still be meaningful upon receipt)
    protectionAlg  [1] AlgorithmIdentifier   OPTIONAL,
    -- algorithm used for calculation of protection bits
    senderKID      [2] KeyIdentifier         OPTIONAL,
    recipKID       [3] KeyIdentifier         OPTIONAL,
    -- to identify specific keys used for protection
    transactionID  [4] OCTET STRING      OPTIONAL,
    -- identifies the transaction; i.e. this will be the same in
    -- corresponding request, response and confirmation messages
    senderNonce    [5] OCTET STRING      OPTIONAL,
    recipNonce     [6] OCTET STRING      OPTIONAL,
    -- nonces used to provide replay protection, senderNonce
    -- is inserted by the creator of this message; recipNonce
    -- is a nonce previously inserted in a related message by
    -- the intended recipient of this message
    freeText       [7] PKIFreeText           OPTIONAL,
    -- this may be used to indicate context-specific instructions
    -- (this field is intended for human consumption)
    generalInfo    [8] SEQUENCE SIZE (1..MAX) OF
                       InfoTypeAndValue      OPTIONAL
    -- this may be used to convey context-specific information
    -- (this field not primarily intended for human consumption)
  }

  PKIFreeText ::= SEQUENCE SIZE (1..MAX) OF UTF8String
    -- text encoded as UTF-8 String (NOTE – each UTF8String should
    -- include an RFC 1766 language tag to indicate the language
    -- of the contained text)

PKIBody ::= CHOICE {                    -- message-specific body elements
    ir       [0]  CertReqMessages,      --Initialization Request
    ip       [1]  CertRepMessage,       --Initialization Response
    cr       [2]  CertReqMessages,      --Certification Request
    cp       [3]  CertRepMessage,       --Certification Response
    p10cr    [4]  CertificationRequest, --for compatibility with [PKCS#10]
```

```
  kur       [7]  CertReqMessages,          --Key Update Request
  kup       [8]  CertRepMessage,           --Key Update Response
  rr        [11] RevReqContent,            --Revocation Requestrp
  rp        [12] RevRepContent,            --Revocation Response
  ccr       [13] CertReqMessages,          --Cross-Cert. Request
  ccp       [14] CertRepMessage,           --Cross-Cert. Response
  ckuann    [15] CAKeyUpdAnnContent,       --CA Key Update Ann.
  cann      [16] CertAnnContent,           --Certificate Ann.
  rann      [17] RevAnnContent,            --Revocation Ann.
  crlann    [18] CRLAnnContent,            --CRL Announcement
  conf      [19] PKIConfirmContent,        --Confirmation nested
  nested    [20] NestedMessageContent,     --Nested Message
  genm      [21] GenMsgContent,            --General Message
  genp      [22] GenRepContent,            --General Response
  error     [23] ErrorMsgContent           --Error Message
}

PKIProtection ::= BIT STRING

ProtectedPart ::= SEQUENCE {
  header    PKIHeader,
  body      PKIBody
}

PasswordBasedMac ::= OBJECT IDENTIFIER --{1 2 840 113533 7 66 13}

PBMParameter ::= SEQUENCE {
  salt              OCTET STRING,
  owf               AlgorithmIdentifier,
  -- AlgorithmIdentifier for a One-Way Function (SHA-1 recommended)
  iterationCount    INTEGER,
  -- number of times the OWF is applied
  mac               AlgorithmIdentifier
  -- the MAC AlgorithmIdentifier (e.g. DES-MAC, Triple-DES-MAC as in PKCS #11,
} -- or HMAC as in RFC2104, RFC2202)

DHBasedMac ::= OBJECT IDENTIFIER --{1 2 840 113533 7 66 30}

DHBMParameter ::= SEQUENCE {
  owf               AlgorithmIdentifier,
  -- AlgorithmIdentifier for a One-Way Function (SHA-1 recommended)
  mac               AlgorithmIdentifier
  -- the MAC AlgorithmIdentifier (e.g, DES-MAC, Triple-DES-MAC as in PKCS #11,
} -- or HMAC RFC2104, RFC2202)

NestedMessageContent ::= PKIMessage

PKIStatus ::= INTEGER {
  granted                   (0),
  -- you got exactly what you asked for
  grantedWithMods           (1),
  -- you got something like what you asked for; the
  -- requester is responsible for ascertaining the differences
  rejection                 (2),
  -- you don't get it, more information elsewhere in the message
  waiting                   (3),
  -- the request body part has not yet been processed,
  -- expect to hear more later
  revocationWarning         (4),
  -- this message contains a warning that a revocation is
  -- imminent
  revocationNotification    (5),
  -- notification that a revocation has occurred
  keyUpdateWarning          (6)
  -- update already done for the oldCertId specified in
  -- CertReqMsg
}
```

```
PKIFailureInfo ::= BIT STRING {
  -- since a request can fail in more than one way!
  -- More codes may be added in the future if/when required.
  badAlg                (0),
  -- unrecognized or unsupported Algorithm Identifier
  badMessageCheck       (1),
  -- integrity check failed (e.g. signature did not verify)
  badRequest            (2),
  -- transaction not permitted or supported
  badTime               (3),
  -- messageTime was not sufficiently close to the system time,
  -- as defined by local policy
  badCertId             (4),
  -- no certificate could be found matching the provided criteria
  badDataFormat         (5),
  -- the data submitted has the wrong format
  wrongAuthority        (6),
  -- the authority indicated in the request is different from the
  -- one creating the response token
  incorrectData         (7),
  -- the requester's data is incorrect (for notary services)
  missingTimeStamp      (8)
  -- when the timestamp is missing but should be there (by policy)
}

PKIStatusInfo ::= SEQUENCE {
  status            PKIStatus,
  statusString      PKIFreeText     OPTIONAL,
  failInfo          PKIFailureInfo  OPTIONAL
}

OOBCert ::= Certificate

OOBCertHash ::= SEQUENCE {
  hashAlg       [0] AlgorithmIdentifier    OPTIONAL,
  certId        [1] CertId                 OPTIONAL,
  hashVal           BIT STRING
-- hashVal is calculated over DER encoding of the
-- subjectPublicKey field of the corresponding cert.
}

CertRepMessage ::= SEQUENCE {
  caPubs        [1] SEQUENCE SIZE (1..MAX) OF Certificate OPTIONAL,
  response          SEQUENCE OF CertResponse
}

CertResponse ::= SEQUENCE {
  certReqId         INTEGER,
  -- to match this response with corresponding request (a value
  -- of -1 is to be used if certReqId is not specified in the
  -- corresponding request)
  status                PKIStatusInfo,
  certifiedKeyPair      CertifiedKeyPair   OPTIONAL,
  rspInfo               OCTET STRING  OPTIONAL
  -- analogous to the id-regInfo-asciiPairs OCTET STRING defined
  -- for regInfo in CertReqMsg
}

CertifiedKeyPair ::= SEQUENCE {
  certOrEncCert       CertOrEncCert,
  privateKey        [0] EncryptedValue       OPTIONAL,
  publicationInfo   [1] PKIPublicationInfo   OPTIONAL
}

CertOrEncCert ::= CHOICE {
  certificate       [0] Certificate,
  encryptedCert     [1] EncryptedValue
}
```

```
KeyRecRepContent ::= SEQUENCE {
  status              PKIStatusInfo,
  newSigCert    [0]   Certificate             OPTIONAL,
  caCerts       [1]   SEQUENCE SIZE (1..MAX) OF
                      Certificate             OPTIONAL,
  keyPairHist   [2]   SEQUENCE SIZE (1..MAX) OF
                      CertifiedKeyPair        OPTIONAL
}

RevReqContent ::= SEQUENCE OF RevDetails

RevDetails ::= SEQUENCE {
  certDetails         CertTemplate,
  -- allows requester to specify as much as they can about
  -- the cert. for which revocation is requested
  -- (e.g. for cases in which serialNumber is not available)
  revocationReason    ReasonFlags         OPTIONAL,
  -- the reason that revocation is requested
  badSinceDate        GeneralizedTime     OPTIONAL,
  -- indicates best knowledge of sender
  crlEntryDetails     Extensions          OPTIONAL
  -- requested crlEntryExtensions
}

RevRepContent ::= SEQUENCE {
  status       SEQUENCE SIZE (1..MAX) OF PKIStatusInfo,
  -- in same order as was sent in RevReqContent
  revCerts   [0] SEQUENCE SIZE (1..MAX) OF CertId OPTIONAL,
  -- IDs for which revocation was requested (same order as status)
  crls       [1] SEQUENCE SIZE (1..MAX) OF CertificateList  OPTIONAL
  -- the resulting CRLs (there may be more than one)
}

CAKeyUpdAnnContent ::= SEQUENCE {
  oldWithNew          Certificate, -- old pub signed with new priv
  newWithOld          Certificate, -- new pub signed with old priv
  newWithNew          Certificate  -- new pub signed with new priv
}

CertAnnContent ::= Certificate

RevAnnContent ::= SEQUENCE {
  status              PKIStatus,
  certId              CertId,
  willBeRevokedAt     GeneralizedTime,
  badSinceDate        GeneralizedTime,
  crlDetails          Extensions  OPTIONAL
  -- extra CRL details(e.g. CRL number, reason, location, etc.)
}

CRLAnnContent ::= SEQUENCE OF CertificateList

PKIConfirmContent ::= NULL

InfoTypeAndValue ::= SEQUENCE {
  infoType            TYPE-IDENTIFIER.&id ({InfoTable}),
  infoValue           TYPE-IDENTIFIER.&Type ({InfoTable}@infoType}) OPTIONAL
}

InfoTable TYPE-IDENTIFIER ::= { ... }

-- Example InfoTypeAndValue contents include, but are not limited to:
--  { CAProtEncCert    = {id-it 1}, Certificate                 }
--  { SignKeyPairTypes = {id-it 2}, SEQUENCE OF AlgorithmIdentifier }
--  { EncKeyPairTypes  = {id-it 3}, SEQUENCE OF AlgorithmIdentifier }
--  { PreferredSymmAlg = {id-it 4}, AlgorithmIdentifier          }
--  { CAKeyUpdateInfo  = {id-it 5}, CAKeyUpdAnnContent           }
--  { CurrentCRL       = {id-it 6}, CertificateList              }
-- where {id-it} = {id-pkix 4} = {1 3 6 1 5 5 7 4}
```

```
  -- This construct may also be used to define new PKIX Certificate
  -- Management Protocol request and response messages, or general-
  -- purpose (e.g. announcement) messages for future needs or for
  -- specific environments.

  GenMsgContent ::= SEQUENCE OF InfoTypeAndValue
  -- May be sent by EE, RA, or CA (depending on message content).
  -- The OPTIONAL infoValue parameter of InfoTypeAndValue will typically
  -- be omitted for some of the examples given above.  The receiver is
  -- free to ignore any contained OBJECT IDs that it does not recognize.
  -- If sent from EE to CA, the empty set indicates that the CA may send
  -- any/all information that it wishes.

  GenRepContent ::= SEQUENCE OF InfoTypeAndValue
  -- The receiver is free to ignore any contained OBJECT IDs that it does
  -- not recognize.

  ErrorMsgContent ::= SEQUENCE {
    pKIStatusInfo          PKIStatusInfo,
    errorCode              INTEGER           OPTIONAL,
    -- implementation-specific error codes
    errorDetails           PKIFreeText       OPTIONAL
    -- implementation-specific error details
  }

END
```

9 Protocolo de estado del certificado en línea (OCSP)

Esta cláusula especifica un protocolo de utilidad en la determinación del estado actual de un certificado sin necesidad de recurrir a las CRL. Las estructuras de datos deberán especificarse de conformidad con el documento RFC 2560. En 9.1 se da una visión de conjunto del protocolo. En 9.2 se especifican los requisitos funcionales. En 9.3 se recogen los detalles del protocolo. La subcláusula 9.4 contiene el módulo ASN.1 para las estructuras de datos necesarias para el protocolo.

NOTA – El código ASN.1 es equivalente al del documento RFC arriba mencionado, si bien la sintaxis parece diferente en algunas partes.

9.1 Visión de conjunto del protocolo

El protocolo de estado del certificado en línea (OCSP, *online certificate status protocol*) permite aplicaciones para determinar el estado de una certificado identificado. El OCSP se puede utilizar para satisfacer alguno de los requisitos operacionales de la provisión de información de revocación más oportuna en el tiempo que si hace uso de las CRL, y puede también aplicarse para obtener información de estado adicional. Una entidad terminal que utilice servicios OCSP emite una petición de estado a una TTP OSCP y suspende la aceptación del certificado en cuestión hasta que la TTP proporcione una respuesta.

Este protocolo especifica los datos que se necesita intercambiar entre una entidad terminal que comprueba el estado de una certificado y la TTP que proporciona dicho estado.

9.1.1 Petición

Una petición OCSP contiene los siguientes datos:

- versión de protocolo;
- petición de servicio;
- identificador del certificado deseado;
- extensiones opcionales que pueden ser procesadas por la TTP OCSP.

Tras la recepción de una petición, una TTP OCSP determina si:

1) el mensaje está bien formado;
2) la propia TTP está configurada para proporcionar el servicio pedido; y
3) la petición contiene la información que necesita la TTP.

Si no se cumple una de las condiciones anteriores, la TTP OCSP produce un mensaje de error; en caso contrario devuelve una respuesta definitiva.

9.1.2 Respuesta

Las respuestas OCSP pueden ser de varios tipos. Una respuesta OCSP se compone de un tipo de respuesta y los bytes de la respuesta real. Hay un tipo básico de respuesta OCSP que deberán soportar todas las TTP proveedoras y todas las entidades usuarias de servicios OCSP. El resto de esta subcláusula trata solamente de este tipo de respuesta básica.

Todos los mensajes de la respuesta definitiva deberán firmarse digitalmente. La clave utilizada para firmar la respuesta deberá pertenecer a una de las siguientes entidades:

– la CA que ha expedido el certificado en cuestión;

– una TTP cuya clave pública sea aceptada como fiable por el peticionario;

– una TTP designada por la CA (TTP autorizada), la cual mantiene retenido un certificado especial expedido por la CA indicando que puede emitir respuestas OCSP para la CA.

Un mensaje de respuesta definitiva se compone de:

- versión de la sintaxis de la respuesta;
- nombre de la TTP;
- respuestas para cada uno de los certificados de una petición;
- extensiones opcionales;
- OID del algoritmo de firma;
- firma computada mediante troceo de la respuesta.

La respuesta de cada uno de los certificados de una petición consta de:

- identificador del certificado deseado;
- valor de estado del certificado;
- intervalo de validez de la respuesta;
- extensiones opcionales.

Esta Recomendación | Norma Internacional define los siguientes indicadores de respuesta definitiva para su utilización en el valor de estado del certificado:

- **good (bueno)**;
- **revoked (revocado)**;
- **unknown (desconocido)**.

El estado "**good**" indica una respuesta positiva a la petición de estado. Como mínimo, esta respuesta positiva indica que el certificado no ha sido revocado, pero no significa necesariamente que el certificado haya sido emitido nunca o que el momento en que se produjo la respuesta se encuentre dentro del intervalo de validez del certificado. Pueden utilizarse las extensiones de la respuesta para cursar información adicional sobre aserciones hechas por la TTP acerca del estado del certificado, tal como una declaración positiva acerca de la expedición, validez, etc.

El estado "**revoked**" indica que el certificado ha sido revocado (sea permanente o temporalmente (en retención)).

El estado "**unknown**" indica que la TTP no tiene conocimiento del certificado solicitado.

9.1.3 Excepciones

En casos de error, la TTP OCSP puede devolver un mensaje de error. Estos mensajes no están firmados. Los errores pueden ser de los tipos siguientes:

- **malformedRequest**;
- **internalError**;
- **tryLater**;
- **sigRequired**;
- **unauthorized**.

Una TTP produce la respuesta "**malformedRequest**" si la petición recibida no es conforme a la sintaxis del OCSP.

La respuesta "**internalError**" indica que la TTP OCSP alcanzó una estado interno inconsistente. Se repetiría la petición, posiblemente con otra TTP.

Cuando la TTP OCSP se encuentra en funcionamiento, pero no está capacitada para devolver un estado del certificado pedido, se puede utilizar la respuesta "**tryLater**" para indicar que el servicio existe, pero es temporalmente incapaz de responder.

La respuesta "**sigRequired**" se devuelve en los casos en que la TTP requiere que la entidad terminal firme la petición para construir una respuesta.

La respuesta "**unauthorized**" es devuelta en los casos en que la entidad terminal no esta autorizada para formular esta petición a esta TTP.

9.1.4 Semántica de thisUpdate, nextUpdate y producedAt

Las respuestas puede contener tres tiempos: **thisUpdate**, **nextUpdate** y **producedAt**. Las semánticas de estos campos son:

- **thisUpdate**: El tiempo en el cual el estado indicado es conocido como correcto;
- **nextUpdate**: El tiempo en el cual, o antes del cual, se dispondrá de información más reciente acerca del estado del certificado;
- **producedAt**: El tiempo en el cual la TTP OCSP firmó esta respuesta.

Si **nextUpdate** no está fijado, la TTP indica que la información de revocación más reciente se encuentra disponible todo el tiempo.

9.1.5 Preproducción de la respuesta

Las TTP OCSP pueden producir respuestas firmadas especificando el estado de certificados en un momento especificado. El momento en el que se conoció que el estado era correcto debe quedar reflejado en el campo **thisUpdate** de la repuesta. El tiempo en el cual, o anterior al cual, se dispondrá de información más reciente se refleja en el campo **nextUpdate**, mientras que el tiempo en el que se produjo la respuesta aparecerá en el campo **producedAt** de la respuesta.

9.1.6 Delegación de la autoridad de firma OCSP

La clave que firma una información de estado de certificado ha de ser la misma clave que firmó el certificado. El expedidor de un certificado delega explícitamente la autoridad de firma OCSP mediante la expedición de un certificado que contiene un valor único de **extendedKeyUsage** en el certificado del firmante OCSP.

9.1.7 Clave de CA comprometida

Si una TTP OCSP sabe que una clave privada de CA concreta ha sido comprometida, puede devolver el estado revocado para todos los certificados expedidos por la CA.

9.2 Requisitos funcionales

9.2.1 Contenido del certificado

A fin de transportar a las entidades terminales un punto de acceso de información perfectamente conocido, las CA deberán proporcionar la capacidad de incluir la extensión **AuthorityInfoAccess** (definida en RFC 2459, sección 4.2.2.1) en los certificados que puedan ser comprobados mediante OCSP. Alternativamente, la **accessLocation** para el proveedor OCSP puede ser configurada localmente en el equipo de las entidades terminales.

Las CA que soporten un servicio OCSP, bien hospedado localmente o bien prestado por una TTP autorizada, pueden proporcionar un valor para una **accessLocation** del **uniformResourceIndicator** (URI) y el valor de OID **id-ad-ocsp** para el **accessMethod** en la **AccessDescription SEQUENCE**.

El valor del campo **accessLocation** en el certificado del sujeto define el mecanismo de transporte (por ejemplo, HTTP) utilizado para acceder a la TTP OCSP, y puede contener otra información dependiente del transporte (por ejemplo, un URL).

9.2.2 Requisitos de aceptación de una respuesta firmada

Antes de aceptar como válida un respuesta firmada, las entidades terminales deberá confirmar que:

1) El certificado identificado en la respuesta recibida corresponde al identificado en la petición correspondiente.
2) La firma de la respuesta es válida.

3) La identidad del firmante concuerda con el receptor deseado de la petición.;

4) El firmante está autorizado actualmente para firmar la respuesta.

5) El momento en el cual se conoce que el estado indicado es el correcto (thisUpdate) es suficientemente reciente.

6) Cuando está disponible, el tiempo en el cual, o antes del cual, se dispondrá de información más reciente acerca del estado del certificado (nextUpdate) es superior al tiempo actual.

9.3 Protocolo detallado

La sintaxis ASN.1 importa términos definidos en RFC 2459. Para el cálculo de la firma, se codifican los datos que han de firmase utilizando las reglas de codificación distinguida (DER) ASN.1.

Se utiliza el rotulado **EXPLICIT** ASN.1 como valor por defecto, a menos que se especifique lo contrario.

Los términos importados de cualquier otro sitio: **Extensions, CertificateSerialNumber, SubjectPublicKeyInfo, Name, AlgorithmIdentifier, CRLReason.**

9.3.1 Peticiones

Esta subcláusula define la especificación ASN.1 para una petición de confirmación.

9.3.1.1 Sintaxis de la petición

```
OCSPRequest     ::=     SEQUENCE {
        tbsRequest              TBSRequest,
        optionalSignature   [0]  Signature OPTIONAL }

TBSRequest      ::=     SEQUENCE {
        version             [0]   Version DEFAULT v1,
        requestorName       [1]   GeneralName OPTIONAL,
        requestList                SEQUENCE OF Request,
        requestExtensions   [2]  Extensions OPTIONAL }

Signature       ::=SEQUENCE {
        signatureAlgorithm      AlgorithmIdentifier,
        signature               BIT STRING,
        certs               [0]  SEQUENCE OF Certificate
        OPTIONAL}

Version         ::=     INTEGER  {  v1(0) }

Request         ::=SEQUENCE {
        reqCert             CertID,
        singleRequestExtensions     [0]  Extensions OPTIONAL }

CertID          ::=SEQUENCE {
        hashAlgorithm           AlgorithmIdentifier,
        issuerNameHash          OCTET STRING, -- Hash of Issuer's DN
        issuerKeyHash           OCTET STRING, -- Hash of Issuer's public key
        serialNumber            CertificateSerialNumber }
```

issuerNameHash es el troceo del nombre distinguido del expedidor. El troceo deberá calcularse sobre la codificación DER del campo nombre del expedidor en el certificado que se está comprobando. **issuerKeyHash** es el troceo de la clave pública del expedidor. El troceo deberá calcularse sobre el valor (excluido el rótulo y la longitud) del campo clave pública del sujeto en el certificado del expedidor. El algoritmo de troceo utilizado para estos dos troceos se identifica como **hashAlgorithm**.

9.3.1.2 Observaciones sobre la sintaxis de la petición

El motivo principal de utilizar el nombre y la clave pública para identificar al expedidor es que puede ocurrir que dos CA tengan la posibilidad de seleccionar el uso del mismo nombre (la unicidad del nombre es un recomendación que no puede ser impuesta). Dos CA no tendrán nunca, sin embargo, la misma clave pública, a menos que las CA bien hayan decidido explícitamente compartir su clave privada, o bien la clave de una de las CA esté comprometida.

El soporte de cualquier extensión específica es opcional. La bandera crítica no debe fijarse para ninguna de ellas. En 9.3.4 se proponen varias extensiones útiles. Las extensiones no reconocidas deberán ignorarse (salvo que tengan la bandera crítica fijada y no sean entendidas).

El peticionario puede decidir si firma la petición OCSP. En tal caso, la firma es calculada sobre la estructura **tbsRequest**. Si la petición es firmada, el peticionario deberá especificar su nombre en el campo **requestorName**. Asimismo, para las peticiones firmadas, el peticionario puede incluir certificados que ayuden a la TTP OCSP a verificar la firma del peticionario en el campo certs de Signature.

9.3.2 Sintaxis de la respuesta

Esta subcláusula determina la especificación ASN.1 para una repuesta de confirmación.

9.3.2.1 Especificación ASN.1 de la respuesta OCSP

Una respuesta OCSP consta como mínimo de un campo **responseStatus** que indica el estado del procesamiento de la petición anterior. Si el valor de **responseStatus** es una de las condiciones de error, el valor de **responseBytes** no se fija.

```
OCSPResponse ::= SEQUENCE {
        responseStatus OCSPResponseStatus,
        responseBytes[0]  ResponseBytes OPTIONAL }

OCSPResponseStatus ::= ENUMERATED {
        successful          (0),  --Response has valid confirmations
        malformedRequest    (1),  --Illegal confirmation request
        internalError       (2),  --Internal error in issuer
        tryLater            (3),  --Try again later
        --(4) is not used
        sigRequired         (5),  --Must sign the request
        unauthorized        (6)  --Request unauthorized
}
```

El valor de responseBytes está formado por un **OBJECT IDENTIFIER** y una sintaxis de respuesta identificada por el OID codificado como una **OCTET STRING**:

```
ResponseBytes ::=SEQUENCE {
        responseType   OBJECT IDENTIFIER,
        response       OCTET STRING }
```

Para una TTP OCSP básica, **responseType** será **id-pkix-ocsp-basic**:

```
id-pkix-ocsp          OBJECT IDENTIFIER ::= { id-ad-ocsp }
id-pkix-ocsp-basic    OBJECT IDENTIFIER ::= { id-pkix-ocsp 1 }
```

Las TTP OCSP serán capaces de responder con respuestas del tipo **id-pkix-ocsp-basic**. En correspondencia, las entidades terminales deberán ser capaces de recibir y procesar respuestas del tipo **id-pkix-ocsp-basic**.

El valor de la respuesta será la codificación DER de **BasicOCSPResponse**:

```
BasicOCSPResponse ::= SEQUENCE {
        tbsResponseData       ResponseData,
        signatureAlgorithm    AlgorithmIdentifier,
        signature             BIT STRING,
        certs                 [0]  SEQUENCE OF Certificate OPTIONAL
}
```

El valor de signature deberá ser calculado sobre el troceo de la codificación DER de los **ResponseData**.

```
ResponseData ::= SEQUENCE {
        version             [0]  Version DEFAULT v1,
        responderID              ResponderID,
        producedAt               GeneralizedTime,
        responses                SEQUENCE OF SingleResponse,
        responseExtensions  [1]  Extensions OPTIONAL }

ResponderID ::= CHOICE {
        byName   [1] Name,
        byKey    [2] KeyHash }

KeyHash ::= OCTET STRING -- SHA-1 hash of TTP's public key (excluding the tag and length fields)
```

```
SingleResponse ::= SEQUENCE {
        certID CertID,
        certStatus CertStatus,
        thisUpdate GeneralizedTime,
        nextUpdate      [0] GeneralizedTime OPTIONAL,
        singleExtensions    [1] Extensions OPTIONAL }

CertStatus ::= CHOICE {
        good        [0]IMPLICIT NULL,
        revoked     [1]IMPLICIT RevokedInfo,
        unknown     [2]IMPLICIT UnknownInfo }

RevokedInfo ::= SEQUENCE {
        revocationTime GeneralizedTime,
        revocationReason[0] CRLReason OPTIONAL }

UnknownInfo ::= NULL
```

9.3.2.2 Observaciones sobre las respuestas OCSP

9.3.2.2.1 Tiempo

Los campos **thisUpdate** y **nextUpdate** definen un intervalo de validez recomendado. Este intervalo corresponde al intervalo {**thisUpdate**, **nextUpdate**} en las listas CRL. Las respuestas cuyo valor **nextUpdate** es anterior al tiempo del sistema local no deben considerarse fiables.

Las respuestas cuyo tiempo **thisUpdate** es posterior al tiempo del sistema local no deben considerarse fiables. Las respuestas en las que el valor de **nextUpdate** no está fijado son equivalentes a una CRL con ningún tiempo para **nextUpdate** (véase 9.1.4).

El tiempo **producedAt** es el tiempo en que ha sido firmada esta respuesta.

9.3.2.2.2 TTP autorizadas

La clave que firma una información de estado del certificado no debe ser la misma clave que firmó el certificado. Un expedidor de certificado puede delegar explícitamente la autoridad de la firma OCSP mediante la expedición de un certificado que incluya una extensión e**xtendedKeyUsage** en el certificado del firmante OCSP que contiene el valor **id-kp-OCSPSigning**.

```
id-kp-OCSPSigning OBJECT IDENTIFIER ::= {id-kp 9}
```

Puesto que una TTP OCSP autorizada proporciona información de estado para una CA, las entidades terminales necesitan conocer el modo de comprobar que un certificado de TTP autorizado no ha sido revocado. Las CA pueden elegir tratar este problema de uno de los tres modos siguientes:

- Una CA puede especificar que una entidad terminal sea capaz de otorgar confianza a una TTP durante el tiempo de vida del certificado de la TTP. La CA realiza esta función incluyendo la extensión **id-pkix-ocsp-nocheck**. Esta extensión debe ser una extensión no crítica. El valor de la extensión debe ser **NULL**. Las CA que expiden tal certificado deben comprobar de modo fehaciente que un compromiso de la clave de la TTP es tan serio como el compromiso de la clave de la CA, al menos para el periodo de validez de este certificado. Las CA pueden preferir expedir este tipo de certificado con un tiempo de vida corto y renovarlo con frecuencia.

  ```
  id-pkix-ocsp-nocheck OBJECT IDENTIFIER ::= { id-pkix-ocsp 5 }
  ```

- Una CA puede especificar el modo de comprobar la revocación del certificado de la TTP. Esto puede realizarse utilizando puntos de distribución CRL si la comprobación debe hacerse mediante listas CRL o puntos de distribución CRL, o utilizando acceso de información de autoridad si la comprobación debe realizarse por algún otro método.

- Una CA puede preferir no especificar ningún método de comprobación de revocación del certificado de la TTP, en cuyo caso se elevaría a la política de seguridad local de la entidad terminal la decisión sobre si dicho certificado debe ser o no comprobado en cuanto a su revocación.

9.3.3 Algoritmos criptográficos obligatorios y opcionales

Las entidades que soliciten servicios OCSP deberán ser capaces de procesar respuestas firmadas utilizando claves DSA identificadas por el **sig-alg-oid** DSA especificado en la sección 7.2.2 de RFC 2459. Las entidades terminales deben también ser capaces de procesar firmas RSA como se especifica en la sección 7.2.1 de RFC 2459. Las TTP OCSP soportarán el algoritmo de troceo SHA-1.

9.3.4 Extensiones

Esta subcláusula define algunas extensiones normalizadas. El soporte de todas las extensiones es facultativo. Para cada extensión, la definición indica su sintaxis, el procesamiento efectuado por la TTP OCSP y cualesquiera extensiones que estén incluidas en la respuesta correspondiente.

9.3.4.1 Nonce

El nonce vincula criptográficamente una petición y una respuesta para impedir la reproducción de agresiones. El nonce se incluye en las peticiones como una de las **requestExtensions**, mientras que en las respuestas se incluiría como una de las **responseExtensions**. En ambos casos, la petición y la respuesta, el nonce será identificado por el identificador de objeto **id-pkix-ocsp-nonce**, mientras que **extnValue** es el valor del nonce.

```
id-pkix-ocsp-nonce     OBJECT IDENTIFIER ::= { id-pkix-ocsp 2 }
```

9.3.4.2 Referencias de CRL

Puede ser de interés para la TTP OCSP indicar la CRL en la que se encuentra un certificado **onHold** o revocado. Esto puede ser de utilidad cuando se utiliza el OCSP entre depósitos, y también como un mecanismo de auditoría. La CRL puede especificarse por un URL (el URL en el cual la CRL está disponible), un número (número de la CRL) o un tiempo (el tiempo en que fue creada la CRL correspondiente). Estas extensiones se especificarán como **singleExtensions**. El identificador de esta extensión será **id-pkix-ocsp-crl**, mientras que el valor será **CrlID**.

```
id-pkix-ocsp-crl        OBJECT IDENTIFIER ::= { id-pkix-ocsp 3 }

CrlID ::= SEQUENCE {
        crlUrl      [0]   IA5String OPTIONAL,
        crlNum      [1]   INTEGER OPTIONAL,
        crlTime     [2]   GeneralizedTime OPTIONAL }
```

Para la elección de **crlUrl**, la **IA5String** especificará el URL en el cual está disponible la CRL. Para **crlNum**, el **INTEGER** especificará el valor de la extensión de número CRL de la CRL pertinente. Para **crlTime**, el **GeneralizedTime** indicará el tiempo en el cual se emitió la CRL pertinente.

9.3.4.3 Tipos de respuestas aceptables

Una entidad terminal puede desear especificar las clases de tipos de respuesta que entiende. Para llevar a cabo esta operación debe utilizar una extensión con el OID **id-pkix-ocsp-response**, y el valor **AcceptableResponses**. Los OID incluidos en **AcceptableResponses** son los OID de los distintos tipos de respuesta que esta entidad terminal puede aceptar (por ejemplo, i**d-pkix-ocsp-basic**).

```
id-pkix-ocsp-response  OBJECT IDENTIFIER ::= { id-pkix-ocsp 4 }

AcceptableResponses ::= SEQUENCE OF OBJECT IDENTIFIER
```

Como se ha señalado en 9.3.2.1, las TTP OCSP será capaces de responder con respuestas del tipo **id-pkix-ocsp-basic.** En correspondencia, las entidades terminales podrán recibir y procesar respuestas del tipo **id-pkix-ocsp-basic**.

9.3.4.4 Cierre de un archivo

Una TTP OCSP puede decidir retener información de revocación después de que caduque un certificado. La fecha obtenida restando el valor de este intervalo de retención del tiempo **producedAt** en una respuesta se define como la fecha de "cierre del archivo".

Las aplicaciones autorizadas OCSP utilizarían una fecha de cierre de archivo OCSP para contribuir a la prueba de que una firma digital era (o no) fiable en la fecha en que se produjo, incluso si el certificado que se ha necesitado para validar la firma ha caducado hace tiempo.

Las TTP OCSP que proporcionan soporte para tal referencia histórica deben incluir en las respuestas una extensión de fecha de cierre de archivo. Caso de estar incluida, deberá suministrarse como una extensión **singleResponse** OCSP identificada por **id-pkix-ocsp-archive-cutoff** y de sintaxis **GeneralizedTime**:

```
id-pkix-ocsp-archive-cutoff  OBJECT IDENTIFIER ::= { id-pkix-ocsp 6 }

ArchiveCutoff ::= GeneralizedTime
```

A título ilustrativo, si una TTP funciona con una política de intervalo de retención de 7 años y el estado se produjo en el tiempo t1, entonces el valor de **ArchiveCutoff** en la respuesta sería (t1 – 7 años).

9.3.4.5 Extensiones de inserción de CRL

Las extensiones de inserción de CRL son también soportadas como **singleExtensions**.

9.3.4.6 Localizador de servicio

Una TTP OCSP puede funcionar en un modo en el cual la TTP recibe un petición y la encamina a la TTP OCSP conocida como entidad autorizada del certificado identificado. La extensión de petición **serviceLocator** se define para esta finalidad.

```
id-pkix-ocsp-service-locator OBJECT IDENTIFIER ::= { id-pkix-ocsp 7 }

ServiceLocator ::= SEQUENCE {
        issuer    Name,
        locator   AuthorityInfoAccessSyntax OPTIONAL }
```

Los valores de estos campos se obtienen a partir de los campos correspondientes en el certificado del sujeto.

9.4 Módulo ASN.1 para OCSP

```
OCSP DEFINITIONS EXPLICIT TAGS ::=

BEGIN

IMPORTS
-- Directory Information Framework (X.501) --
  Name
    FROM InformationFramework {
      joint-iso-itu-t ds(5) module(1) informationFramework(1) 3 }
  -- Directory Authentication Framework (X.509) --
  AlgorithmIdentifier, Certificate, CertificateSerialNumber,
  Extensions
    FROM AuthenticationFramework {
      joint-iso-itu-t ds(5) module(1) authenticationFramework(7) 3 }
  -- Directory Certificate Extensions (X.509) --
  CRLReason, GeneralName
    FROM CertificateExtensions {joint-iso-itu-t ds(5)
      module(1) certificateExtensions(26) 0 }
  -- PKIX (RFC 2459) --
  AuthorityInfoAccessSyntax
    FROM PKIX1Implicit93 {
      iso(1) identified-organization(3) dod(6) internet(1)
        security(5) mechanisms(5) pkix(7) id-mod(0)
          id-pkix1-implicit-93(4) }
  id-kp, id-ad-ocsp
    FROM PKIX1Explicit93 {
      iso(1) identified-organization(3) dod(6) internet(1)
        security(5) mechanisms(5) pkix(7) id-mod(0)
          id-pkix1-explicit-93(3) };

OCSPRequest   ::=SEQUENCE {
            tbsRequest      TBSRequest,
            optionalSignature  [0] Signature OPTIONAL }

TBSRequest   ::=SEQUENCE {
            version                 [0] Version DEFAULT v1,
            requestorName           [1] GeneralName OPTIONAL,
            requestList                 SEQUENCE OF Request,
            requestExtensions       [2] Extensions OPTIONAL }
```

```
Signature           ::=SEQUENCE {
        signatureAlgorithm   AlgorithmIdentifier,
        signature BIT STRING,
        certs   [0] SEQUENCE OF Certificate OPTIONAL }

Version  ::=  INTEGER  {  v1(0) }

Request ::=    SEQUENCE {
        reqCert     CertID,
        singleRequestExtensions   [0] Extensions OPTIONAL }

CertID ::= SEQUENCE {
        hashAlgorithm      AlgorithmIdentifier,
        issuerNameHash OCTET STRING, -- Hash of Issuer's DN
        issuerKeyHash OCTET STRING, -- Hash of Issuer's public key
        serialNumber CertificateSerialNumber }

OCSPResponse ::= SEQUENCE {
        responseStatus      OCSPResponseStatus,
        responseBytes       [0] ResponseBytes OPTIONAL }

OCSPResponseStatus ::= ENUMERATED {
        successful(0), --Response has valid confirmations
        malformedRequest(1), --Illegal confirmation request
        internalError(2), --Internal error in issuer
        tryLater(3), --Try again later
        --(4) is not used
        sigRequired(5), --Must sign the request
        unauthorized(6) --Request unauthorized
}

ResponseBytes ::=SEQUENCE {
        responseType   OBJECT IDENTIFIER,
        response       OCTET STRING }

BasicOCSPResponse ::= SEQUENCE {
        tbsResponseData ResponseData,
        signatureAlgorithm AlgorithmIdentifier,
        signature BIT STRING,
        certs   [0] SEQUENCE OF Certificate OPTIONAL
}

ResponseData ::= SEQUENCE {
        version                 [0] Version DEFAULT v1,
        responderID                 ResponderID,
        producedAt                  GeneralizedTime,
        responses                   SEQUENCE OF SingleResponse,
        responseExtensions  [1] Extensions OPTIONAL }

ResponderID ::= CHOICE {
        byName   [1] Name,
        byKey    [2] KeyHash }

KeyHash ::= OCTET STRING --SHA-1 hash of TTP's public key
        --(excluding the tag and length fields)

SingleResponse ::= SEQUENCE {
        certID CertID,
        certStatus CertStatus,
        thisUpdate GeneralizedTime,
        nextUpdate[0] GeneralizedTime OPTIONAL,
        singleExtensions[1] Extensions OPTIONAL }

CertStatus ::= CHOICE {
        good        [0]     IMPLICIT NULL,
        revoked     [1]     IMPLICIT RevokedInfo,
        unknown     [2]     IMPLICIT UnknownInfo }

RevokedInfo ::= SEQUENCE {
        revocationTime   GeneralizedTime,
        revocationReason[0]         CRLReason OPTIONAL }

UnknownInfo ::= NULL
```

```
ArchiveCutoff ::= GeneralizedTime

AcceptableResponses ::= SEQUENCE OF OBJECT IDENTIFIER

ServiceLocator ::= SEQUENCE {
        issuer    Name,
        locator   AuthorityInfoAccessSyntax }

-- Object Identifiers
id-kp-OCSPSigning          OBJECT IDENTIFIER ::= { id-kp 9 }
id-pkix-ocsp               OBJECT IDENTIFIER ::= { id-ad-ocsp }
id-pkix-ocsp-basic         OBJECT IDENTIFIER ::= { id-pkix-ocsp 1 }
id-pkix-ocsp-nonce         OBJECT IDENTIFIER ::= { id-pkix-ocsp 2 }
id-pkix-ocsp-crl           OBJECT IDENTIFIER ::= { id-pkix-ocsp 3 }
id-pkix-ocsp-response      OBJECT IDENTIFIER ::= { id-pkix-ocsp 4 }
id-pkix-ocsp-nocheck       OBJECT IDENTIFIER ::= { id-pkix-ocsp 5 }
id-pkix-ocsp-archive-cutoff OBJECT IDENTIFIER ::= { id-pkix-ocsp 6 }
id-pkix-ocsp-service-locator OBJECT IDENTIFIER ::= { id-pkix-ocsp 7 }
END
```

Anexo A

Interfuncionamiento

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

La prestación de servicios de gestión de certificados puede requerir el interfuncionamiento de diferentes proveedores de servicios. Si se considera un conjunto de autoridades CA, cada una de las cuales presta su servicio dentro de su propio campo, las entidades deben tener claves públicas aseguradas de CA de otros dominios. Hay dos modelos básicos posibles: el primero es un modelo jerárquico que se basa en cadenas de certificados, y en el segundo cada una de las CA puede mantener certificaciones cruzadas con las demás. Pueden existir modelos híbridos entre estos dos.

En el primer modelo las autoridades están colocadas jerárquicamente por debajo de una CA "raíz" que expide certificados a las CA denominadas subordinadas. Estas CA pueden emitir certificados a otras CA situadas por debajo en esta jerarquía, o a entidades.

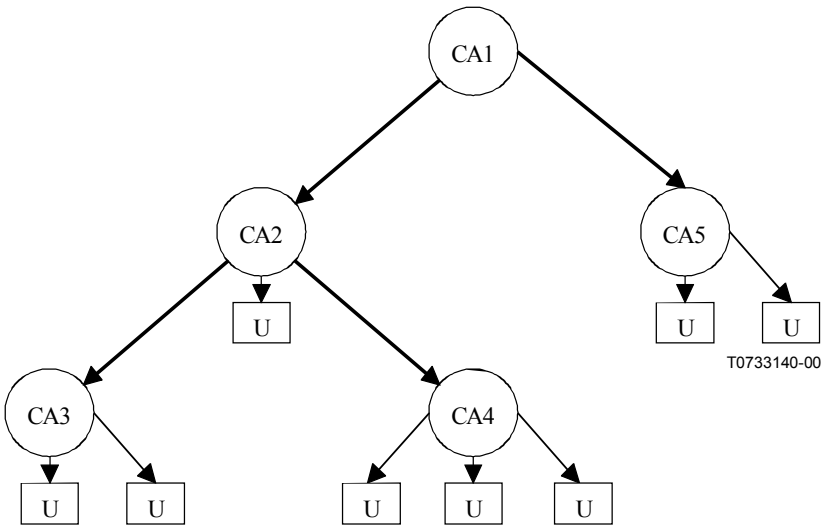

Figura A.1 – Jerarquía de las CA

En la figura A.1, CA1 se denomina CA "raíz". La función de la CA raíz es registrar sus CA subordinadas, que son CA2 y CA5 en la figura A.1. Las CA subordinadas pueden registrar más CA y/o entidades. Cada CA debe operar de acuerdo con una política común, de modo que se pueda alcanzar un nivel común de confianza o calidad de servicio.

En el segundo modelo, las CA independientes intercambian certificaciones cruzadas con las demás dando como resultado una red de CA interconectadas. La certificación recíproca es un arreglo bilateral entre dos CA, una de las cuales expide un certificado para la otra o ambas expiden un certificado para la otra.

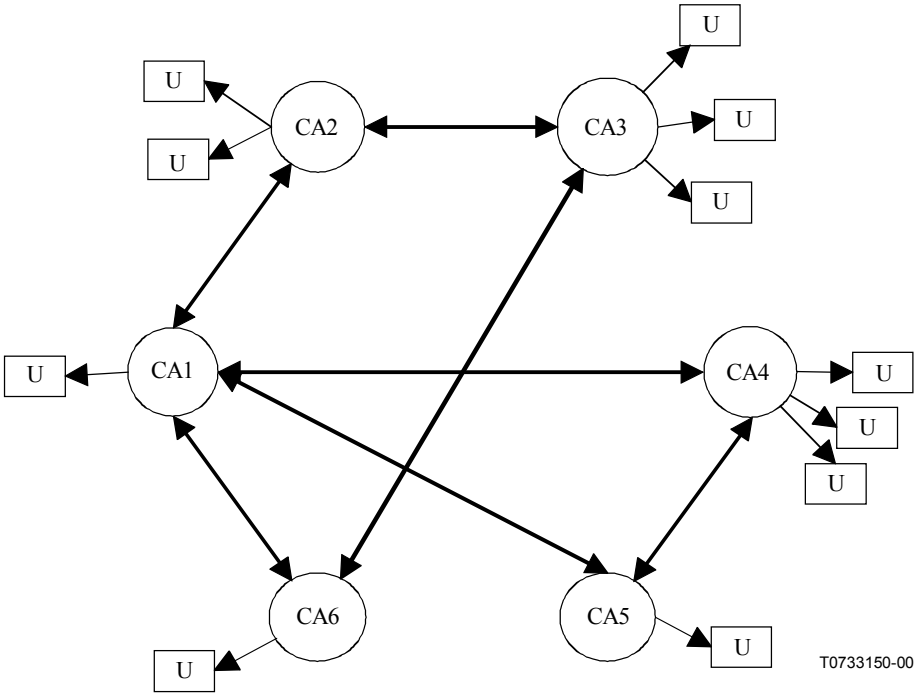

Figura A.2 – Ejemplo de red de las CA con certificación cruzada

Anexo B

Algoritmos

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

B.1 Algoritmos de troceo

En el contexto de las firmas digitales son adecuados los siguientes algoritmos de troceo:

a) RIPEMD-160:

DOBBERTIN, BOSSELAERS (A.), PRENEEL (B.): *RIPEMD-160: A strengthened version of RIPEMD, Fast Software Encryption,* Cambridge Workshop 1996, LNCS, Band 1039, S. 71-82, Springer-Verlag, 1996.

b) SHA-1:

NIST: FIPS Publication 180-1: *Secure Hash Standard (SHS-1),* mayo de 1995.

Ambos se describen en:

ISO/CEI 10118-3:1998, *Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions.*

B.2 Algoritmos de firmas digitales

En el contexto de esta Recomendación | Norma Internacional son adecuados los siguientes algoritmos de firmas digitales:

a) DSA

NIST: FIPS Publication 186: *Digital Signature Standard (DSS)*, mayo de 1994.

b) RSA

RIVEST, SHAMIR (A.), ADLEMAN (B): A method for obtaining digital signatures and public key cryptosystems, *Communications of the ACM*, Vol. 21, No. 2, 1978.

c) Esquemas similares a DSA basados en curvas elípticas

- ISO/CEI 14883-3, anexo A.2.1 ("Elliptic Curve DSA").
- ISO/CEI WD 15946-2.
- IEEE: Norma (proyecto) P1363, 6 de febrero de 1997, cláusula 5.3.3 ("Nyberg-Rueppel version").
- IEEE: Norma (proyecto) P1363, 6 de febrero de 1997, cláusula 5.3.4 ("DSA version").

Algunos algoritmos se describen en:

ISO/CEI 14888-3:1998, *Information technology – Security techniques – Digital signatures with appendix – Part 3: Certificate-based mechanisms.*

Anexo C

Bibliografía

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

COM(1997)503, *'Ensuring Security and Trust in Electronic Communication', Communication from the Commission to the European Parliament, the Council, the Economic and Social Committee and the Committee of the Regions,* octubre de 1997.

Directive 1999/93/EC of the European Parliament and of the Council on a Communinty framework for electronic signatures of 13 December 1999.

ECBS TR 402, European Committee for Banking Standards, Technical Report 402: Certification Authorities, Vol. 1, 1997.

ETSI EG/SEC-003000 Requirements for Trusted Third Party Services (Edition 1), Version 7.0, julio de 1997.

FIPS PUB 140-1, *Federal Information Processing Standard Publication 140-1, "Security Requirements for Cryptographic Modules", U.S. Department of commerce, National Institute of Standards and Technology,* enero de 1994.

Recomendación UIT-T X.511 (1997) | ISO/CEI 9594-3:1998, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Definición de servicio abstracto*.

Recomendación UIT-T X.519 (1997) | ISO/CEI 9594-5:1998, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Especificaciones de protocolo*.

Recomendación UIT-T X.810 (1995) | ISO/CEI 10181-1:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad para sistemas abiertos: Visión general.*

Recomendación UIT-T X.811 (1995) | ISO/CEI 10181-2:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad para sistemas abiertos: Marco de autenticación.*

Recomendación UIT-T X.812 (1995) | ISO/CEI 10181-3:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad para sistemas abiertos: Marco de control de acceso*.

Recomendación UIT-T X.813 (1996) | ISO/CEI 10181-4:1997, *Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad en sistemas abiertos: Marco de no rechazo.*

Recomendación UIT-T X.814 (1995) | ISO/CEI 10181-5:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad para sistemas abiertos: Marco de confidencialidad.*

Recomendación UIT-T X.815 (1995) | ISO/CEI 10181-6:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad para sistemas abiertos: Marco de integridad.*

Recomendación UIT-T X.816 (1995) | ISO/CEI 10181-7:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad para sistemas abiertos: Marco de auditoría y alarmas de seguridad.*

ISO/DIS 15782-1, *Certificate Management for financial services – Part 1: Public Key Certificates*.

ISO/CEI 10118-1:1994, *Information technology – Security techniques – Hash-functions – Part 1: General*.

ISO/CEI 10118-2:1994, *Information technology – Security techniques – Hash-functions – Part 2: Hash-functions using an n-bit block cipher algorithm*.

ISO/CEI 10118-3:1998, *Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions*.

ISO 9735:1998, *Electronic data interchange for administration, commerce and transport (EDIFACT) – Application level syntax rules*.

ISO/CEI 15408-1:1999, *Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model*.

ISO/CEI 15408-2:1999 *Information technology – Security techniques – Evaluation criteria for IT Security – Part 2: Security functional requirements*.

ISO/CEI 15408-3:1999, *Information technology – Security techniques – Evaluation criteria for IT Security – Part 3: Security assurance requirements.*

ITSEC, *Information Technology Security Evaluation Criteria (ITSEC), Harmonized Criteria of France, Germany, the Netherlands, the United Kingdom, Versión 1.2,* junio de 1992.

Minimum Interoperability Specification for PKI Components (MISPC), NIST Special Publication 800-15, septiembre de 1997.

PKCS Papers

PKCS #1, RSA Laboratories, PKCS #1: *RSA Encryption Standard,* Versión 1.5, noviembre de 1993.

PKCS #10, RSA Laboratories, PKCS #10: *Certification Request Standard*, Versión 1.5, noviembre de 1993.

PKCS #11, RSA Laboratories, PKCS #11: *Cryptographic Token Interface Standard*, Versión 1.0, abril de 1995.

PKCS #3, RSA Laboratories, PKCS #3: *Diffie-Hellman Key Agreement Standard,* Versión 1.4, noviembre de 1993.

PKCS #5, RSA Laboratories, PKCS #5: *Password-Based Encryption Standard*, Versión 1.5, noviembre de 1993.

PKCS #6, RSA *Laboratories*, PKCS #6: *Extended-Certificate Syntax Standard*, Versión 1.5, noviembre de 1993.

PKCS #7, RSA Laboratories, PKCS #7: *Cryptographic Message Standard*, Versión 1.5, noviembre de 1993, Extensions and revisions, 1997.

PKCS #8, RSA Laboratories, PKCS #8: *Private Key Information Syntax Standard*, Versión 1.5, noviembre de 1993.

PKCS #9, RSA Laboratories, PKCS #9: *Selected Attribute Types*, Versión 1.5, noviembre de 1993.

PKIX Papers

RFC 1421, *Privacy Enhancement for Electronic Mail: Part 1: Message Encryption and Authentication Procedures,* febrero de 1993.

RFC 1422, *Privacy Enhancement for Electronic Mail: Part 2: Certificate-Based Key Management,* febrero de 1993.

RFC 1423, *Privacy Enhancement for Electronic Mail: Part 3: Algorithms, Modes, and Identifiers,* febrero de 1993.

RFC 1424, *Privacy Enhancement for Electronic Mail: Part 4: Key Certification and Related Services,* febrero de 1993.

RFC 1510, *The Kerberos Network Authentication Services,* septiembre de 1993.

RFC 1750, *Randomness Recommendations for Security,* diciembre de 1994.

RFC 1766, *Tags for the Identification of Languages,* marzo de 1995.

RFC 2104, *HMAC: Keyed-Hashing for Message Authentication,* febrero de 1997.

RFC 2459, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile,* enero de 1999.

RFC 2510, *Internet X.509 Public Key Infrastructure Certificate Management Protocols,* marzo de 1999.

RFC 2511, *Internet X.509 Public Key Infrastructure Certificate Request Message Format,* marzo de 1999.

RFC 2527, *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework,* marzo de 1999.

RFC 2559, *Internet X.509 Public Key Infrastructure Operational Protocols – LDAPv2,* abril de 1999.

RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP,* junio de 1999.

Time Stamp Protocols, Internet Draft, septiembre de 1998 (work in progress); Adams C., Cain P., Pinkas D., Zuccherato R.

SET Secure Electronic Transaction Specification, Book 1: Business Description, Versión 1.0, 31 de mayo de 1997.

SET Secure Electronic Transaction Specification, Book 2: Programmer's Guide, Versión 1.0, 31 de mayo 1997.

SET Secure Electronic Transaction Specification, Book 3: Formal Protocol Definition, Versión 1.0, 31 de mayo de 1997.

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	**Redes de datos y comunicación entre sistemas abiertos**
Serie Y	Infraestructura mundial de la información y aspectos del protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación

Ginebra, 2002