

Reemplazada por una versión más reciente



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

X.509

(08/97)

SERIE X: REDES DE DATOS Y COMUNICACIÓN
ENTRE SISTEMAS ABIERTOS

Directorio

**Tecnología de la información – Interconexión
de sistemas abiertos – El directorio: Marco de
autenticación**

Recomendación UIT-T X.509

Reemplazada por una versión más reciente

(Anteriormente Recomendación del CCITT)

Reemplazada por una versión más reciente

RECOMENDACIONES DE LA SERIE X DEL UIT-T

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.399

SISTEMAS DE TRATAMIENTO DE MENSAJES

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.

Reemplazada por una versión más reciente

NORMA INTERNACIONAL 9594-8

RECOMENDACIÓN UIT-T X.509

TECNOLOGÍA DE LA INFORMACIÓN – INTERCONEXIÓN DE SISTEMAS ABIERTOS – EL DIRECTORIO: MARCO DE AUTENTICACIÓN

Resumen

La presente Recomendación | Norma Internacional define un marco para la prestación por el directorio de servicios de autenticación a sus usuarios. Describe dos niveles de autenticación: autenticación simple, mediante el uso de una contraseña como verificación de la identidad alegada, y autenticación fuerte, con credenciales formadas utilizando técnicas criptográficas. Si bien la autenticación simple ofrece cierta protección limitada contra el acceso no autorizado, sólo se debe utilizar la autenticación fuerte como base para proporcionar servicios seguros.

Orígenes

El texto de la Recomendación UIT-T X.509 se aprobó el 9 de agosto de 1997. Su texto se publica también, en forma idéntica, como Norma Internacional ISO/CEI 9594-8.

Reemplazada por una versión más reciente

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 Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), 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 N.º 1 de la CMNT.

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 *empresa de explotación reconocida (EER)* designa a toda persona, compañía, empresa u organización gubernamental que explote un servicio de correspondencia pública. Los términos *Administración, EER y correspondencia pública* están definidos en la *Constitución de la UIT (Ginebra, 1992)*.

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.

© UIT 1999

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

Reemplazada por una versión más reciente

ÍNDICE

	<i>Página</i>
SECCIÓN 1 – GENERALIDADES	1
1 Alcance	1
2 Referencias normativas.....	2
2.1 Recomendaciones Normas Internacionales idénticas	2
2.2 Pares de Recomendaciones Normas Internacionales de contenido técnico equivalente.....	3
2.3 Otras referencias.....	3
3 Definiciones.....	3
3.1 Definiciones relativas a la arquitectura de seguridad del modelo de referencia OSI	3
3.2 Definiciones relativas al modelo de directorio	4
3.3 Definiciones relativas al marco de autenticación	4
4 Abreviaturas	5
5 Convenios	6
SECCIÓN 2 – AUTENTICACIÓN SIMPLE.....	7
6 Procedimiento de autenticación simple	7
6.1 Generación de información de identificación protegida.....	8
6.2 Procedimiento de autenticación simple protegida	8
6.3 Tipo de atributo contraseña de usuario.....	9
SECCIÓN 3 – AUTENTICACIÓN FUERTE.....	9
7 Base de la autenticación fuerte	9
8 Obtención de una clave pública de usuario	10
8.1 Optimización del volumen de información obtenido del directorio	13
8.2 Ejemplo	14
9 Firmas digitales.....	16
10 Procedimientos de autenticación fuerte	18
10.1 Visión de conjunto	18
10.2 Autenticación unidireccional.....	18
10.3 Autenticación bidireccional.....	19
10.4 Autenticación tridireccional	20
11 Gestión de claves y certificados	21
11.1 Generación de pares de claves.....	21
11.2 Gestión de certificados	21
12 Extensiones de certificados y de CRL	23
12.1 Introducción	23
12.2 Información de claves y de políticas	24
12.2.1 Requisitos	24
12.2.2 Campos de extensión de certificados y de CRL	25
12.3 Atributos de sujeto de certificado y de expedidor de certificado	29
12.3.1 Requisitos	29
12.3.2 Campos de extensión de certificado y de CRL.....	30
12.4 Constricciones de trayecto de certificación	31
12.4.1 Requisitos	31
12.4.2 Campos de extensión de certificado	32
12.4.3 Procedimiento de procesamiento del trayecto de certificación	35
12.5 Extensiones de la CRL básica	37
12.5.1 Requisitos	37
12.5.2 Campos de extensión de CRL y de asiento de CRL	38

Reemplazada por una versión más reciente

	<i>Página</i>
12.6 Puntos de distribución de CRL y CRL delta	40
12.6.1 Requisitos	40
12.6.2 Campos de extensión de certificado	41
12.6.3 Campos de extensión de CRL y de asiento de CRL	42
12.6.4 Tipo de atributo para las CRL delta	44
12.7 Reglas de concordancia	44
12.7.1 Concordancia exacta de certificados	44
12.7.2 Concordancia de certificados	44
12.7.3 Concordancia exacta de pares de certificados	45
12.7.4 Concordancia de pares de certificados	45
12.7.5 Concordancia exacta de listas de certificados	46
12.7.6 Concordancia de listas de certificados	46
12.7.7 Concordancia de identificadores de algoritmo	47
13 Obtención de atributos certificados	47
13.1 Certificados de atributos	47
13.2 Atributo certificado de atributos	48
13.3 Regla de concordancia de certificado de atributos	48
13.4 Trayectos de certificados de atributos	48
13.5 Lista de revocación de certificados de atributos	49
Anexo A – Marco de autenticación en ASN.1	50
Anexo B – Requisitos de seguridad	60
B.1 Peligros	60
B.2 Servicios de seguridad	60
B.3 Mecanismos de seguridad	61
B.4 Peligros contra los que protegen los servicios de seguridad	62
B.5 Negociación de servicios y mecanismos de seguridad	62
Anexo C – Introducción a la criptografía de claves públicas	63
Anexo D – El criptosistema de claves públicas RSA	65
D.1 Alcance y campo de aplicación	65
D.2 Definiciones	65
D.3 Símbolos y abreviaturas	65
D.4 Descripción	66
D.5 Requisitos de seguridad	66
D.5.1 Longitudes de las claves	66
D.5.2 Generación de claves	66
D.6 Exponente público	66
D.7 Conformidad	67
Anexo E – Funciones de troceo (Hash)	68
E.1 Requisitos de las funciones de troceo	68
Anexo F – Peligros contra los que ofrece protección el método de autenticación fuerte	69
Anexo G – Confidencialidad de los datos	70
G.1 Introducción	70
G.2 Confidencialidad de los datos por cifrado asimétrico	70
G.3 Confidencialidad de los datos por cifrado simétrico	70
Anexo H – Definición de referencia de los identificadores de objeto para algoritmo	71
Anexo I – Bibliografía	72
Anexo J – Ejemplos de la utilización de constricciones del trayecto de certificación	73
J.1 Ejemplo 1: Utilización de constricciones básicas	73
J.2 Ejemplo 2: Utilización de constricciones de nombre	73
J.3 Ejemplo 3: Utilización de constricciones de correspondencia de políticas y de política	73
Anexo K – Enmiendas y corrigenda	75

Reemplazada por una versión más reciente

Introducción

Esta Recomendación | Norma Internacional, junto con otras Recomendaciones | Normas Internacionales, ha sido elaborada para facilitar la interconexión de los sistemas de procesamiento de información con el fin de proporcionar servicios de directorio. El conjunto de todos estos sistemas, junto con la información de directorio que contienen, puede considerarse como un todo integrado, llamado el *directorio*. La información contenida por el directorio, denominada colectivamente base de información de directorio (DIB, *directory information base*), se utiliza típicamente para facilitar la comunicación entre, con o sobre objetos tales como entidades de aplicación, personas, terminales y listas de distribución.

El directorio desempeña un papel importante en la interconexión de sistemas abiertos (OSI), cuyo objetivo es permitir, con un mínimo de acuerdos técnicos fuera de las propias normas de interconexión, la interconexión de sistemas de procesamiento de información:

- de diferentes fabricantes;
- sometidos a gestiones diferentes;
- de diferentes grados de complejidad; y
- de diferentes fechas de construcción.

Muchas aplicaciones tienen exigencias de seguridad para la protección contra las amenazas a la comunicación de información. Algunos peligros o amenazas comúnmente conocidos, junto con los servicios de seguridad y los mecanismos que se pueden utilizar contra ellos, se describen brevemente en el anexo B. Virtualmente, todos los servicios de seguridad dependen de que las identidades de las partes comunicantes sean fiablemente conocidas, es decir, de la autenticación.

Esta Recomendación | Norma Internacional define un marco para el suministro de servicios de autenticación por el directorio a sus usuarios. Estos usuarios incluyen el propio directorio, así como otras aplicaciones y servicios. El directorio puede emplearse muy bien para satisfacer sus necesidades de autenticación y de otros servicios de seguridad, porque es el lugar natural del cual las partes comunicantes pueden obtener la información de autenticación de cada una de las demás: el conocimiento que es la base de la autenticación. El directorio es el lugar natural porque contiene otras informaciones que se requieren para la comunicación y que se obtienen con anterioridad al inicio de la comunicación. La obtención de la información de autenticación de un copartípe potencial en la comunicación, desde el directorio, es, con este enfoque, similar a la obtención de una dirección. Debido al vasto alcance del directorio para los fines de comunicación, se espera que este marco de autenticación será ampliamente usado por una gama de aplicaciones.

Esta tercera edición revisa y mejora técnicamente la segunda edición de la presente Recomendación | Norma Internacional, pero no la substituye. Las implementaciones pueden seguir alegando conformidad con la segunda edición. Sin embargo, en algún punto, no se admitirá la segunda edición (es decir, los defectos informados ya no serán resueltos). Se recomienda que las implementaciones se conformen con esta tercera edición lo antes posible.

Esta tercera edición especifica las versiones 1 y 2 de los protocolos de directorio.

Las ediciones primera y segunda especifican también la versión 1. La mayor parte de los servicios y protocolos especificados en esta edición están diseñados para funcionar según la versión 1. Cuando se ha negociado la versión 1, las diferencias entre los servicios y entre los protocolos definidos en las tres ediciones se acomodan utilizando las reglas de extensibilidad definidas en la edición de 1997 de la Rec. UIT-T X.519 | ISO/CEI 9594-5. No obstante, algunos servicios y protocolos mejorados, por ejemplo, los errores signados, no funcionarán a menos que todas las entidades de directorio que participan en la operación hayan negociado la versión 2.

Los implementadores deben observar que existe un proceso de resolución de defectos y que se pueden aplicar correcciones a esta Recomendación | Norma Internacional en forma de corrigenda técnicos. Se aplicarán correcciones idénticas a esta Recomendación | Norma Internacional en forma de una guía del implementador. Se puede obtener de la Secretaría del subcomité una lista de los corrigenda técnicos aprobados para esta Recomendación | Norma Internacional. Los corrigenda técnicos publicados están disponibles en las organizaciones nacionales de normalización. La guía del implementador puede obtenerse en el sitio Web de la UIT.

El anexo A, que es parte integrante de esta Recomendación | Norma Internacional, proporciona el módulo ASN.1, que contiene todas las definiciones asociadas con el marco de autenticación.

El anexo B, que no es parte integrante de esta Recomendación | Norma Internacional, describe los requisitos de seguridad.

El anexo C, que no es parte integrante de esta Recomendación | Norma Internacional, es una introducción a la criptografía de claves públicas.

Reemplazada por una versión más reciente

El anexo D, que no es parte integrante de esta Recomendación | Norma Internacional, describe el criptosistema de claves públicas RSA.

El anexo E, que no es parte integrante de esta Recomendación | Norma Internacional, describe las funciones hash.

El anexo F, que no es parte integrante de esta Recomendación | Norma Internacional, describe los peligros contra los que ofrece protección el método de autenticación.

El anexo G, que no es parte integrante de esta Recomendación | Norma Internacional, describe la confidencialidad de los datos.

El anexo H, que no es parte integrante de esta Recomendación | Norma Internacional, define los identificadores de objeto asignados a los algoritmos de autenticación y criptación, en ausencia de un registro formal.

El anexo I, que no es parte integrante de esta Recomendación | Norma Internacional, contiene una bibliografía.

El anexo J, que no es parte integrante de esta Recomendación | Norma Internacional, contiene ejemplos de la utilización de las constricciones del trayecto de certificación.

El anexo K, que no es parte integrante de esta Recomendación | Norma Internacional, enumera las enmiendas e informes de defectos que han sido incorporados en esta edición de la presente Recomendación | Norma Internacional.

NORMA INTERNACIONAL

RECOMENDACIÓN UIT-T

**TECNOLOGÍA DE LA INFORMACIÓN – INTERCONEXIÓN DE SISTEMAS
ABIERTOS – EL DIRECTORIO: MARCO DE AUTENTICACIÓN**

SECCIÓN 1 – GENERALIDADES

1 Alcance

Esta Recomendación | Norma Internacional:

- indica la forma de la información de autenticación contenida por el directorio;
- describe cómo puede obtenerse la información de autenticación a partir del directorio;
- enuncia los supuestos formulados en cuanto a la formación y al emplazamiento de esa información de autenticación en el directorio;
- define tres modos en los cuales las aplicaciones pueden usar esa información de autenticación para realizar la autenticación, y describe cómo otros servicios de seguridad pueden ser soportados por autenticación.

Esta Recomendación | Norma Internacional describe dos niveles de autenticación: autenticación simple, mediante el uso de una contraseña como verificación de una identidad pretendida, y autenticación fuerte, que implica credenciales formadas usando técnicas criptográficas. Si bien la autenticación simple ofrece cierta protección limitada contra el acceso no autorizado, sólo la autenticación fuerte debe servir de base para ofrecer servicios seguros. No se pretende con ello establecer un marco general para la autenticación; no obstante, puede ser de uso general para aplicaciones en que estas técnicas se consideran adecuadas.

La autenticación (y otros servicios de seguridad) sólo puede suministrarse dentro del contexto de una política de seguridad definida para una aplicación particular. Incumbe a los usuarios de una aplicación definir su propia política de seguridad, la cual puede verse constreñida por los servicios proporcionados según una norma.

Incumbe a las normas definir las aplicaciones que *usan* el marco de autenticación para especificar los intercambios de protocolo que necesitan ser realizados para lograr la autenticación basada en la información de autenticación de directorio. El protocolo usado por las aplicaciones para obtener la información de autenticación de directorio es el protocolo de acceso a directorio (DAP, *directory access protocol*), especificado en la Rec. UIT-T X.519 | ISO/CEI 9594-5.

El método de autenticación fuerte especificado en esta Recomendación | Norma Internacional se basa en los criptosistemas de claves públicas. Es una gran ventaja de esos sistemas el que los certificados de usuario puedan estar contenidos en el directorio como atributos, y ser comunicados libremente dentro del sistema de directorio y obtenidos por los usuarios de directorio del mismo modo que otra información de directorio. Se supone que los certificados de usuario están formados por medios "fuera de línea", y que posteriormente pueden ser introducidos en el directorio. La generación de certificados de usuario la efectúa cierta autoridad de certificación "fuera de línea" que está completamente separada de los DSA en el directorio. En particular, no se imponen requisitos especiales a los suministradores del directorio para almacenar o comunicar certificados de usuario en una manera segura.

En el anexo C se presenta una breve introducción a la criptografía de claves públicas.

En general, el marco de autenticación no depende del uso de un determinado algoritmo criptográfico, siempre que tenga las propiedades descritas en la cláusula 7. Es probable, en la práctica, que se use cierto número de algoritmos diferentes. Sin embargo, dos usuarios que quieran autenticar tienen que soportar el mismo algoritmo criptográfico para que la autenticación se realice correctamente. Así, dentro del contexto de un conjunto de aplicaciones conexas, la elección de un algoritmo único servirá para maximizar la comunidad de usuarios capaces de autenticar y comunicar de manera segura. En el anexo D se presenta un ejemplo de un algoritmo criptográfico de claves públicas.

De manera similar, dos usuarios que desean autenticar tienen que admitir la misma función de troceo (hash) (véase 3.3.14) (utilizada en la formación de credenciales y testigos de autenticación). En principio, también en este caso se podría utilizar un número de funciones de troceo alternativas, a expensas de reducir las comunidades de usuarios capaces de autenticar. En el anexo E se presenta una breve introducción a las funciones de troceo.

2 Referencias normativas

Las siguientes Recomendaciones y Normas Internacionales contienen disposiciones que, mediante la referencia hecha en este texto, constituyen disposiciones de la presente Recomendación | Norma Internacional. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y Normas son objeto de revisiones, por lo que se preconiza que los participantes en acuerdos basados en la presente Recomendación | Norma Internacional investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y Normas enumeradas a continuación. Los miembros de la CEI y de la ISO mantienen registros de las Normas Internacionales vigentes. La Oficina de Normalización de las Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones del UIT-T vigentes.

2.1 Recomendaciones | Normas Internacionales idénticas

- Recomendación UIT-T X.411 (1995) | ISO/CEI 10021-4:1997, *Tecnología de la información – Sistemas de tratamiento de mensajes – Sistema de transferencia de mensajes: Definición del servicio abstracto y procedimientos.*
- Recomendación UIT-T X.500 (1997) | ISO/CEI 9594-1:1999, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Visión de conjunto de conceptos, modelos y servicios.*
- Recomendación UIT-T X.501 (1997) | ISO/CEI 9594-2:1997, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Modelos.*
- Recomendación UIT-T X.511 (1997) | ISO/CEI 9594-3:1997, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Definición del servicio abstracto.*
- Recomendación UIT-T X.518 (1997) | ISO/CEI 9594-4:1997, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Procedimientos para operación distribuida.*
- Recomendación UIT-T X.519 (1997) | ISO/CEI 9594-5:1997, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Especificaciones de protocolo.*
- Recomendación UIT-T X.520 (1997) | ISO/CEI 9594-6:1997, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Tipos de atributos seleccionados.*
- Recomendación UIT-T X.521 (1997) | ISO/CEI 9594-7:1997, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Clases de objetos seleccionadas.*
- Recomendación UIT-T X.525 (1997) | ISO/CEI 9594-9:1999, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Replicación.*
- Recomendación UIT-T X.530 (1997) | ISO/CEI 9594-10:1999, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Uso de la gestión de sistemas para la administración del directorio.*
- Recomendación X.660 del CCITT (1992) | ISO/CEI 9834-1:1992, *Tecnología de la información – Interconexión de sistemas abiertos – Procedimientos para la operación de autoridades de registro para interconexión de sistemas abiertos: Procedimientos generales.*
- Recomendación UIT-T X.680 (1994) | ISO/CEI 8824-1:1995, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*
- Recomendación UIT-T X.681 (1994) | ISO/CEI 8824-2:1995, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de objetos de información.*
- Recomendación UIT-T X.682 (1994) | ISO/CEI 8824-3:1995, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de constricciones.*
- Recomendación UIT-T X.683 (1994) | ISO/CEI 8824-4:1995, *Tecnología de la información – Notación de sintaxis abstracta uno: Parametrización de especificaciones de la notación de sintaxis abstracta uno.*

- Recomendación UIT-T X.690 (1994) | ISO/CEI 8825-1:1995, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de las reglas de codificación básica, de las reglas de especificación canónica y de las reglas de codificación distinguida.*
- Recomendación UIT-T X.690¹⁾ corr.2 | ISO/CEI 8825-1¹⁾ corr.2.
- Recomendación UIT-T X.880 (1994) | ISO/CEI 13712-1:1995, *Tecnología de la información – Operaciones a distancia: Conceptos, modelo y notación.*
- Recomendación UIT-T X.881 (1994) | ISO/CEI 13712-2:1995, *Tecnología de la información – Operaciones a distancia: Realizaciones de interconexión de sistemas abiertos: Definición de servicio del elemento de servicio de operaciones a distancia.*

2.2 Pares de Recomendaciones | Normas Internacionales de contenido técnico equivalente

- Recomendación CCITT X.800 (1991), *Arquitectura de seguridad para la interconexión de sistemas abiertos para aplicaciones del CCITT.*
ISO 7498-2:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture.*

2.3 Otras referencias

- ISO/CEI 11770-1, *Information technology – Security techniques – Key management – Part 1: Framework.*
- POSTEL (J.B.), Internet Protocol, RFC 791, *Internet Activities Board*, 1981.
- CROCKER (D.H.), Standard for the Format of ARPA Internet Text Messages, RFC 822, *Internet Activities Board*, 1982.
- MOCKAPETRIS (P.), Domain Names – Implementation and Specification, RFC 1035, *Internet Activities Board*, 1987.
- BERNERS-LEE (T.), Universal Resource Identifiers in WWW, RFC 1630, *Internet Activities Board*, 1994.

3 Definiciones

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

3.1 Definiciones relativas a la arquitectura de seguridad del modelo de referencia OSI

Los siguientes términos se definen en la Rec. X.800 del CCITT e ISO 7498-2:

- a) asimétrico (cifrado);
- b) intercambio de autenticaciones;
- c) información de autenticación;
- d) confidencialidad;
- e) credenciales;
- f) criptografía;
- g) autenticación del origen de datos;
- h) descifrado;
- i) cifrado;
- j) clave;
- k) contraseña;
- l) autenticación de entidad par;
- m) simétrico (cifrado).

¹⁾ Actualmente en estado de proyecto.

3.2 Definiciones relativas al modelo de directorio

Los siguientes términos se definen en la Rec. UIT-T X.501 | ISO/CEI 9594-2:

- a) atributo;
- b) base de información de directorio;
- c) árbol de información de directorio;
- d) agente de sistema de directorio;
- e) agente de usuario de directorio;
- f) nombre distinguido;
- g) inserción o asiento;
- h) objeto;
- i) raíz.

3.3 Definiciones relativas al marco de autenticación

Los siguientes términos se definen en esta Recomendación | Norma Internacional.

3.3.1 certificado de atributo: Conjunto de atributos de un usuario junto con alguna otra información, hechos infalsificables por la firma digital que se ha creado utilizando la clave privada de la autoridad de certificación que la emitió.

3.3.2 testigo de autenticación: Información transportada durante un intercambio de autenticación fuerte, que se puede utilizar para autenticar a quien la envió.

3.3.3 certificado de usuario; certificado de clave pública; certificado: Las claves públicas de un usuario, junto con alguna otra información, hechas infalsificables por el cifrado con la clave privada de la autoridad de certificación que la emitió.

3.3.4 certificado de autoridad de certificación: Certificado para una autoridad de certificación emitida por otra autoridad de certificación.

3.3.5 política de certificado: Conjunto denominado de reglas que indica la aplicabilidad de un certificado a una determinada comunidad y/o clase de aplicación con requisitos de seguridad comunes. Por ejemplo, una determinada política de certificados pudiera indicar la aplicabilidad de un tipo de certificado a la autenticación de transacciones de intercambio electrónico de datos para el comercio de bienes dentro de una gama de precios dada.

3.3.6 usuario de certificado: Entidad que necesita conocer, con certidumbre, la clave pública de otra entidad.

3.3.7 sistema que utiliza el certificado: Una implementación de las funciones definidas en esta Especificación del directorio que son utilizadas por un usuario de certificado.

3.3.8 autoridad de certificación: Autoridad a la cual uno o más usuarios han confiado la creación y asignación de certificados. Facultativamente, la autoridad de certificación puede crear las claves de los usuarios.

3.3.9 trayecto de certificación: Secuencia ordenada de certificados de objetos en el árbol de información de directorio que, junto con la clave pública del objeto inicial en el trayecto, puede ser procesada para obtener la del objeto final en el trayecto.

3.3.10 punto de distribución de lista de revocación de certificado: Una inserción o asiento de directorio u otra fuente de distribución para las listas de revocación de certificado; una lista de revocación de certificado distribuida a través de un punto de distribución de lista de revocación de certificado puede contener asientos de revocación solamente para un subconjunto del conjunto completo de certificados emitidos por una autoridad de certificación o puede contener asientos de revocación para múltiples autoridades de certificación.

3.3.11 sistema criptográfico, criptosistema: Colección de transformaciones de texto claro en texto cifrado y viceversa, en la que la transformación o transformaciones que se han de utilizar son seleccionadas por claves. Las transformaciones son definidas normalmente por un algoritmo matemático.

3.3.12 lista de revocación de certificados-delta: Una lista de revocación de certificados parcial que indica solamente los cambios efectuados desde una emisión anterior de la lista de revocación de certificados.

3.3.13 entidad final: Sujeto del certificado que utiliza su clave pública para otros fines distintos que firmar certificados.

3.3.14 función de troceo: Función (matemática) que hace corresponder valores de un dominio grande (posiblemente muy grande) con una gama más pequeña. La función de troceo es "buena" cuando los resultados de la aplicación de la función a un (gran) conjunto de valores en el dominio se distribuyen uniformemente (y aparentemente al azar) en la gama.

3.3.15 acuerdo de clave: Método para negociar un valor de clave en línea sin transferir la clave, incluso en forma criptada, por ejemplo, la técnica de Diffie-Hellman (para más información sobre los mecanismos de acuerdos de clave, véase ISO/CEI 11770-1).

3.3.16 función unidireccional: Función (matemática) "f" que es fácil de calcular, pero que para un valor "y" en la gama es difícil de calcular para hallar un valor "x" en el dominio de modo que $f(x) = y$. Puede haber unos pocos valores "y" para los cuales hallar "x" no sea difícil computacionalmente.

3.3.17 correspondencia de políticas: Reconocimiento de que, cuando una autoridad de certificación en un dominio certifica una autoridad de certificación en otro dominio, una determinada política de certificado en el segundo dominio puede ser considerada por la autoridad del primer dominio como equivalente (pero no necesariamente idéntica en todos los aspectos) a una determinada política de certificado en el primer dominio.

3.3.18 clave pública: (En un criptosistema de claves públicas) la clave de un par de claves de usuario que es conocida públicamente.

3.3.19 clave privada; clave secreta (término desaconsejado): (En un criptosistema de claves públicas) la clave de un par de claves de usuario que sólo es conocida por ese usuario.

3.3.20 autenticación simple: Autenticación por medio de arreglos de contraseñas simples.

3.3.21 política de seguridad: Conjunto de reglas establecidas por la autoridad de seguridad que rigen la utilización y prestación de servicios y facilidades de seguridad.

3.3.22 autenticación fuerte: Autenticación por medio de credenciales derivadas criptográficamente.

3.3.23 fiduciario: En general, se puede decir que una entidad acepta como "fiduciaria" a una segunda entidad cuando aquélla (la primera entidad) supone que la segunda entidad se comportará exactamente como ella lo espera. Esta relación de confianza se puede aplicar solamente para alguna función específica. El cometido principal de la confianza en el marco de la autenticación es describir la relación entre una entidad autenticadora y una entidad de certificación; una entidad autenticadora tiene que estar segura de que puede confiar en que la autoridad de certificación crea solamente certificados válidos y fiables.

3.3.24 número de serie de certificado: Valor entero, único dentro de la autoridad de certificación expedidora, que está asociado inequívocamente con un certificado expedido por dicha autoridad de certificación.

4 Abreviaturas

A los efectos de la presente Recomendación | Norma Internacional, se utilizan las siguientes siglas.

CA	Autoridad de certificación (<i>certification authority</i>)
CRL	Lista de revocación de certificados (<i>certificate revocation list</i>)
DIB	Base de información de directorio (<i>directory information base</i>)
DIT	Árbol de información de directorio (<i>directory information tree</i>)
DSA	Agente de sistema de directorio (<i>directory system agent</i>)
DUA	Agente de usuario de directorio (<i>directory user agent</i>)
PKCS	Criptosistema de claves públicas (<i>public key cryptosystem</i>)

5 Convenios

Con pequeñas excepciones, esta Especificación de directorio se ha preparado con arreglo a las directrices de "Presentación de textos comunes UIT-T | ISO/CEI" que figuran en la Guía para la cooperación entre el UIT-T y el JTC 1 de la ISO/CEI, de marzo de 1993.

El término "Especificación de directorio" (como en "esta Especificación de directorio") se entenderá en el sentido de esta Recomendación | Norma Internacional. El término "Especificaciones de directorio" se entenderá que designa a todas las Recomendaciones ISO/CEI 9594 de la serie X.500.

Esta Especificación de directorio utiliza el término "sistemas de la edición 1988" para hacer referencia a los sistemas conformes a la primera edición (1988) de las Especificaciones de directorio, es decir, la edición de 1988 de las Recomendaciones CCITT de la serie X.500 y la edición de ISO/CEI 9594:1990. Esta Especificación de directorio utiliza el término "sistemas de la edición 1993" para hacer referencia a los sistemas conformes a la segunda edición (1993) de las Especificaciones de directorio, es decir, la edición de 1993 de las Recomendaciones UIT-T de la serie X.500 y la edición de ISO/CEI 9594:1995. Los sistemas conformes a esta tercera edición de las Especificaciones de directorio se indican como "sistemas de la edición 1997".

Esta Especificación de directorio presenta la notación ASN.1 con caracteres del tipo Times Roman, de 9 puntos en negritas. Cuando los tipos y valores ASN.1 aparecen en texto normal, se diferencian del texto normal presentándolos en el tipo Times Roman, de 9 puntos en negritas. Los nombres de los procedimientos, a los que se hace referencia cuando se especifica la semántica del procesamiento, se diferencian del texto normal presentándolos en el tipo Helvética en negritas. Los permisos de control de acceso se presentan en el tipo Helvética en cursivas.

Si los elementos de una lista están numerados (en lugar de utilizar "-" o letras), se considerarán pasos de un procedimiento.

La notación usada en esta Especificación de directorio se define en el cuadro 1.

Cuadro 1 – Notación

Notación	Significado
X_p	Clave pública de un usuario X.
X_s	Clave privada de X.
$X_p[I]$	Cifrado de alguna información, I, mediante la clave pública de X.
$X_s[I]$	Cifrado de I mediante la clave privada de X.
$X\{I\}$	La firma de I por el usuario de X. Consiste en I con un sumario cifrado añadido.
$CA(X)$	Autoridad de certificación del usuario X.
$CA^n(X)$	(Donde $n > 1$): $CA(CA(\dots n \text{ veces} \dots (X)))$
$X_1 \langle X_2 \rangle$	Certificado del usuario X_2 emitido por la autoridad de certificación X_1 .
$X_1 \langle X_2 \rangle X_2 \langle X_3 \rangle$	Cadena de certificados (puede tener una longitud arbitraria), donde cada ítem es el certificado para la autoridad de certificación que produjo el siguiente. Es funcionalmente equivalente al siguiente certificado $X_1 \langle X_{n+1} \rangle$. Por ejemplo, la posesión de $A \langle B \rangle B \langle C \rangle$ confiere la misma capacidad que $A \langle C \rangle$, a saber, la aptitud para hallar C_p dada A_p .
$X_{1p} \bullet X_1 \langle X_2 \rangle$	La operación de desenvolver un certificado (o cadena de certificados) para extraer una clave pública. Es un operador infijo, cuyo operando izquierdo es la clave pública de una autoridad de certificación, y cuyo operando derecho es un certificado emitido por esa autoridad de certificación. El resultado es la clave pública del usuario cuyo certificado es el operando derecho. Por ejemplo: $A_p \bullet A \langle B \rangle B \langle C \rangle$ denota la operación de usar la clave pública de A para obtener la clave pública de B, B_p , de su certificado, seguido por el uso de B_p para desenvolver el certificado de C. El resultado de la operación es la clave pública de C, C_p .
$A \rightarrow B$	Un trayecto de certificación de A a B, formado por una cadena de certificados, que comienza por $CA(A) \langle CA^2(A) \rangle$ y termina por $CA(B) \langle B \rangle$.
NOTA – Cuando se introducen las notaciones, los símbolos X, X_1 , X_2 , etc., aparecen en lugar de los nombres de los usuarios, mientras que el símbolo I aparece en lugar de una información arbitraria.	

SECCIÓN 2 – AUTENTICACIÓN SIMPLE

6 Procedimiento de autenticación simple

La autenticación simple tiene por objeto proporcionar una autorización local basada en un nombre distinguido de usuario, una contraseña (opcional) convenida bilateralmente y un entendimiento mutuo sobre los medios para utilizar y tratar esta contraseña dentro de un solo dominio. La utilización de la autenticación simple tiene como finalidad inicial el uso local solamente, es decir, a la autenticación de entidades pares entre un DUA y un DSA, o entre un DSA y otro DSA. La autenticación simple puede efectuarse de varios modos:

- la transferencia del nombre distinguido del usuario y la contraseña (opcional) en lenguaje ordinario (no protegido) al receptor, para su evaluación;
- la transferencia del nombre distinguido del usuario, la contraseña, y un número aleatorio y/o una indicación de tiempo, todo lo cual se protege mediante la aplicación de una función unidireccional;
- la transferencia de la información protegida descrita en b) junto con un número aleatorio y/o una indicación de tiempo, todo lo cual se protege por la aplicación de una función unidireccional.

NOTA 1 – No se exige que las funciones unidireccionales aplicadas sean diferentes.

NOTA 2 – Los procedimientos de señalización para proteger las contraseñas pueden ser una cuestión de interés para la ampliación de esta Recomendación | Norma Internacional.

Cuando las contraseñas no están protegidas, se proporciona un mínimo grado de seguridad para impedir un acceso no autorizado. Esto no debe considerarse una base para servicios seguros. La protección del nombre distinguido y de la contraseña del usuario da un mayor grado de seguridad. Los algoritmos para uso en el mecanismo de protección son, típicamente, funciones unidireccionales no cifrantes, que son muy fáciles de implementar.

El procedimiento general para la obtención de una autenticación simple se muestra en la figura 1.

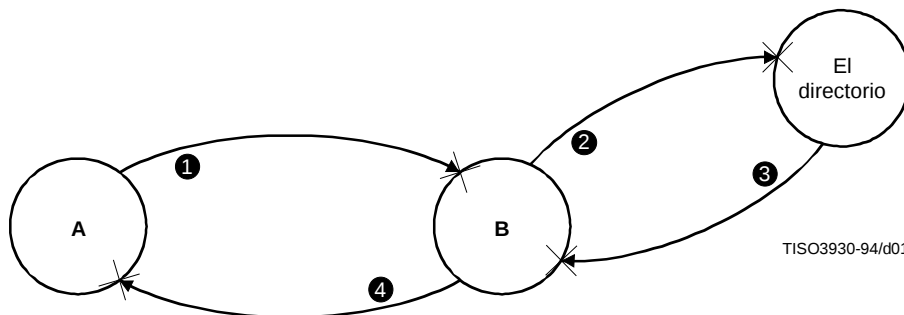


Figura 1 – Procedimiento de autenticación simple no protegida

Comprende los siguientes pasos:

- un usuario originador A envía su nombre distinguido y contraseña a un usuario receptor (o destinatario) B;
- B envía el nombre distinguido contemplado y la contraseña de A al directorio, donde la contraseña se comprueba contra la mantenida como el atributo de **UserPassword** (contraseña de usuario) dentro de la inserción de directorio para A (usando la operación comparar del directorio);
- el directorio confirma (o rechaza) a B que las credenciales son válidas;
- el éxito (o fracaso) de la autenticación puede comunicarse a A.

La forma básica de la autenticación simple comprende solamente el paso 1) y, después de que B ha verificado el nombre distinguido y la contraseña, puede incluir el paso 4).

6.1 Generación de información de identificación protegida

La figura 2 muestra dos métodos que pueden emplearse para generar información de identificación protegida. f_1 y f_2 son funciones unidireccionales (que pueden ser idénticas o diferentes) y las indicaciones de tiempo y los números aleatorios son opcionales y están sujetos a acuerdos bilaterales.

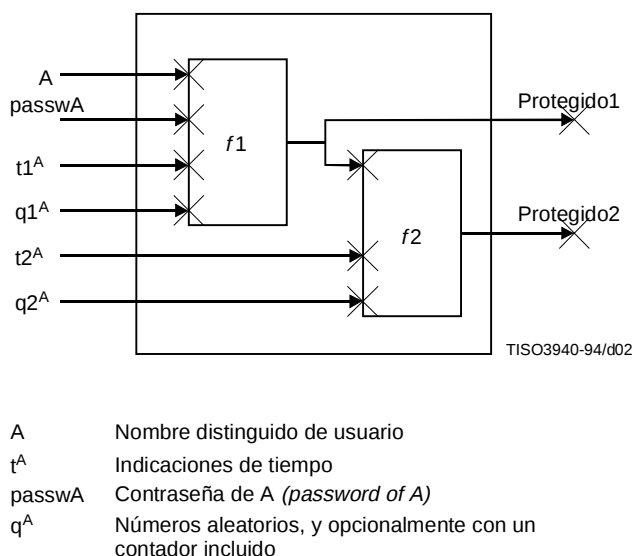


Figura 2 – Autenticación simple protegida

6.2 Procedimiento de autenticación simple protegida

La figura 3 ilustra el procedimiento de autenticación simple protegida.

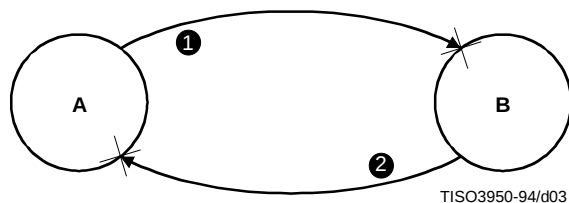


Figura 3 – El procedimiento de autenticación simple protegida

Comprende los siguientes pasos (inicialmente sólo se utiliza f_1):

- 1) El usuario de origen, usuario A, envía su información de identificación protegida (Autenticador1) al usuario B. La protección se consigue aplicando la función unidireccional (f_1) de la figura 2, donde la indicación de tiempo y/o el número aleatorio (si se utiliza) tienen por finalidad minimizar la reproducción y ocultar la contraseña.

La protección de la contraseña de A se realiza de la siguiente forma:

$$Protegido1 = f_1(t_1^A, q_1^A, A; passwdA)$$

La información transportada a B tiene la forma siguiente:

$$Autenticador1 = t_1^A, q_1^A, A; protegido1$$

- 2) B verifica la información de identificación protegida ofrecida por A (utilizando para ello el nombre distinguido y, opcionalmente, la indicación de tiempo y/o el número aleatorio proporcionado por A, junto con una copia local de la contraseña de A) y genera una copia protegida local de la contraseña de A (de la

forma protegido1). B compara según el criterio de igualdad la información de identificación contemplada (protegido1) con el valor generado localmente.

- 3) B confirma (o rechaza) a A la verificación de la información de identificación protegida.

El procedimiento descrito puede modificarse para dar una mayor protección mediante el empleo de f_1 y f_2 . Las diferencias principales son las siguientes:

- 1) A envía su información de identificación protegida (adicionalmente) (autenticador2) a B. Una protección adicional se obtiene aplicando una segunda función unidireccional, f_2 , como se ilustra en la figura 2. Esta mayor protección adopta la forma siguiente:

$$\text{Protegido2} = f_2(t_2^A, q_2^A, \text{protegido1})$$

La información transportada a B tiene la forma:

$$\text{Autenticador2} = t_1^A, t_2^A, q_1^A, q_2^A, A, \text{protegido2}$$

Para la comparación, B genera un valor local de la contraseña adicionalmente protegida de A y lo compara (según el criterio de igualdad) con el de protegido2 (esto es similar en principio al paso 1) de 6.2).

- 2) B confirma (o rechaza) a A la verificación de la información de identificación protegida.

NOTA – Los procedimientos definidos en estos incisos se especifican sobre la base de A y B. Atendiendo a la aplicación al directorio (especificada en la Rec. UIT-T X.511 | ISO/CEI 9594-3 y Rec. UIT-T X.518 | ISO/CEI 9594-4), A podría ser un DUA vinculado a un DSA, B; alternatively A, podría ser un DSA vinculado a otro DSA, B.

6.3 Tipo de atributo contraseña de usuario

Un tipo de atributo contraseña de usuario contiene la contraseña de un objeto. Un valor de atributo para la contraseña de usuario es una cadena especificada por el objeto.

userPassword	ATTRIBUTE ::=	{
WITH SYNTAX		OCTET STRING (SIZE (0..ub-user-password))
EQUALITY MATCHING RULE		octetStringMatch
ID		id-at-userPassword }

SECCIÓN 3 – AUTENTICACIÓN FUERTE

7 Base de la autenticación fuerte

El enfoque de la autenticación fuerte adoptado en esta Especificación de directorio utiliza las propiedades de una familia de sistemas criptográficos, conocidos como criptosistemas de claves públicas (PKCS, *public key cryptosystems*). Estos criptosistemas, también descritos como asimétricos, implican un par de claves, una privada y una pública, y no una sola clave, como los sistemas criptográficos convencionales. El anexo C da una breve introducción a estos criptosistemas y sus propiedades útiles para la autenticación. Para que un PKCS sea utilizable en este marco de autenticación, actualmente, debe tener la propiedad de que ambas claves del par de claves puedan ser usadas para el cifrado, empleándose la clave secreta para descifrar si se usó la clave pública, y empleándose la clave pública para descifrar si se usó la clave secreta. Dicho sea en otras palabras, $X_p \bullet X_s = X_s \bullet X_p$ siendo X_p/X_s funciones de cifrado/descifrado que utilizan las claves pública/secreta de X.

NOTA – En una futura y posible ampliación podrán especificarse otros tipos de PKCS, es decir, tipos que no requieran la propiedad de permutabilidad y que puedan ser soportados sin grandes modificaciones de esta Especificación de directorio.

Este marco de autenticación no obliga a usar un criptosistema en particular. Se pretende que el marco sea aplicable a cualquier criptosistema de clave pública adecuado, y soportará por consiguiente cambios en los métodos usados como un resultado de futuros avances en criptografía, técnicas matemáticas o capacidades de computación. Sin embargo, dos usuarios que desean autenticar tienen que soportar el mismo algoritmo criptográfico para que la autenticación se realice correctamente. Así, dentro del contexto de un conjunto de aplicaciones relacionadas, la elección de un solo algoritmo servirá para maximizar la comunidad de usuarios capaces de autenticar y comunicar con seguridad. Un algoritmo criptográfico, que probablemente sea ampliamente usado, se especifica en el anexo D.

La autenticación se basa en que cada usuario posea un nombre distinguido único. La atribución de nombres distinguidos es responsabilidad de las autoridades de denominación. Cada usuario tiene por consiguiente que confiar en que las autoridades de denominación no expidan nombres distinguidos duplicados.

Cada usuario queda identificado por el hecho de estar en posesión de la clave secreta. Un segundo usuario puede determinar si su copartípe en la comunicación está en posesión de la clave secreta, y puede usar esto para corroborar que su copartípe en la comunicación es en realidad el usuario. La validez de esta corroboración depende de que la clave secreta permanezca confidencial para el usuario.

Para que un usuario determine que su copartípe en la comunicación está en posesión de la clave secreta de otro usuario, deberá, él mismo, estar en posesión de la clave pública de ese usuario. Si bien la obtención del valor de esta clave pública a partir de la inserción del usuario en el directorio es inmediata, la verificación de su corrección plantea ciertos problemas. Puede haber varias formas posibles de realizar esto: la cláusula 8 describe un proceso por el cual una clave pública de usuario puede ser verificada por referencia al directorio. Este proceso sólo puede operar si hay una cadena ininterrumpida de puntos de confianza, en el directorio, entre los usuarios que solicitan autenticación. Esta cadena puede construirse identificando un punto común de confianza. Este punto común de confianza deberá estar enlazado con cada usuario por una cadena ininterrumpida de puntos de confianza.

8 Obtención de una clave pública de usuario

Para que un usuario confíe en el procedimiento de autenticación, tiene que obtener la clave pública del otro usuario desde una fuente en la cual confía. Dicha fuente, llamada autoridad de certificación (CA, *certification authority*), usa el algoritmo de clave pública para certificar la clave pública, produciendo un *certificado*. El certificado, cuya forma se especifica en esta cláusula, tiene las siguientes propiedades:

- cualquier usuario con acceso a la clave pública de la autoridad de certificación puede extraer la clave pública que fue certificada;
- ninguna parte que no sea la autoridad de certificación puede modificar el certificado sin que esto sea detectado (los certificados son infalsificables).

Como los certificados son infalsificables, pueden publicarse insertándolos en el directorio, sin que éste tenga que tomar disposiciones especiales para protegerlos.

NOTA 1 – Aunque las CA están definidas inequívocamente por un nombre distinguido en el DIT, esto no implica que exista una relación entre la organización de las CA y el DIT.

Una autoridad de certificación produce el certificado de un usuario firmando (véase la cláusula 9) una colección de informaciones, incluidos el nombre distinguido y la clave pública del usuario, así como un *identificador único* opcional con información adicional sobre el usuario. No se especifica aquí la forma exacta del contenido del identificador único, que se deja a la autoridad de certificación y podría ser, por ejemplo, un identificador de objeto, un certificado, una fecha u otra forma de certificación sobre la validez del nombre distinguido. Específicamente, el certificado de un usuario con el nombre distinguido A, producido por la autoridad de certificación CA, tiene la forma siguiente:

$$CA\langle\langle A \rangle\rangle = CA\{V, SN, AI, CA, UCA, A, UA, Ap, T^A\}$$

donde V es la versión del certificado, SN es el número de serie de certificado, AI es el identificador del algoritmo utilizado para firmar el certificado, UCA es el indicador único opcional del CA, UA es el identificador único opcional del usuario A y T^A indica el periodo de validez del certificado, y consiste en dos fechas, la primera y la última en las que el certificado es válido. El periodo de validez del certificado es el intervalo de tiempo durante el cual la CA garantiza que mantendrá la información sobre la situación del certificado, es decir que publicará la fecha de revocación. Dado que se supone que T^A se cambie en periodos de no menos de 24 horas, se espera que los sistemas puedan usar el Tiempo Universal Coordinado como una base de tiempo de referencia. La firma puede ser comprobada en cuanto a su validez por cualquier usuario que conozca CAp. El siguiente tipo de datos ASN.1 puede usarse para representar certificados:

Certificate	::= SIGNED { SEQUENCE {
version	[0] Version DEFAULT v1,
serialNumber	CertificateSerialNumber,
signature	AlgorithmIdentifier,
issuer	Name,
validity	Validity,
subject	Name,
subjectPublicKeyInfo	SubjectPublicKeyInfo,

```

issuerUniqueIdentifier [1] IMPLICIT UniqueIdentifier OPTIONAL,
                        -- if present, version must be v2 or v3
subjectUniqueIdentifier [2] IMPLICIT UniqueIdentifier OPTIONAL,
                        -- if present, version must be v2 or v3
extensions [3] Extensions OPTIONAL
                        -- If present, version must be v3 -- } }

Version ::= INTEGER { v1(0), v2(1), v3(2) }

CertificateSerialNumber ::= INTEGER

AlgorithmIdentifier ::= SEQUENCE {
    algorithm ALGORITHM.&id ({SupportedAlgorithms}),
    parameters ALGORITHM.&Type ({SupportedAlgorithms}{ @algorithm}) OPTIONAL }
-- Definition of the following information object set is deferred, perhaps to standardized
-- profiles or to protocol implementation conformance statements. The set is required to
-- specify a table constraint on the parameters component of AlgorithmIdentifier

SupportedAlgorithms ALGORITHM ::= { ... }

Validity ::= SEQUENCE {
    notBefore Time,
    notAfter Time }

SubjectPublicKeyInfo ::= SEQUENCE {
    algorithm AlgorithmIdentifier,
    subjectPublicKey BIT STRING }

Time ::= CHOICE {
    utcTime UTCTime,
    generalizedTime GeneralizedTime }

Extensions ::= SEQUENCE OF Extension

Extension ::= SEQUENCE {
    extnId EXTENSION.&id ({ExtensionSet}),
    critical BOOLEAN DEFAULT FALSE,
    extnValue OCTET STRING
    -- contains a DER encoding of a value of type &ExtnType
    -- for the extension object identified by extnId -- }

ExtensionSet EXTENSION ::= { ... }

```

Antes que se utilice un valor de **Time** en cualquier operación de comparación, por ejemplo, como parte de una regla de concordancia en una búsqueda, y si se ha escogido la sintaxis de **Time** como el tipo **UTCTime**, el valor del campo de año de dos cifras se racionalizará en un valor de año de cuatro cifras, como sigue:

- Si el valor de 2 cifras es 00 a 49 inclusive, se sumará 2000 al mismo.
- Si el valor de 2 cifras es 50 a 99 inclusive, se sumará 1900 al mismo.

NOTA 2 – El empleo de **GeneralizedTime** puede impedir el interfuncionamiento con implementaciones que no tienen la posibilidad de escoger entre **UTCTime** o **GeneralizedTime**. Es responsabilidad de quienes especifican los dominios en los que se utilizarán los certificados definidos en esta Especificación de directorio, por ejemplo, grupos de configuración, en lo referente a cuando se puede utilizar **GeneralizedTime**. En ningún caso se debe utilizar **UTCTime** para representar fechas más allá del año 2049.

El campo de **extensiones** permite añadir nuevos campos a la estructura sin modificar la definición ASN.1. Un campo de extensión está formado por un identificador de extensión, una bandera de condición crítica, y una codificación canónica de un valor de datos de un tipo ASN.1 asociado con la extensión identificada. Para las extensiones en las que el orden de las extensiones individuales dentro de la SECUENCIA es importante, la especificación de estas extensiones individuales incluirá las reglas relativas al significado de su orden. Cuando una implementación que procesa un certificado no reconoce una extensión, si la bandera de condición crítica está puesta a FALSO, puede pasar por alto esa extensión. Si la bandera está puesta a VERDADERO, las extensiones no reconocidas motivarán que la estructura sea considerada no válida, es decir, en un certificado, una extensión crítica no reconocida motivará que fracase la validación de una firma que utiliza ese certificado.

Si dentro de la extensión aparecen elementos desconocidos, y la extensión no está marcada como crítica, se ignorarán los elementos desconocidos conforme a las reglas de extensibilidad documentada en 7.5.2.2 de la Rec. UIT-T X.519 | ISO/CEI 9594-5.

Las extensiones específicas pueden ser definidas en Recomendaciones | Normas Internacionales o por cualquier organización que lo necesite. El identificador de objeto que identifica a una extensión se definirá de acuerdo con la Rec. CCITT X.660 | ISO/CEI 9834-1. Las extensiones de normas para certificado se definen en la cláusula 12.

Se utiliza la siguiente clase de objetos para definir extensiones específicas.

EXTENSION ::= CLASS

```
{
    &id          OBJECT IDENTIFIER UNIQUE,
    &ExtnType
}
WITH SYNTAX
{
    SYNTAX      &ExtnType
    IDENTIFIED BY &id
}
```

NOTA 3 – Cuando la autoridad de denominación reasigne un nombre distinguido a un usuario diferente, las CA pueden utilizar el identificador único para distinguir entre los casos de reutilización. Sin embargo, si el mismo usuario recibe certificados de múltiples CA, se recomienda que las CA coordinen la asignación de identificadores únicos como parte de sus procedimientos de registro de usuarios.

La inserción directorio de cada usuario, A, que está participando en una autenticación fuerte, contiene el certificado (o los certificados) de A. Dicho certificado es generado por una autoridad de certificación de A, que es una entidad en el DIT. Una autoridad de certificación de A, que puede no ser única, se denota por CA(A), o simplemente CA, si se sobreentiende A. La clave pública de A puede ser así descubierta por cualquier usuario que conoce la clave pública de CA. El descubrimiento de claves públicas es por tanto recursivo.

Si el usuario A, que trata de obtener la clave pública del usuario B, ya ha obtenido la clave pública de CA(B) el proceso habrá terminado. A fin de permitir que A obtenga la clave pública de CA(B), la inserción de directorio de cada autoridad de certificación, X, contiene un número de certificados. Estos certificados son de dos tipos. En primer lugar, hay certificados de X en sentido de ida, denominado directos, generados por otras autoridades de certificación. En segundo lugar, hay certificados de X en sentido de retorno, denominados inversos, generados por la propia X, los cuales son claves públicas certificadas de otras autoridades de certificación. La existencia de estos certificados permite a los usuarios construir trayectos de certificación de un punto a otro.

La lista de los certificados que se necesitan para permitir a un determinado usuario descubrir la clave pública de otro se conoce como el *trayecto de certificación*. Cada ítem en la lista es un certificado de la autoridad de certificación para la siguiente. Un trayecto de certificación de A a B (designado por $A \rightarrow B$):

- comienza por un certificado producido por CA(A), a saber $CA(A) \ll X^1 \gg$ para alguna entidad X^1 ;
- continúa con ulteriores certificados $X^i \ll X^{i+1} \gg$;
- finaliza con el certificado de B.

Un trayecto de certificación forma lógicamente una cadena ininterrumpida de puntos de confianza en el árbol de información de directorio, entre dos usuarios que desean autenticar. El método preciso empleado por los usuarios A y B para obtener trayectos de certificación $A \rightarrow B$ y $B \rightarrow A$ puede variar. Una manera de facilitar esto consiste en organizar una jerarquía de CA, que puede o no coincidir con la totalidad o una parte de la jerarquía del DIT. La ventaja de esto es que los usuarios que tienen CA en la jerarquía pueden establecer entre sí un trayecto de certificación utilizando el directorio sin ninguna información previa; para que esto sea posible, cada CA puede almacenar un certificado y un certificado inverso designado como correspondiente a su CA superior.

Los certificados están contenidos en inserciones de directorio como atributos de tipo **UserCertificate**, **CACertificate** y **CrossCertificatePair**. Estos tipos de atributos son conocidos por el directorio. Se puede actuar sobre estos atributos utilizando las mismas operaciones de protocolo empleadas para atributos. La definición de estos tipos puede encontrarse en 3.3; la especificación de estos tipos de atributo es la siguiente:

```
userCertificate ATTRIBUTE ::= {
    WITH SYNTAX      Certificate
    EQUALITY MATCHING RULE certificateExactMatch
    ID               id-at-userCertificate}
```

```

cACertificate ATTRIBUTE ::= {
  WITH SYNTAX Certificate
  EQUALITY MATCHING RULE certificateExactMatch
  ID id-at-cACertificate }

crossCertificatePair ATTRIBUTE ::= {
  WITH SYNTAX CertificatePair
  EQUALITY MATCHING RULE certificatePairExactMatch
  ID id-at-crossCertificatePair }

CertificatePair ::= SEQUENCE {
  forward [0] Certificate OPTIONAL,
  reverse [1] Certificate OPTIONAL
  -- at least one of the pair shall be present -- }

```

Un usuario puede obtener uno o más certificados de una o más autoridades de certificación. Cada certificado lleva el nombre de la autoridad de certificación que lo expidió. Los siguientes tipos de datos ASN.1 pueden utilizarse para representar certificados y un trayecto de certificación:

```

Certificates ::= SEQUENCE {
  userCertificate Certificate,
  certificationPath ForwardCertificationPath OPTIONAL }

CertificationPath ::= SEQUENCE {
  userCertificate Certificate,
  theCACertificates SEQUENCE OF CertificatePair OPTIONAL }

```

Además, puede utilizarse el siguiente tipo de datos ASN.1 para representar el trayecto de certificación de ida. Este componente contiene el trayecto de certificación que puede volver a señalar hacia el originador.

```

ForwardCertificationPath ::= SEQUENCE OF CrossCertificates

CrossCertificates ::= SET OF Certificate

```

8.1 Optimización del volumen de información obtenido del directorio

En el caso general, antes de que los usuarios puedan autenticar mutuamente, el directorio tiene que suministrar los trayectos completos de certificación, y de certificación de retorno. Sin embargo, en la práctica, la cantidad de información que hay que obtener del directorio para una instancia particular de autenticación se puede reducir por los medios siguientes:

- si los usuarios que quieren autenticar son servidos por la misma autoridad de certificación, el trayecto de certificación resulta trivial y los usuarios desenvuelven directamente los certificados de cada uno de los otros;
- si los CA de los usuarios forman una jerarquía, un usuario podría almacenar claves públicas, certificados y certificados inversos de todas las autoridades de certificación entre el usuario y la raíz del DIT. Típicamente, esto entrañaría que el usuario conociera las claves públicas y los certificados de solamente tres o cuatro autoridades de certificación. El usuario sólo necesitaría entonces obtener los trayectos de certificación desde el punto común de confianza;
- si un usuario se comunica frecuentemente con usuarios certificados por otra CA en particular, este usuario pudiera aprender el trayecto de certificación a ese CA y el trayecto de certificación de retorno desde ese CA, con lo que sólo sería necesario obtener el certificado del otro usuario, desde el directorio;
- las autoridades de certificación pueden certificarse mutuamente unas a otras, por acuerdos bilaterales. Como resultado de esto se acorta el trayecto de certificación;
- si dos usuarios han comunicado antes y cada uno ha aprendido el certificado del otro, podrán autenticar sin recurrir al directorio.

De todas formas, los usuarios, después de haber conocido los certificados de cada uno de los demás en base al trayecto de certificación, deberán verificar la validez de los certificados recibidos.

8.2 Ejemplo

La figura 4 ilustra un ejemplo teórico de un fragmento del DIT, en el cual las CA forman una jerarquía. Además de la información indicada en las CA, se supone que cada usuario conoce la clave pública de su autoridad de certificación, y sus propias claves pública y secreta.

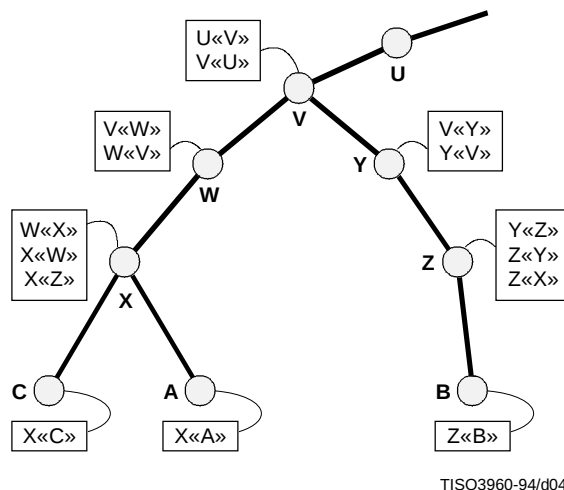


Figura 4 – Jerarquía de CA – Ejemplo teórico

Si las CA de los usuarios forman una jerarquía, A puede obtener los siguientes certificados de directorio para establecer un trayecto de certificación a B:

$$X\langle W\rangle, W\langle V\rangle, V\langle Y\rangle, Y\langle Z\rangle, Z\langle B\rangle$$

Una vez que A ha obtenido estos certificados, puede desenvolver secuencialmente el trayecto de certificación para obtener el contenido del certificado de B, incluido Bp:

$$B_p = X_p \bullet X\langle W\rangle W\langle V\rangle V\langle Y\rangle Y\langle Z\rangle Z\langle B\rangle$$

En general A tiene también que adquirir del directorio los siguientes certificados para establecer el trayecto de certificación de retorno de B a A:

$$Z\langle Y\rangle, Y\langle V\rangle, V\langle W\rangle, W\langle X\rangle, X\langle A\rangle$$

Cuando B recibe estos certificados desde A, puede desenvolver secuencialmente el trayecto de certificación de retorno para obtener el contenido del certificado de A, incluido Ap:

$$A_p = Z_p \bullet Z\langle Y\rangle Y\langle V\rangle V\langle W\rangle W\langle X\rangle X\langle A\rangle$$

Aplicando las optimizaciones de 8.1:

- a) tomando A y C, por ejemplo: ambos conocen X_p , de modo que, sencillamente, A tiene que adquirir directamente el certificado de C. El desenvolvimiento del trayecto de certificación se reduce a:

$$C_p = X_p \bullet X\langle C\rangle$$

y el desenvolvimiento del trayecto de certificación de retorno se reduce a:

$$A_p = X_p \bullet X\langle A\rangle$$

- b) suponiendo que A conociera así $W\langle X\rangle$, W_p , $V\langle W\rangle$, V_p , $U\langle V\rangle$, hacia arriba, etc., la información que A tiene que obtener del directorio para formar el trayecto de autenticación se reduce a:

$$V\langle Y\rangle, Y\langle Z\rangle, Z\langle B\rangle$$

y la información que A tiene que obtener del directorio para formar el trayecto de certificación de retorno se reduce a:

$$Z\langle Y\rangle, Y\langle V\rangle$$

- c) suponiendo que A comunica frecuentemente con usuarios certificados por Z, él puede aprender [además de las claves públicas aprendidas en b)] $V\langle Y\rangle$, $Y\langle V\rangle$, $Y\langle Z\rangle$, y $Z\langle Y\rangle$. Para comunicar con B, sólo necesita por consiguiente obtener $Z\langle B\rangle$ del directorio;
- d) suponiendo que los usuarios certificados por X y Z comunican frecuentemente, entonces $X\langle Z\rangle$ estaría contenido en la inserción de directorio para X, y viceversa (esto se muestra en la figura 4). Si A quiere autenticar hacia B, sólo necesita obtener:

$$X\langle Z\rangle, Z\langle B\rangle$$

para formar el trayecto de certificación, y:

$$Z\langle X\rangle$$

para formar el trayecto de certificación de retorno;

- e) suponiendo que los usuarios A y C han comunicado antes y han aprendido sus certificados respectivos, cada uno puede usar directamente la clave del otro, por ejemplo:

$$C_p = X_p \bullet X\langle C\rangle$$

y

$$A_p = X_p \bullet X\langle A\rangle$$

En el caso más general, las autoridades de certificación no guardan una relación jerárquica. En el ejemplo hipotético de la figura 5, supóngase que un usuario D, certificado por U, desea autenticar al usuario E, certificado por W. La inserción de directorio del usuario D contendrá el certificado $U\langle D\rangle$ y la inserción del usuario E contendrá el certificado $W\langle E\rangle$.

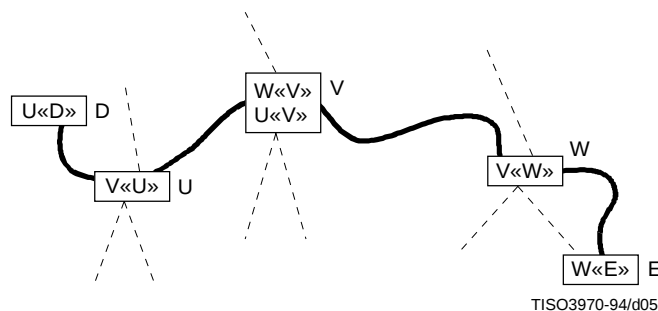


Figura 5 – Ejemplo de trayecto de certificación no jerárquico

Sea V una CA con la cual las CA, U y W han efectuado anteriormente cierto intercambio de redes públicas en una situación de confianza. Como resultado de esto se han generado y almacenado en el directorio certificados $U\langle V\rangle$, $V\langle U\rangle$, $W\langle V\rangle$ y $V\langle W\rangle$. Supóngase que $U\langle V\rangle$ y $W\langle V\rangle$ están almacenados en la inserción de V, $V\langle U\rangle$ está almacenado en el asiento de U, y $V\langle W\rangle$ está almacenado en la inserción de W.

El usuario D debe encontrar un trayecto de certificación E. Este usuario podría utilizar diversos métodos. Uno de ellos consistiría en considerar los usuarios y CA como nodos, y los certificados como arcos en un gráfico dirigido. En estos términos, D debe efectuar una búsqueda en el gráfico para encontrar un trayecto de U a E, siendo uno de ellos $U\langle V\rangle$, $V\langle W\rangle$, $W\langle E\rangle$. Una vez descubierto este trayecto, se puede construir también el trayecto inverso $W\langle V\rangle$, $V\langle U\rangle$, $U\langle D\rangle$.

9 Firmas digitales

En esta cláusula no se pretende especificar una norma para firmas (o firmas) digitales en general, sino especificar los medios para firmar los testigos en el directorio.

La información (info) se firma añadiéndole un sumario cifrado de la información. El sumario se produce por medio de una función hash unidireccional, mientras que el cifrado se lleva a cabo usando la clave secreta del firmante (véase la figura 6). Así:

$$X\{\text{Info}\} = \text{Info}, X_s[h(\text{Info})]$$

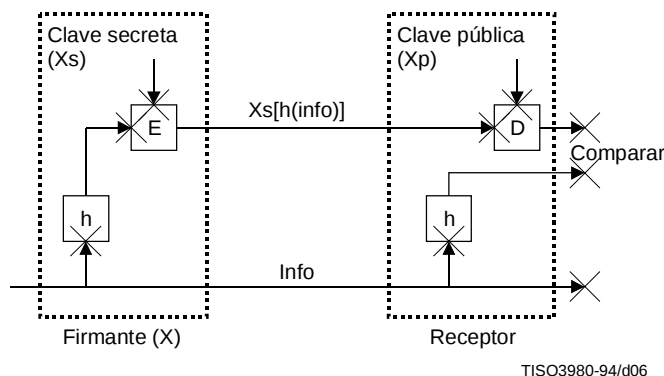


Figura 6 – Firmas digitales

NOTA 1 – El cifrado mediante la clave secreta asegura que la firma no puede ser falsificada. La naturaleza unidireccional de la función hash asegura que la información falsa, generada como para tener el mismo resultado hash (y por consiguiente la firma), no puede ser introducida en sustitución.

El receptor de información firmada verifica la firma:

- aplicando la función hash unidireccional a la información;
- comparando el resultado con el obtenido descifrando la firma mediante la clave pública del firmante.

Este marco de autenticación no impone una sola función hash unidireccional para uso en firmado. Se pretende que el marco sea aplicable a cualquier función hash adecuada, y que por consiguiente admita cambios de los métodos usados, como un resultado de futuros avances en criptografía, técnicas matemáticas o capacidades de computación. Sin embargo, dos usuarios que quieran autenticar tienen que soportar la misma función hash para que la autenticación se realice correctamente. Por consiguiente, dentro del contexto de un conjunto de aplicaciones relacionadas, la elección de una sola función servirá para maximizar la comunidad de usuarios capaces de autenticar y comunicar con seguridad.

La información firmada incluye indicadores que identifican el algoritmo de la función hash y el algoritmo de criptación utilizados para computar la firma digital.

El cifrado de cierto ítem de datos puede describirse utilizando la siguiente macro ASN.1:

```

ENCRYPTED { ToBeEnciphered } ::= BIT STRING ( CONSTRAINED BY {
  -- must be the result of applying an encipherment procedure --
  -- to the BER-encoded octets of a value of -- ToBeEnciphered } )

```

El valor de la cadena de bits se genera tomando los octetos que forman la codificación completa (utilizando las reglas de codificación básica ASN.1 – Rec. UIT-T X.690 | ISO/CEI 8825-1) del valor del tipo **ToBeEnciphered** y aplicando un procedimiento de cifrado a esos octetos.

NOTA 2 – El procedimiento de criptación requiere un acuerdo sobre el algoritmo a aplicar, incluyendo los eventuales parámetros de algoritmo, así como toda clave, valor de inicialización e instrucción de relleno que pueda necesitarse. En los procedimientos de criptación se especificarán los medios para obtener la sintonización de los datos del emisor y del receptor, lo que puede incluir información en los bits que deban transmitirse.

NOTA 3 – El procedimiento de criptación deberá admitir como entrada una cadena de octetos y generar una cadena única de bits, como resultado.

NOTA 4 – El mecanismo para el acuerdo de seguridad sobre el algoritmo de criptación y sus parámetros, el emisor y el receptor de los datos, están fuera del ámbito de esta Especificación de directorio.

La firma de ciertos ítems de datos se forma encriptando una transformación abreviada o "troceada" del ítem, y puede describirse utilizando la siguiente macro ASN.1:

```
ENCRYPTED-HASH { ToBeSigned } ::= BIT STRING ( CONSTRAINED BY {
    -- must be the result of applying a hashing procedure to the DER-encoded octets --
    -- of a value of -- ToBeSigned -- and then applying an encipherment procedure to those octets -- })
```

```
SIGNATURE { ToBeSigned } ::= SEQUENCE {
    algorithmIdentifier AlgorithmIdentifier,
    encrypted ENCRYPTED-HASH { ToBeSigned }}
```

NOTA 5 – El procedimiento de criptación requiere los acuerdos enumerados en la nota 2, y también el acuerdo sobre si los octetos troceados se criptan directamente, o sólo después de codificados como BIT STRING (cadena de bits) utilizando las reglas de codificación básica ASN.1.

Cuando deba asociarse una firma a un tipo de datos, puede utilizarse la siguiente macro ASN.1 para definir el tipo de datos resultantes de la aplicación de una firma a un determinado tipo de datos.

```
SIGNED { ToBeSigned } ::= SEQUENCE {
    toBeSigned ToBeSigned,
    COMPONENTS OF SIGNATURE { ToBeSigned }}
```

A fin de permitir la validación de los tipos **SIGNED** y **SIGNATURE** en un entorno distribuido, se requiere una codificación distinguida. Una codificación distinguida de un valor de datos **SIGNED** o **SIGNATURE** se obtendrá aplicando las reglas de codificación básica definidas en ISO/8825 con las siguientes limitaciones:

- se utilizará la forma definida de codificación de longitud, codificada en el mínimo número de octetos;
- para los tipos cadena, no se utilizará la forma construida de codificación;
- si el valor de un tipo es su valor por defecto, deberá estar ausente;
- los componentes de un tipo conjunto deberán codificarse en orden ascendente de su valor de rótulo;
- los componentes de un tipo conjunto-de se codificarán en orden ascendente de su valor de octeto;
- si el valor de un tipo booleano es verdadero, el octeto de contenido de la codificación deberá fijarse a "FF"₁₆;
- todo bit no utilizado en el octeto final de la codificación de un valor cadena de bits, si existe, deberá fijarse a cero;
- el tipo real se codificará de una manera tal que no se utilicen las bases 8, 10 y 16, y el factor de escala ("scaling factor") binario será cero;
- la codificación de un tiempo UTC se efectuará conforme a lo especificado en la Rec. UIT-T X.690/corr.2 | ISO/CEI 8825-1/corr.2;
- la codificación de un tiempo generalizado se efectuará conforme a lo especificado en la Rec. UIT-T X.690 | ISO/CEI 8825-1.

La generación de una codificación distinguida requiere que la sintaxis abstracta de los datos que se han de codificar se comprenda totalmente. El directorio puede requerir la firma de datos o la comprobación de la signatura de datos o la comprobación de la signatura de datos que contiene extensiones de protocolo desconocidas o sintaxis de atributos desconocidas. El directorio seguirá las siguientes reglas:

- preservará la codificación de la información recibida cuya sintaxis abstracta no conoce totalmente y que espera firmar posteriormente;
- cuando se firman datos para emitir, enviará datos cuya sintaxis se conoce totalmente con una codificación distinguida y cualesquiera otros datos con su codificación protegida, y firmará la codificación real que envía;
- cuando se verifica la firma de los datos recibidos, comprobará la firma frente a los datos recibidos reales en lugar de la conversión de los datos recibidos a una codificación distinguida.

10 Procedimientos de autenticación fuerte

10.1 Visión de conjunto

El enfoque básico de la autenticación se ha resumido anteriormente, esto es: corroborar la identidad demostrando la posesión de una clave secreta. Sin embargo, son posibles muchos procedimientos de autenticación que emplean este enfoque. En general incumbe a una aplicación específica el determinar los procedimientos apropiados, de modo que se cumpla su política de seguridad. Esta cláusula describe tres procedimientos distintos de autenticación, que quizás resulten útiles en una gama de aplicaciones.

NOTA – Esta Especificación de directorio no especifica los procedimientos con el detalle requerido para la implementación. Sin embargo, pueden preverse Recomendaciones | Normas Internacionales adicionales que lo hicieran, sea de una manera específica a la aplicación o en un modo de propósito general.

Los tres procedimientos comprenden diferentes números de intercambios de información de autenticación, y en consecuencia, proporcionan diferentes tipos de seguridades a los participantes. Específicamente:

- a) La autenticación unidireccional, descrita en 10.2 implica una transferencia simple de información desde un usuario (A) prevista para otro (B), y determina lo siguiente:
 - la identidad de A, y que el testigo de autenticación fue generado realmente por A;
 - la identidad de B, y que el testigo de autenticación se previó realmente enviarlo a B;
 - la integridad y "originalidad" (la propiedad de no haber sido enviado dos o más veces) del testigo de autenticación que está siendo transferido.

Las últimas propiedades pueden ser determinadas también para todo otro dato arbitrario adicional en la transferencia.
- b) La autenticación bidireccional, descrita en 10.3, implica, además, una respuesta de B a A. Determina además lo siguiente:
 - que el testigo de autenticación generado en la respuesta fue generado realmente por B y estaba previsto para ser enviado a A;
 - la integridad y originalidad del testigo de autenticación enviado en la respuesta;
 - (opcionalmente), el secreto mutuo de una parte de los testigos.
- c) La autenticación tridireccional, descrita en 10.4, implica, además, una transferencia ulterior de A a B. Determina las mismas propiedades que la autenticación bidireccional, pero lo hace sin necesidad de comprobación de la indicación de la hora de la asociación.

En cada caso donde va a tener lugar una autenticación fuerte, A tiene que obtener la clave pública de B y el trayecto de certificación de retorno de B a A, previamente a cualquier intercambio de información. Esto puede implicar acceso al directorio, como se describió en la cláusula 7 anteriormente. Tal tipo de acceso no se vuelve a mencionar en la descripción de los procedimientos que siguen.

La comprobación de las indicaciones de hora mencionadas en las siguientes subcláusulas solamente es aplicable cuando, o bien se usan relojes sincronizados en un entorno local, o cuando los relojes están sincronizados lógicamente por acuerdos bilaterales. En cualquier caso, se recomienda que se use el Tiempo Universal Coordinado.

En cada uno de los procedimientos de autenticación descritos a continuación se supone que la parte A ha comprobado la validez de todos los certificados en el trayecto de certificación.

10.2 Autenticación unidireccional

Se siguen los pasos indicados en la figura 7:

- 1) A genera r^A , un número no repetitivo, que se usa para detectar ataques de reactuación y para prevenir la falsificación.
- 2) A envía el siguiente mensaje a B:

$$B \rightarrow A, A\{t^A, r^A, B\}$$

donde t^A es una indicación de tiempo. t^A consta de una o dos fechas: la hora de generación del testigo (que es opcional) y la fecha de expiración. Como otra posibilidad, si se debe proporcionar autenticación del origen de datos de "sgnData" por firma digital:

$$B \rightarrow A, A\{t^A, r^A, B, \text{sgnData}\}$$

En los casos en que haya que transportar información que vaya a utilizarse posteriormente como una clave secreta (a dicha información se le llama "encData"):

$$B \rightarrow A, A\{t^A, r^A, B, \text{sgnData}, Bp[\text{encData}]\}$$

La utilización de "encData" como una clave secreta implica que deberá elegirse ésta con cuidado; por ejemplo, deberá procurarse que sea una clave fuerte para cualquier criptosistema utilizado, como se indica en el campo "sgnData" del testigo.

3) B efectúa las acciones siguientes:

- obtiene A_p de $B \rightarrow A$, comprobando que el certificado de A no ha expirado;
- verifica la firma, y por consiguiente la integridad de la información firmada;
- comprueba que el mismo B es el receptor deseado;
- comprueba que la indicación de tiempo está "actual";
- opcionalmente, comprueba que r^A no ha sido reactivada. Esto pudiera lograrse, por ejemplo, haciendo que r^A incluya una parte secuencial que es comprobada por una implementación local para detectar que su valor es único.

r^A es válido hasta la fecha de expiración indicada por t^A . r^A va siempre acompañado por una parte secuencial, que indica que A no repetirá el testigo durante el intervalo de tiempo t^A , y por tanto que no es necesaria la verificación del valor de r^A propiamente dicho.

En todo caso, es razonable para B almacenar la parte secuencial junto con la indicación de hora t^A en lenguaje ordinario junto con la parte del testigo a que se aplicó la función hash, durante el rango de tiempo t^A .

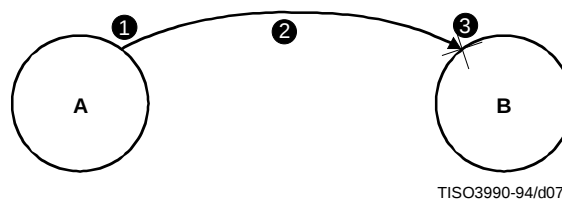


Figura 7 – Autenticación unidireccional

10.3 Autenticación bidireccional

Se siguen los pasos indicados en la figura 8:

- Como en 10.2.
- Como en 10.2.
- Como en 10.2.
- B genera r^B , un número no repetitivo, utilizado para fines similares a los de r^A .
- B envía el siguiente testigo de autenticación a A:

$$B\{t^B, r^B, A, r^A\}$$

donde t^B es una indicación de tiempo definida de la misma manera que t^A .

Como otra posibilidad, si debe proporcionarse autenticación de origen de datos de "sgnData" por firma digital:

$$B\{t^B, r^B, A, r^A, \text{sgnData}\}$$

En los casos en que haya que transportar información que vaya a utilizarse posteriormente como una clave secreta (a dicha información se le llama "encData"):

$$B\{t^B, r^B, A, r^A, \text{sgnData}, A_p[\text{encData}]\}$$

La utilización de "encData" como clave secreta implica que deberá elegirse con cuidado; por ejemplo, deberá ser una clave fuerte para cualquier criptosistema que se utilice en el campo "sgnData" del testigo.

- 6) A ejecuta las siguientes acciones:
 - a) verifica la firma, y por tanto la integridad de la información firmada;
 - b) comprueba que A es el receptor deseado;
 - c) comprueba que la indicación de hora t^B es "corriente";
 - d) opcionalmente, comprueba que r^B no ha sido reactuado [véase 10.2, apartado 3 d)].

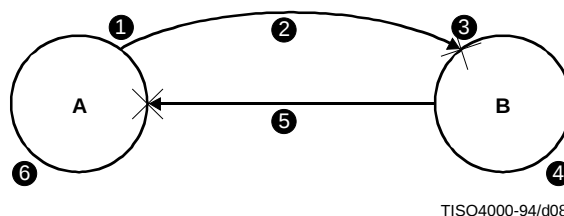


Figura 8 – Autenticación bidireccional

10.4 Autenticación tridireccional

Se siguen los pasos indicados en la figura 9:

- 1) Como en 10.3.
- 2) Como en 10.3. La indicación de tiempo t^A puede ser cero.
- 3) Como en 10.3, excepto que la indicación de tiempo no necesita ser comprobada.
- 4) Como en 10.3.
- 5) Como en 10.3. La indicación de tiempo t^B puede ser cero.
- 6) Como en 10.3, excepto que la indicación de tiempo no necesita ser comprobada.
- 7) A comprueba que el r^A recibido es idéntico al r^A que fue enviado.
- 8) A envía el siguiente testigo de autenticación a B:

$$A\{r^B, B\}$$
- 9) B efectúa las siguientes acciones:
 - a) comprueba la firma y por consiguiente la integridad de la información firmada;
 - b) comprueba que el r^B recibido es idéntico al r^B que fue enviado por B.

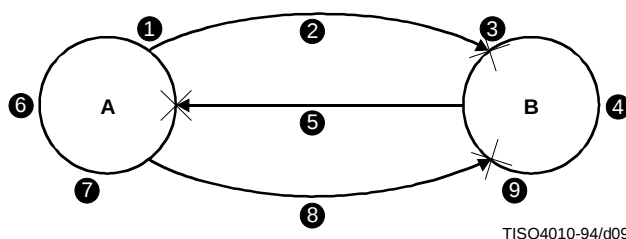


Figura 9 – Autenticación tridireccional

11 Gestión de claves y certificados

11.1 Generación de pares de claves

La política en general de gestión de seguridad de una implementación definirá el ciclo de vida de los pares de claves, y está por consiguiente fuera del alcance del marco de autenticación. Sin embargo, es vital a la seguridad general que todas las claves privadas permanezcan privadas, es decir, sólo conocidas por el usuario al que pertenecen.

Los datos de la clave no son fáciles de recordar por un usuario humano, por lo que hay que emplear un método apropiado para almacenarla en un modo transportable conveniente. Un mecanismo posible sería el uso de una "tarjeta inteligente" ("smart card") que podría contener las claves secreta y (opcionalmente) pública del usuario, el certificado del usuario, y una copia de la clave pública de la autoridad de certificación. El uso de esta tarjeta podría también asegurarse por medio de un número de identificación personal (PIN, *personal identification number*), que aumenta la seguridad del sistema al requerir del usuario la posesión de la tarjeta y que sepa cómo acceder al sistema. El método exacto escogido para almacenar tales datos, sin embargo, está fuera del ámbito de esta Especificación de directorio.

Hay tres modos en los cuales un par de claves del usuario pueden ser producidos:

- El usuario genera su propio par de claves – Este método tiene la ventaja de que una clave privada del usuario nunca es pasada a otra entidad, pero requiere un cierto nivel de competencia por el usuario, como se describe en el anexo D.
- El par de claves es generado por una tercera entidad – La tercera entidad tiene que pasar la clave privada al usuario de una manera físicamente segura, y entonces destruir activamente toda la información relacionada a la creación del par de claves así como las propias claves. Hay que emplear medidas de seguridad física adecuadas para garantizar que la tercera entidad y las operaciones de datos no son objeto de maniobras fraudulentas.
- El par de claves se genera por la CA. Éste es un caso especial de b) y las consideraciones hechas allí son aplicables.

NOTA – La autoridad de certificación ya presenta funcionalidad fiduciaria con respecto al usuario, y estará sujeta a las medidas de seguridad física necesarias. Este método tiene la ventaja de no requerir una transferencia securizada de datos a la CA para la certificación.

El criptosistema en uso impone constricciones (técnicas) particulares a la generación de claves.

11.2 Gestión de certificados

Un certificado asocia la clave pública y el nombre distinguido único del usuario que el mismo describe. Por consiguiente:

- una autoridad de certificación tiene que cerciorarse de la identidad de un usuario antes de crear un certificado para él;
- una autoridad de certificación no expedirá certificados para dos usuarios con el mismo nombre.

Los certificados son producidos fuera de línea y no deberán efectuarse con un mecanismo automático de pregunta/respuesta. La ventaja de esta certificación es que debido a la clave privada de la autoridad de certificación, la CA, nunca se conoce excepto en el caso de la CA aislada y físicamente segura; por tanto, la clave privada de la CA sólo puede ser averiguada por un ataque a la propia CA, lo que hace poco probable un eventual peligro.

Es importante que la transferencia de información a la autoridad de certificación no sea comprometida, y hay que tomar medidas de seguridad física adecuadas. A este respecto:

- Se produciría una seria brecha en la seguridad si la CA expidiera un certificado para un usuario con una clave pública que haya sido objeto de maniobras fraudulentas.
- Si se emplea el medio de generación de los pares de claves de 11.1 b) u 11.1 c), la clave privada de usuario debe ser transferida al usuario de manera segura.
- Si se emplea el medio de generación de pares claves descrito en 11.1 a) u 11.1 b), el usuario puede utilizar diferentes métodos (en línea o fuera de línea) para comunicar su clave pública a la CA de una manera segura. Los métodos en-línea pueden proporcionar una mayor flexibilidad para las operaciones a distancia efectuadas entre el usuario y la CA.

Un certificado es una información disponible públicamente, y no se necesita emplear medidas de seguridad específicas con respecto a su transporte al directorio. Como éste es producido por una autoridad de certificación fuera de línea a nombre de un usuario que recibirá una copia del mismo, el usuario necesita solamente almacenar esta información en su inserción de directorio en un acceso ulterior al directorio. Alternativamente, la CA podría custodiar el certificado para el usuario, en cuyo caso a este agente tendrían que otorgársele derechos de acceso adecuados.

Los certificados tendrán asociada cierta duración, al final de la cual caducan (expiran). A fin de asegurar la continuidad del servicio, la CA garantizará el suministro oportuno de certificados de sustitución que reemplazan a los caducados o próximos a caducar. Hay dos cuestiones conexas:

- La validez de los certificados deberá organizarse de tal modo que la validez de uno entrañe la caducidad del precedente, o se puede permitir que sus periodos de validez se superpongan. Esto último evita que las CA tengan que instalar y distribuir un gran número de certificados que pudieran agotarse en la misma fecha de caducidad (expiración).
- Los certificados caducados normalmente serán sacados del directorio. Es cuestión de política de seguridad y de responsabilidad de la CA mantener los certificados antiguos durante cierto periodo de tiempo, si se presta el servicio de "no repudio de los datos".

Los certificados pueden ser revocados antes de su expiración, por ejemplo si se supone que la clave privada del usuario puede quedar comprometida, o si el usuario ya no debe ser certificado por la CA, o si se supone que el certificado de la CA ha quedado comprometido. Hay cuatro cuestiones conexas:

- La revocación de un certificado de usuario o de un certificado de CA debe ponerse en conocimiento de la CA, y deberá expedirse un nuevo certificado si fuese procedente. La CA podrá entonces informar al propietario del certificado, sobre su revocación, por un procedimiento fuera de línea.
- La CA mantendrá:
 - a) una lista, con indicación de tiempo, de los certificados expedidos que han sido revocados;
 - b) una lista, con indicación de tiempo, de los certificados revocados de todas las CA, conocidos por la CA, expedidos por la CA.

Ambas listas deberán existir, aunque pueden estar vacías.
- El mantenimiento de inserciones de directorio afectadas por las listas de revocaciones, por la CA, es responsabilidad del directorio y sus usuarios, quienes actúan de acuerdo con la política de seguridad. Por ejemplo, el usuario puede modificar su inserción de objeto reemplazando el antiguo certificado por uno nuevo. Este último se utilizará entonces para autenticar el usuario ante el directorio.
- Las listas de revocaciones ("listas negras") se mantienen dentro de inserciones como atributos de tipos "CertificateRevocationList" y "AuthorityRevocationList". Esos atributos pueden ser operados utilizando los mismos procedimientos empleados para otros atributos. Estos tipos de atributo se definen como sigue:

```

certificateRevocationList ATTRIBUTE ::= {
    WITH SYNTAX CertificateList
    EQUALITY MATCHING RULE certificateListExactMatch
    ID id-at-certificateRevocationList }

authorityRevocationList ATTRIBUTE ::= {
    WITH SYNTAX CertificateList
    EQUALITY MATCHING RULE certificateListExactMatch
    ID id-at-authorityRevocationList }

CertificateList ::=
    SIGNED { SEQUENCE {
        version Version OPTIONAL,
        -- if present, version must be v2
        signature AlgorithmIdentifier,
        issuer Name,
        thisUpdate Time,
        nextUpdate Time OPTIONAL,
        revokedCertificates SEQUENCE OF SEQUENCE {
            userCertificate CertificateSerialNumber,
            revocationDate Time,
            crlEntryExtensions Extensions OPTIONAL } OPTIONAL,
            crlExtensions [0] Extensions OPTIONAL }
    }

```

NOTA 1 – La verificación de la lista completa de certificados es un asunto local. Esta lista no tiene porqué estar en un orden determinado a no ser que la CA expedidora haya especificado reglas de ordenación específicas, por ejemplo en esa política de la CA.

NOTA 2 – Si un servicio de no repudio de datos depende de las claves proporcionadas por la CA, dicho servicio deberá asegurar que todas las claves pertinentes de la CA (revocadas o caducadas) y las listas de revocaciones con indicación de tiempo son archivadas y certificadas por una autoridad vigente.

NOTA 3 – Si cualesquiera extensiones incluidas en una **CertificateList** (lista de certificados) se definen como críticas, el elemento de versión de la **CertificateList** estará presente. Si no se incluyen extensiones definidas como críticas, el elemento de versión estará ausente. Esto puede permitir que una implementación que admite únicamente la versión 1 de la CRL siga utilizando la CRL si en su examen de la secuencia **revokedCertificates** de la CRL no encuentra una extensión. Una implementación que admite la versión 2 (o una versión superior) de las CRL puede ser capaz de optimizar su procesamiento si en las primeras fases de éste puede determinar que en la CRL no está presente ninguna extensión crítica.

NOTA 4 – Cuando una implementación que procesa una lista de revocación de certificados no reconoce una extensión crítica en el campo **crlEntryExtensions** (extensiones de inserciones de CRL), supondrá que, como mínimo, el certificado identificado ha sido revocado y ya no es válido, y realizará las acciones adicionales relacionadas con ese certificado revocado dictadas por la política local. Cuando una implementación no reconoce una extensión crítica en el campo **crlExtensions** (extensiones de CRL), supondrá que los certificados identificados han sido revocados y ya no son válidos. Sin embargo, en el último caso, como la lista puede no estar completa, no se puede suponer que los certificados que no han sido identificados como revocados son válidos. En este caso, la política local indicará la acción que se ha de ejecutar. En cualquier caso, la política local puede indicar acciones adicionales y/o más estrictas que las señaladas en esta Especificación.

NOTA 5 – Si una extensión afecta al tratamiento de la lista (por ejemplo, se han de explorar múltiples CRL para examinar toda la lista de certificados revocados, o una inserción puede representar una gama de certificados), esa extensión se indicará como crítica en el campo **crlExtensions** con independencia de dónde se coloca la extensión en la CRL. Una extensión indicada en el campo **crlEntryExtensions** de una inserción se colocará en esa inserción y sólo afectará al certificado o certificados especificados en esa inserción.

NOTA 6 – Las extensiones normalizadas de las CRL se definen en la cláusula 12.

12 Extensiones de certificados y de CRL

12.1 Introducción

Esta cláusula especifica las siguientes extensiones:

- a) *Información de claves y de política:* Estas extensiones de certificados y de CRL transportan información adicional sobre las claves en cuestión, incluidos los identificadores de claves para claves de sujeto y de expedidor, indicadores de utilización prevista o restringida de claves e indicadores de política de certificados.
- b) *Atributos de sujeto y de expedidor:* Estas extensiones de certificados y CRL admiten nombres alternativos, de diversas formas de nombre, para un sujeto de certificado, un expedidor de certificado o un expedidor de CRL. Estas extensiones pueden transmitir también información adicional de atributos sobre el sujeto del certificado, para ayudar a un usuario de certificado a confiar en que el sujeto del certificado es una persona o entidad particular.
- c) *Restricciones de trayecto de certificación:* Estas extensiones de certificado permiten incluir especificaciones de constricciones en certificados de CA, es decir, certificados para CA expedidos por otras CA, con el fin de facilitar el procesamiento automatizado de trayectos de certificación cuando participan múltiples políticas de certificado. Estas múltiples políticas de certificado pueden darse cuando las políticas varían para diferentes aplicaciones en un entorno o cuando se produce interfuncionamiento con entornos externos. Las constricciones pueden restringir los tipos de certificados que pueden ser emitidos por la CA o que pueden producirse subsiguientemente en un trayecto de certificación.
- d) *Extensiones de la CRL básica:* Estas extensiones de CRL permiten que una CRL incluya indicaciones del motivo de revocación, para proporcionar la suspensión temporal de un certificado y que incluya números de secuencia de expedición de CRL para que los usuarios de certificado puedan detectar las CRL que faltan en una secuencia recibida de un expedidor de CRL.
- e) *Puntos de distribución de CRL y CRL-delta:* Estas extensiones de certificados y de CRL permiten dividir el conjunto completo de información de revocación de una CA en CRL separadas y combinar la información de revocación de múltiples CA en una CRL. Estas extensiones admiten también el uso de CRL parciales que indican sólo cambios con respecto a una CRL expedida anteriormente.

La inclusión de cualquier extensión en un certificado o CRL es una opción de la autoridad que emite ese certificado o CRL.

En un certificado o CRL, una extensión se indica como crítica o no crítica mediante banderas. Si una extensión se indica como crítica con banderas y un sistema que utiliza el certificado no reconoce el tipo de campo de extensión o no aplica la semántica de la extensión, dicho sistema considerará que el certificado es no válido. Si una extensión se indica no crítica mediante banderas, un sistema que utiliza el certificado que no reconoce o aplica ese tipo de extensión puede procesar el resto del certificado pasando por alto la extensión. Las definiciones de tipo de extensión en esta Especificación de directorio indican si la extensión es siempre crítica, siempre no crítica o si su condición de crítica puede ser decidida por el expedidor del certificado o de la CRL. El motivo de que algunas extensiones tengan que ser siempre no críticas es permitir que las implementaciones que utilizan el certificado y que no necesitan utilizar estas extensiones las omitan sin afectar la capacidad de interfundar con todas las autoridades de certificación.

NOTA – Un sistema que utiliza certificados puede requerir que ciertas extensiones no críticas estén presentes en un certificado para que ese certificado se considere aceptable. La necesidad de inclusión de esta extensión puede ser requerida por reglas de política local del usuario del certificado o puede ser una regla de política de la CA indicada al sistema que utiliza el certificado mediante la inclusión de un determinado identificador de política de certificado indicando esa extensión como crítica mediante banderas.

Para todas las extensiones de certificados, extensiones de CRL y extensiones de asientos de CRL definidas en esta Especificación de directorio, no habrá más de un caso de cada tipo de extensión en cualquier certificado, CRL, o asiento de CRL, respectivamente.

Esta cláusula define también las reglas de concordancia para facilitar la selección de certificados o de CRL con características específicas a partir de atributos con múltiples valores que tienen múltiples certificados o CRL.

12.2 Información de claves y de políticas

12.2.1 Requisitos

Los siguientes requisitos se relacionan con la información de claves y de política:

- a) La actualización de pares de claves de CA se puede producir a intervalos regulares o en circunstancias especiales. Es necesario que un campo de certificado transporte un identificador de la clave pública que se ha de utilizar para verificar la forma del certificado. Un sistema que utiliza certificados puede emplear estos identificadores para hallar el certificado de CA correcto con el fin de validar la clave pública del expedidor del certificado.
- b) En general, un sujeto de certificado tiene diferentes claves públicas y, en consecuencia, diferentes certificados para diferentes fines, por ejemplo, firma digital y acuerdo de claves de cifrado. Se necesita un campo de certificado para ayudar a un usuario de certificado a seleccionar el certificado correcto para un sujeto dado, para una finalidad determinada o para que una CA pueda estipular que una clave certificada sólo puede ser utilizada para una finalidad determinada.
- c) La actualización de pares de claves de sujeto se puede producir a intervalos regulares o en circunstancias especiales. Es necesario que un campo de certificado transporte un identificador para distinguir entre diferentes claves públicas para el mismo sujeto utilizadas en diferentes instantes de tiempo. Un sistema que emplea certificados puede usar estos identificadores para hallar el certificado correcto.
- d) La clave privada que corresponde a una clave pública certificada se utiliza generalmente durante un periodo diferente de la validez de la clave pública. Con claves de firma digital, el periodo de utilización para la clave privada de firma suele ser más corto que para la clave pública de verificación. El periodo de validez del certificado indica un periodo durante el cual se puede utilizar la clave pública, que no es necesariamente igual al periodo de utilización de la clave privada. En el caso de una clave privada comprometida, el periodo de exposición puede estar limitado si el verificador de la firma conoce el periodo de utilización legítima para la clave privada. Por consiguiente, es necesario poder indicar el periodo de utilización de la clave privada en un certificado.
- e) Como los certificados pueden ser utilizados en entornos en los que se aplican múltiples políticas de certificado, hay que prever la inclusión de información de política de certificado en los certificados.
- f) En el caso de certificación recíproca de una organización a otra, a veces se puede acordar que algunas de las políticas de las dos organizaciones puedan ser consideradas equivalentes. Un certificado de CA tiene que permitir que el expedidor del certificado indique que una de sus propias políticas de certificado es equivalente a otra política de certificado en el dominio de la otra CA. Esto se conoce como correspondencia de políticas.

- g) Un usuario de un sistema de cifrado o de firma digital que utiliza certificados definidos en esta Especificación de directorio tiene que ser capaz de identificar de antemano los algoritmos admitidos por otros usuarios.

12.2.2 Campos de extensión de certificados y de CRL

Se definen los siguientes campos de extensión:

- a) *Identificador de clave de autoridad.*
- b) *Identificador de clave de sujeto.*
- c) *Utilización de clave.*
- d) *Periodo de utilización de clave privada.*
- e) *Políticas de certificado.*
- f) *Correspondencias de políticas.*

Estos campos de extensión se utilizarán solamente como extensiones de certificados, salvo para el identificador de clave de autoridad, que también se puede utilizar como una extensión de CLR. A menos que se indique otra cosa, estas extensiones se pueden utilizar en los certificados de CA y en los certificados de entidad final.

Además, se define un atributo de directorio para sustentar la selección de un algoritmo que se ha de utilizar al comunicar con una entidad final distante que emplea certificados definidos en esta Especificación de directorio.

12.2.2.1 Campo de identificador de clave de autoridad

Este campo, que se puede utilizar como extensión de certificado o extensión de CRL, identifica la clave pública que se ha de utilizar para verificar la firma de este certificado o CRL. Permite distinguir distintas claves utilizadas por la misma CA (por ejemplo, cuando se produce actualización de claves). Este campo se define como sigue:

authorityKeyIdentifier EXTENSION ::= {

SYNTAX **AuthorityKeyIdentifier**
IDENTIFIED BY **id-ce-authorityKeyIdentifier }**

AuthorityKeyIdentifier ::= SEQUENCE {

keyIdentifier **[0] KeyIdentifier OPTIONAL,**
authorityCertIssuer **[1] GeneralNames OPTIONAL,**
authorityCertSerialNumber **[2] CertificateSerialNumber OPTIONAL }**
(WITH COMPONENTS { ..., authorityCertIssuer PRESENT,
 authorityCertSerialNumber PRESENT} |
WITH COMPONENTS { ..., authorityCertIssuer ABSENT,
 authorityCertSerialNumber ABSENT})

KeyIdentifier ::= OCTET STRING

La clave puede ser identificada por un identificador de clave explícito en el componente **keyIdentifier**, mediante la identificación de un certificado para la clave (que da el expedidor de certificado en el componente **authorityCertIssuer** y el número de serie de certificado en el componente **authorityCertSerialNumber**), o por el identificador de clave explícito y la identificación de un certificado para la clave. Si se utilizan ambas formas de identificación, el expedidor del certificado o de la CRL asegurará que son coherentes. Un identificador de clave será único con respecto a todos los identificadores de clave para la autoridad expedidora del certificado o CRL que contiene la extensión. Una implementación que admite esta extensión no tiene que ser capaz de procesar todas las formas de nombres en el componente **authorityCertIssuer**. (Véase 12.3.2.1 para los detalles del tipo **GeneralNames**.)

Las autoridades de certificación asignarán números de serie de certificado de modo que cada par (expedidor, número de serie de certificado) identifique únicamente a un solo certificado.

Esta extensión siempre es no crítica.

12.2.2.2 Campo de identificador de clave de sujeto

Este campo identifica la clave pública que se certifica. Permite diferenciar distintas claves utilizadas por el mismo sujeto (por ejemplo, cuando se produce la actualización de claves). Este campo se define como sigue:

subjectKeyIdentifier EXTENSION ::= {

SYNTAX SubjectKeyIdentifier

IDENTIFIED BY id-ce-subjectKeyIdentifier }

SubjectKeyIdentifier ::= KeyIdentifier

Un identificador de claves será único con respecto a todos los identificadores de claves para el sujeto con el cual se utiliza. Esta extensión siempre es no crítica.

12.2.2.3 Campo de utilización de claves

Este campo, que indica la finalidad para la cual se utiliza la clave pública certificada, se define como sigue:

keyUsage EXTENSION ::= {

SYNTAX KeyUsage

IDENTIFIED BY id-ce-keyUsage }

KeyUsage ::= BIT STRING {

digitalSignature (0),

nonRepudiation (1),

keyEncipherment (2),

dataEncipherment (3),

keyAgreement (4),

keyCertSign (5),

cRLSign (6),

encipherOnly (7),

decipherOnly (8) }

Los bits en el tipo **KeyUsage** son los siguientes:

- a) **digitalSignature:** Para verificar firmas digitales que tienen finalidad distinta a las identificadas en b), f) o g) a continuación.
- b) **nonRepudiation:** Para verificar firmas digitales utilizadas con el fin de proporcionar un servicio de no repudio que protege contra la entidad firmante que niega falsamente alguna acción (excluida la firma de certificado o CRL, como en f) o g) a continuación).
- c) **keyEncipherment:** Para cifrar claves u otra información de seguridad, por ejemplo, para el transporte de claves.
- d) **dataEncipherment:** Para cifrar datos de usuario, pero no claves u otra información de seguridad como se indica en c).
- e) **keyAgreement:** Para utilización como una clave de acuerdo de clave pública.
- f) **keyCertSign:** Para verificar una firma de CA en certificados.
- g) **cRLSing:** Para verificar una firma de CA en la CRL.
- h) **encipherOnly:** Clave de acuerdo de clave pública para utilizar solamente en datos cifrados cuando se utiliza con el bit **keyAgreement** también fijado (el significado con otro bit de utilización de clave fijado es indefinido).
- i) **decipherOnly:** Clave de acuerdo de clave pública para utilización solamente en datos de cifrado cuando se utiliza con el bit **keyAgreement** también fijado (el significado con otro bit de utilización de clave fijado es indefinido).

El bit **keyCertSign** se ha de utilizar solamente en certificados de CA. Si **KeyUsage** se pone en **keyCertSign** y si la extensión de constricciones básicas está presente en el mismo certificado, el valor del componente **cA** de esa extensión se pondrá en **VERDADERO**. Las CA pueden emplear también otros de los bits de utilización de clave definidos en **KeyUsage**, por ejemplo, **digitalSignature**, para proporcionar la autenticación e integridad de las transacciones de administración en línea.

A opción del expedidor del certificado, esta extensión puede ser crítica o no crítica.

Si la extensión se indica como crítica mediante banderas, el certificado se utilizará solamente para los fines para los cuales el bit de utilización de clave correspondiente se ha puesto a uno.

Si la extensión se indica como no crítica mediante banderas, indica la finalidad o finalidades previstas de la clave, y se puede utilizar para hallar la clave/certificado correctos de una entidad que tiene múltiples claves/certificados. Es un campo de asesoramiento y no supone que la utilización de la clave está restringida a la finalidad indicada. Un bit puesto a cero indica que la clave no está prevista para esa finalidad. Si todos los bits son cero, indica que la clave está prevista para alguna finalidad distinta a las enumeradas.

12.2.2.4 Campo de utilización de clave ampliado

Este campo indica una o más finalidades en las cuales puede utilizarse la clave pública certificada, además o en lugar de las finalidades básicas indicadas en el campo de extensión de utilización de clave. Este campo se define como sigue:

```
extKeyUsage EXTENSION ::= {
    SYNTAX      SEQUENCE SIZE (1..MAX) OF KeyPurposeId
    IDENTIFIED BY id-ce-extKeyUsage }
```

KeyPurposeId ::= OBJECT IDENTIFIER

Las finalidades de la clave pueden ser definidas por cualquier organización que lo necesite. Los identificadores de objeto utilizados para identificar las finalidades de la clave se asignarán con arreglo a la Rec. CCITT X.660 | ISO/CEI 9834-1.

A opción del expedidor del certificado, esta extensión puede ser crítica o no crítica.

Si la extensión está indicada como crítica mediante banderas, el certificado utilizará únicamente una de las finalidades indicadas.

Si la extensión se indica como no crítica mediante banderas, indica la finalidad o finalidades previstas de la clave, y se puede utilizar para hallar la clave/certificado correctos de una entidad que tiene múltiples claves/certificados. Es un campo de asesoramiento y no supone que la utilización de la clave esté restringida por la autoridad de certificación a la finalidad indicada. (No obstante, la utilización de aplicaciones puede requerir que se indique una finalidad determinada a fin de que el certificado sea aceptable para esa aplicación.)

Si un certificado contiene un campo de utilización de clave crítico y un campo de utilización de clave ampliado crítico, ambos campos se procesarán independientemente y el certificado será utilizado únicamente para una finalidad compatible con ambos campos. Si no hay ninguna finalidad compatible con ambos campos, el certificado no será utilizado para ninguna finalidad.

12.2.2.5 Campo de periodo de utilización de clave privada

Este campo indica el periodo de utilización de la clave privada correspondiente a la clave pública certificada. Es aplicable solamente para claves de firma digital. Este campo se define como sigue:

```
privateKeyUsagePeriod EXTENSION ::= {
    SYNTAX      PrivateKeyUsagePeriod
    IDENTIFIED BY id-ce-privateKeyUsagePeriod }
```

```
PrivateKeyUsagePeriod ::= SEQUENCE {
    notBefore    [0]      GeneralizedTime OPTIONAL,
    notAfter     [1]      GeneralizedTime OPTIONAL }
( WITH COMPONENTS { ..., notBefore PRESENT } |
  WITH COMPONENTS { ..., notAfter PRESENT } )
```

El componente **notBefore** indica la fecha y hora a partir de las cuales la clave privada se podrá utilizar para la firma. Si el componente **notBefore** no está presente, no se proporciona información sobre cuándo comienza el periodo de uso válido de la clave privada. El componente **notAfter** indica la última fecha y hora en la cual se podrá utilizar la clave privada para la firma. Si el componente **notAfter** no está presente, no se proporciona información sobre cuándo termina el periodo de uso válido de la clave privada.

Esta extensión siempre es no crítica.

NOTA 1 – El periodo de uso válido de la clave privada puede ser diferente de la validez certificada de la clave pública indicada por el periodo de validez de certificado. Con claves de firma digital, el periodo de utilización para la clave privada de firma suele ser más corto que el de la clave pública de verificación.

NOTA 2 – Si el verificador de una firma digital desea comprobar que la clave no ha sido revocada, por ejemplo, debido a que la clave está comprometida, hasta la verificación, debe existir aún un certificado válido para la clave pública en el momento de la verificación. Después que el certificado o certificados para la clave pública han expirado, un verificador de firmas no puede depender de los compromisos notificados mediante las CRL.

12.2.2.6 Campo de políticas de certificado

Este campo enumera políticas de certificado reconocidas por la CA expedidora, que se aplican al certificado, junto con información calificadora facultativa perteneciente a estas políticas de certificado. Generalmente, diferentes políticas de certificado se relacionarán con diferentes aplicaciones que pueden utilizar la clave certificada. Este campo se define como sigue:

```
certificatePolicies EXTENSION ::= {
    SYNTAX      CertificatePoliciesSyntax
    IDENTIFIED BY id-ce-certificatePolicies }

CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
    policyIdentifier  CertPolicyId,
    policyQualifiers SEQUENCE SIZE (1..MAX) OF
        PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
    policyQualifierId  CERT-POLICY-QUALIFIER.&id
        ({SupportedPolicyQualifiers}),
    qualifier          CERT-POLICY-QUALIFIER.&Qualifier
        ({SupportedPolicyQualifiers}{@policyQualifierId})
        OPTIONAL }

SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { ... }
```

Un valor del tipo **PolicyInformation** identifica y transporta información calificadora para una política de certificado. El componente **policyIdentifier** contiene un identificador de una política de certificado y el componente **policyQualifiers** contiene valores de calificador de política para ese elemento.

A opción del expedidor del certificado, esta extensión puede ser crítica o no crítica.

Si la extensión se indica como crítica mediante banderas, significa que el certificado sólo se utilizará para esa finalidad y de acuerdo con las reglas de una de las políticas de certificado indicadas. Las reglas de una política determinada pueden requerir que el sistema que utiliza el certificado procese el valor calificador de una manera particular.

Si la extensión se indica como no crítica mediante banderas, la utilización de esta extensión no constriñe necesariamente la extensión del certificado a las políticas enumeradas. Sin embargo, un usuario de certificado puede requerir que una política determinada esté presente para utilizar el certificado (véase 12.4.3). A opción del usuario del certificado, los calificadores de política pueden ser procesados o pasados por alto.

Las políticas de certificado y los tipos de calificador de políticas de certificado pueden ser definidos por cualquier organización que lo necesite. Los identificadores de objeto utilizados para identificar políticas de certificado y tipos de calificador de políticas de certificado se asignarán de acuerdo con la Rec. CCITT X.660 | ISO/CEI 9834-1. Se utiliza la siguiente clase de objeto ASN.1 para definir los tipos de calificador de políticas de certificado:

```
CERT-POLICY-QUALIFIER ::= CLASS {
    &id      OBJECT IDENTIFIER UNIQUE,
    &Qualifier OPTIONAL }
WITH SYNTAX {
    POLICY-QUALIFIER-ID      &id
    [QUALIFIER-TYPE      &Qualifier] }
```

La definición de un tipo de calificador de política incluirá:

- una declaración de la semántica de los posibles valores; y
- una indicación de si el identificador de calificador puede aparecer en una extensión de políticas de certificado sin un valor acompañante y, si es así, la semántica requerida en dicho caso.

NOTA – Se puede especificar que un calificador tenga cualquier tipo ASN.1. Cuando se prevé que el calificador ha de ser utilizado principalmente con aplicaciones que no tienen funciones de decodificación de ASN.1, se recomienda que se especifique el tipo **OCTET STRING**. El valor **OCTET STRING** de ASN.1 se puede transportar como un valor de calificador codificado de acuerdo con cualquier convenio especificado por la organización que define el elemento de política.

12.2.2.7 Campo de correspondencias de políticas

Este campo, que sólo se utilizará en certificados de CA, permite que un expedidor de certificado indique que, para los fines del usuario de un trayecto de certificación que contiene este certificado, una de las políticas de certificado del expedidor se puede considerar equivalente a una política de certificado diferente utilizada en el dominio de la CA de que se trata. Este campo se define como sigue:

```
policyMappings EXTENSION ::= {
    SYNTAX      PolicyMappingsSyntax
    IDENTIFIED BY id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
    issuerDomainPolicy      CertPolicyId,
    subjectDomainPolicy     CertPolicyId }
```

El componente **issuerDomainPolicy** indica una política de certificado que es reconocida en el dominio de la CA expedidora y que se puede considerar equivalente a la política de certificado indicada en el componente **subjectDomainPolicy** que se reconoce en el dominio de la otra CA.

Esta extensión siempre es no crítica.

NOTA 1 – A continuación figura un ejemplo de correspondencia de políticas: el dominio del Gobierno de Estados Unidos de América tiene una política denominada *Canadian Trade* y el Gobierno de Canadá puede tener una política denominada *US Trade*. Aunque la identificación y definición de las dos políticas son distintas, puede haber un acuerdo entre los dos gobiernos para aceptar trayectos de certificación que se extienden a través de las fronteras dentro de las reglas de estas dos políticas para fines pertinentes.

NOTA 2 – La correspondencia de políticas supone gastos administrativos importantes y la participación de personal adecuadamente diligente y autorizado para la toma de decisiones conexas. En general, es preferible acordar una utilización más global de políticas comunes que aplicar la correspondencia de políticas. En el ejemplo anterior, sería preferible que Estados Unidos de América, Canadá y México acordasen una política común para *North American Trade*.

NOTA 3 – Se prevé que la correspondencia de políticas será práctica solamente en entornos limitados en los cuales las declaraciones de política son muy simples.

12.2.2.8 Atributo de algoritmos admitidos

Se define un atributo de directorio para sustentar la selección de un algoritmo que se ha de utilizar cuando se comunica con una entidad final distante que emplea certificados definidos en esta Especificación de directorio. La siguiente ASN.1 define este atributo (de múltiples valores):

```
supportedAlgorithms ATTRIBUTE ::= {
    WITH SYNTAX SupportedAlgorithm
    EQUALITY MATCHING RULE algorithmIdentifierMatch
    ID id-at-supportedAlgorithms }

SupportedAlgorithm ::= SEQUENCE {
    algorithmIdentifier      AlgorithmIdentifier,
    intendedUsage            [0] KeyUsage OPTIONAL,
    intendedCertificatePolicies [1] CertificatePoliciesSyntax OPTIONAL }
```

Cada valor de un atributo de múltiples valores tendrá un valor **algorithmIdentifier** distinto. El valor del componente **intendedUsage** proporciona una indicación de la utilización prevista del algoritmo (véase 12.2.2.3 para usos reconocidos). El valor del componente **intendedCertificatePolicies** identifica a las políticas de certificado y, facultativamente, a los calificadores de políticas de certificado con las cuales se puede utilizar el algoritmo identificado.

12.3 Atributos de sujeto de certificado y de expedidor de certificado

12.3.1 Requisitos

Los siguientes requisitos se relacionan con los atributos de sujeto de certificado y expedidor de certificado:

- a) Los certificados tienen que ser utilizables por aplicaciones que emplean una variedad de formas de nombre, incluidos los nombres de correo electrónico Internet, nombres de dominio Internet, direcciones de originador/recibiente X.400 y nombres de participantes en intercambio electrónico de datos (EDI). Por consiguiente, es necesario poder asociar con seguridad múltiples nombres de una variedad de formas de nombre con un sujeto de certificado o un expedidor de certificado o de CRL.

- b) Un usuario de certificado puede necesitar conocer con seguridad cierta información de identificación sobre un sujeto para tener confianza en que el sujeto es realmente la persona o cosa deseada. Por ejemplo, se puede requerir información, tal como la dirección postal, el puesto ocupado en una organización, o una imagen. Esta información puede ser representada convenientemente como atributos de directorio, pero estos atributos no forman parte necesariamente del nombre distinguido. En consecuencia, se necesita un campo de certificado para transportar atributos de directorio adicionales, además de los que figuran en el nombre distinguido.

12.3.2 Campos de extensión de certificado y de CRL

Se definen los siguientes campos de extensión:

- Nombre alternativo de sujeto.*
- Nombre alternativo de expedidor.*
- Atributos de directorio del sujeto.*

Estos campos se utilizarán solamente como extensiones de certificado, salvo el nombre alternativo de expedidor, que se puede utilizar también como una extensión de CRL. Como extensiones de certificado, pueden estar presentes en certificados de CA o en certificados de entidad final.

12.3.2.1 Campo de nombre alternativo de sujeto

Este campo contiene uno o más nombres alternativos, utilizando cualquiera de una variedad de formas de nombre, para la entidad que está confinada por la CA a la clave pública certificada. Este campo se define como sigue:

```
subjectAltName EXTENSION ::= {
    SYNTAX GeneralNames
    IDENTIFIED BY id-ce-subjectAltName }
```

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

```
GeneralName ::= CHOICE {
    otherName           [0]  INSTANCE OF OTHER-NAME,
    rfc822Name          [1]  IA5String,
    dNSName             [2]  IA5String,
    x400Address         [3]  ORAddress,
    directoryName       [4]  Name,
    ediPartyName        [5]  EDIPartyName,
    uniformResourceIdentifier [6] IA5String,
    iPAddress           [7]  OCTET STRING,
    registeredID        [8]  OBJECT IDENTIFIER }
```

OTHER-NAME ::= TYPE-IDENTIFIER

```
EDIPartyName ::= SEQUENCE {
    nameAssigner      [0]  DirectoryString {ub-name} OPTIONAL,
    partyName         [1]  DirectoryString {ub-name} }
```

Los valores en las alternativas del tipo **GeneralName** son nombres de diversas formas, como sigue:

- **otherName** es un nombre de cualquier forma definido como un caso de la clase de objeto de información **OTHER-NAME**.
- **rfc822Name** es una dirección de correo electrónico Internet definida de acuerdo con Internet RFC 822.
- **dNSName** es un nombre de dominio Internet definido de acuerdo con Internet RFC 1035.
- **x400Address** es una dirección de O/R definida de acuerdo con la Rec. UIT-T X.411 | ISO/CEI 10021-4.
- **directoryName** es un nombre de directorio definido de acuerdo con la Rec. UIT-T X.501 | ISO/CEI 9594-2.
- **ediPartyName** es un nombre de una forma acordada entre socios de intercambio electrónico de datos que comunican; el componente **nameAssigner** identifica a una autoridad que asigna valores únicos de nombre en el componente **partyName**.
- **uniformResourceIdentifier** es un identificador de recurso uniforme para World-Wide Web definido de acuerdo con Internet RFC 1630.
- **iPAddress** es una dirección de Protocolo Internet definida de acuerdo con Internet RFC 791, representada como una cadena binaria.
- **registeredID** es un identificador de cualquier objeto registrado asignado de acuerdo con la Rec. CCITT X.660 | ISO/CEI 9834-1.

Para cada forma de nombre utilizada en el tipo **GeneralName** habrá un sistema de registro de nombres que asegura que todo nombre utilizado inequívocamente identifica a una entidad ante los expedidores de certificado y los usuarios de certificados.

A opción del expedidor del certificado, esta extensión puede ser crítica o no crítica. Una implementación que admite esta extensión no tiene que ser capaz de procesar todas las formas de nombre. Si la extensión se indica como crítica mediante banderas, por lo menos una de las formas de nombre que está presente será reconocida y procesada; en los demás casos, el certificado se considerará no válido. Aparte de la restricción precedente, un sistema que utiliza certificados puede pasar por alto cualquier nombre con una forma de nombre no reconocida o no admitida. Se recomienda que, a condición de que el campo de sujeto del certificado contenga un nombre de directorio que identifique inequívocamente al sujeto, este campo se indique como no crítico mediante banderas.

NOTA 1 – La utilización de la clase **TYPE-IDENTIFIER** se describe en los anexos A y C a la Rec. UIT-T X.681 | ISO/CEI 8824-2.

NOTA 2 – Si este campo de extensión está presente y se indica como crítico mediante banderas, el campo **subject** del certificado puede contener un nombre nulo (por ejemplo, una secuencia de nombre distinguidos relativos cero), en cuyo caso el sujeto es identificado solamente por el nombre o nombres en esta extensión.

12.3.2.2 Campo de nombre alternativo de expedidor

Este campo contiene uno o más nombres alternativos, utilizando cualquiera de una variedad de formas de nombre, para el expedidor del certificado o de la CRL. Este campo se define como sigue:

```
issuerAltName EXTENSION ::= {
    SYNTAX      GeneralNames
    IDENTIFIED BY id-ce-issuerAltName }
```

A opción del expedidor del certificado o de la CRL, esta extensión puede ser crítica o no crítica. Una implementación que admite esta extensión tiene que ser capaz de procesar todas las formas de nombres. Si la extensión se indica como crítica, por lo menos una de las formas de nombres que está presente debe ser reconocida y procesada; en los demás casos, el certificado o CRL se considerará no válido. Aparte de la restricción precedente, un sistema que utiliza certificados puede pasar por alto cualquier nombre con una forma de nombre no reconocida o no admitida. Se recomienda que, a condición de que el campo de expedidor de certificado o CRL contenga un nombre de directorio que identifique inequívocamente a la autoridad expedidora, este campo se indique como no crítico mediante banderas.

NOTA – Si este campo de extensión está presente y se indica como crítico mediante banderas, el campo **issuer** del certificado o CRL puede contener un nombre nulo (por ejemplo, una secuencia de nombres distinguidos relativos cero), en cuyo caso el expedidor es identificado solamente por el nombre o nombres en esta extensión.

12.3.2.3 Campo de atributos de directorio de sujeto

Este campo transporta cualesquiera valores de atributos de directorio deseados para el sujeto del certificado. Este campo se define como sigue:

```
subjectDirectoryAttributes EXTENSION ::= {
    SYNTAX AttributesSyntax
    IDENTIFIED BY id-ce-subjectDirectoryAttributes }
```

AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF **Attribute**

Esta extensión siempre es no crítica.

12.4 Constricciones de trayecto de certificación

12.4.1 Requisitos

Para el procesamiento del trayecto de certificación:

- Los certificados de la entidad final tienen que ser distinguibles de los certificados de CA, para poder ofrecer protección contra entidades finales que se establecen a sí mismas como CA sin autorización. Es necesario también que sea posible que una CA limite la longitud de una cadena subsiguiente resultante de una CA sujeto certificada, por ejemplo, a no más de un certificado más o a no más de dos certificados más.
- Una CA tiene que ser capaz de especificar constricciones que permitan al usuario del certificado verificar que las CA de menor confianza en un trayecto de certificación (es decir, las CA más abajo en el trayecto de certificación de la CA cuya clave pública comienza el certificado de usuario) no están violando su confianza expidiendo certificados a sujetos en un espacio de nombre inapropiado. El cumplimiento de estas constricciones tiene que ser automáticamente verificable por el usuario del certificado.

c) El procesamiento del trayecto de certificación tiene que poder implementarse en un módulo automatizado e independiente. Esto es necesario para permitir que se implementen módulos de soporte físico o de soporte lógico fiables que ejecuten las funciones de procesamiento del trayecto de certificación.

- d) Debe ser posible efectuar el procesamiento del trayecto de certificación sin depender de interacciones en tiempo real con el usuario local.
- e) Debe ser posible efectuar el procesamiento del trayecto de certificación sin depender de la utilización de las bases de datos locales de confianza que contienen información de descripción de políticas. (Se necesita alguna información local de confianza – una clave pública inicial, como mínimo – para el procesamiento del trayecto de certificación, pero el volumen de esta información se debe minimizar.)
- f) Los trayectos de certificación tienen que funcionar en entornos en los cuales se reconocen múltiples políticas de certificado. Una CA tiene que ser capaz de estipular en cuáles CA de otro dominio confía y para qué fines. Hay que admitir el encadenamiento a través de dominios de múltiples políticas.
- g) Se requiere una completa flexibilidad en los modelos de confianza. Un modelo jerárquico estricto que es adecuado para una organización no es adecuado cuando se consideran las necesidades de múltiples empresas interconectadas. Se requiere flexibilidad para seleccionar la primera CA de confianza en un trayecto de certificación. En particular, debe ser posible requerir que el trayecto de certificación comience en el dominio de seguridad local del sistema usuario de la clave pública.
- h) Las estructuras de denominación no deben estar limitadas por la necesidad de utilizar nombres en certificados, es decir, las estructuras de nombres de directorio consideradas naturales para organizaciones o zonas geográficas no necesitarán ajuste para acomodar requisitos de la autoridad de certificación.
- i) Los campos de extensión de certificado tienen que ser compatibles hacia atrás con el sistema de trayecto de certificación sin constricciones especificado en ediciones anteriores de esta Recomendación | Norma Internacional.
- j) Una CA tiene que ser capaz de inhibir la utilización de correspondencia de políticas y requerir que estén presentes identificadores de políticas de certificados explícitos en certificados subsiguientes en un trayecto de certificación.

NOTA – En cualquier sistema que utiliza certificados, el procesamiento de un trayecto de certificación requiere un nivel de seguridad apropiado. Esta Especificación de directorio define funciones que pueden ser utilizadas en implementaciones que tienen que conformarse con declaraciones de seguridad específicas. Por ejemplo, un requisito de seguridad pudiera indicar que el procesamiento del trayecto de certificación debe estar protegido contra la subversión del proceso (como la alteración del soporte lógico o la modificación de los datos). El nivel de seguridad debe ser conmensurado con el riesgo comercial. Por ejemplo:

- a) se puede requerir el procesamiento interno de un módulo criptográfico apropiado para claves públicas utilizadas con el fin de validar transferencias de fondos de mucho valor; mientras que
- b) el procesamiento en soporte lógico puede ser apropiado para indagaciones de balances bancarios en el hogar.

En consecuencia, las funciones de procesamiento del trayecto de certificación pueden ser adecuadas para su implementación en módulos criptográficos, de soporte físico o testigos criptográficos, como una opción.

12.4.2 Campos de extensión de certificado

Se definen los siguientes campos de extensión:

- a) *Constricciones básicas.*
- b) *Constricciones de nombre.*
- c) *Constricciones de política.*

Estos campos de extensión se utilizarán solamente como extensiones de certificado. Las constricciones de nombres y las constricciones de políticas se utilizarán solamente en certificados de CA; las constricciones básicas se pueden utilizar también en certificados de entidad final. En el anexo L figuran ejemplos de la utilización de estas extensiones.

12.4.2.1 Campo de constricciones básicas

Este campo indica si el sujeto puede actuar como una CA utilizando la clave pública certificada para verificar firmas certificadas. En caso afirmativo, se puede especificar también una restricción de longitud de trayecto de certificación. Este campo se define como sigue:

```
basicConstraints EXTENSION ::= {
    SYNTAX          BasicConstraintsSyntax
    IDENTIFIED BY    id-ce-basicConstraints }
```

```

BasicConstraintsSyntax ::= SEQUENCE {
    cA                                BOOLEAN DEFAULT FALSE,
    pathLenConstraint                INTEGER (0..MAX) OPTIONAL }

```

El componente **cA** indica si la clave pública certificada se puede utilizar para verificar firmas certificadas.

El componente **pathLenConstraint** estará presente solamente si **cA** está puesto a verdadero. Indica el número máximo de certificados de CA que pueden seguir a este certificado en un trayecto de certificación. El valor 0 indica que el sujeto de este certificado puede expedir certificados solamente a entidades finales y no a otras CA. Si no aparece el campo **pathLenConstraint** en cualquier certificado de un trayecto de certificación, no hay límite a la longitud autorizada del trayecto de certificación.

A opción del expedidor del certificado, esta extensión puede ser crítica o no crítica. Se recomienda que se indique como crítica mediante banderas; en los demás casos, una entidad que no esté autorizada para ser una CA puede emitir certificados y un sistema que utiliza certificados puede emplear inadvertidamente tal certificado.

Si esta extensión está presente y se indica como crítica mediante banderas, entonces:

- si el valor de **cA** no está puesto a verdadero, la clave pública certificada no se utilizará para verificar una firma certificada;
- si el valor de **cA** está puesto a verdadero y **pathLenConstraint** está presente, el sistema que utiliza el certificado comprobará que el trayecto de certificación que se procesa concuerda con el valor de **pathLenConstraint**.

NOTA 1 – Si esta extensión no está presente o se indica como no crítica con banderas y no es reconocida por un sistema que utiliza certificados, ese sistema debe emplear otros medios para determinar si se puede usar la clave pública certificada para verificar firmas certificadas.

NOTA 2 – Para limitar a un sujeto de certificado a que sólo sea una entidad final, es decir, no una CA, el expedidor puede incluir este campo de extensión que contiene solamente un valor **SEQUENCE** vacío.

12.4.2.2 Campo de constricciones de nombre

Este campo, que se utilizará solamente en un certificado de CA, indica un espacio de nombre dentro del cual deben estar ubicados todos los nombres de sujetos en certificados subsiguientes en un trayecto de certificación. Este campo se define como sigue:

```

nameConstraints EXTENSION ::= {
    SYNTAX      NameConstraintsSyntax
    IDENTIFIED BY id-ce-nameConstraints }

```

```

NameConstraintsSyntax ::= SEQUENCE {
    permittedSubtrees  [0]      GeneralSubtrees OPTIONAL,
    excludedSubtrees  [1]      GeneralSubtrees OPTIONAL }

```

```

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

```

```

GeneralSubtree ::= SEQUENCE {
    base              GeneralName,
    minimum           [0]      BaseDistance DEFAULT 0,
    maximum           [1]      BaseDistance OPTIONAL }

```

```

BaseDistance ::= INTEGER (0..MAX)

```

Si están presentes, los componentes **permittedSubtrees** y **excludedSubtrees** cada uno especifica uno o más subárboles de denominación, cada definido por el nombre de la raíz del subárbol y, facultativamente dentro de ese subárbol, una zona que está limitada por niveles superiores y/o inferiores. Si **permittedSubtrees** está presente, de todos los certificados expedidos por esa CA y las CA subsiguientes en el trayecto de certificación, sólo los certificados con nombres de sujeto dentro de estos subárboles son aceptables. Si **excludedSubtrees** está presente, cualquier certificado expedido por esa CA o la CA subsiguiente en el trayecto de certificación que tiene un nombre de sujeto dentro de estos subárboles es inaceptable. Si están presentes **permittedSubtrees** y **excludedSubtrees** y los espacios de nombres se superponen, tiene precedencia la declaración de exclusión.

De las formas de nombres disponibles a través del tipo **GeneralName**, sólo se pueden utilizar en estos campos las formas de nombres que tienen una estructura jerárquica bien definida. La forma de nombre **directoryName** satisface este requisito; al utilizar esta forma de nombre, un subárbol de denominación corresponde a un subárbol del DIT. Las implementaciones conformes no tienen que reconocer todas las formas de nombres posibles. Si la extensión se indica como crítica mediante banderas y una implementación que emplea certificados no reconoce una forma de nombre utilizada en cualquier componente de **base**, el certificado se tratará como si se hubiese encontrado una extensión crítica

no reconocida. Si la extensión se indica como no crítica mediante banderas y una implementación que emplea certificados no reconoce una forma de nombre utilizada en cualquier componente de **base**, se puede pasar por alto esa especificación de subárbol. Cuando un sujeto de certificado tiene múltiples nombres de la misma forma de nombre (incluido, en el caso de la forma de nombre **directoryName**, el nombre en el campo de sujeto del certificado si no es nulo), se probará la coherencia de todos estos nombres con una restricción de nombre de esa forma de nombre.

NOTA – Al probar la coherencia de nombres de sujetos de certificados con una restricción de nombre, se deben procesar los nombres en extensiones de nombres alternativos de sujetos no críticos, y no pasarlos por alto.

El campo **minimum** especifica el límite superior de la zona dentro del subárbol. Todos los nombres cuyo componente de nombre final está por encima del nivel especificado no están contenidos dentro de la zona. Un valor de **minimum** igual a cero (el valor por defecto) corresponde a la base, es decir, el nodo superior del subárbol. Por ejemplo, si **minimum** está puesto a uno, el subárbol de denominación excluye el nodo de base pero incluye nodos subordinados.

El campo **maximum** especifica el nivel inferior de la zona dentro del subárbol. Todos los nombres cuyo último componente está por debajo del nivel especificado no están contenidos dentro de la zona. Un valor de **maximum** de cero corresponde a la base, es decir, la parte superior del subárbol. Un componente **maximum** ausente indica que no se debe imponer un límite inferior en la zona dentro del subárbol. Por ejemplo, si **maximum** está puesto a uno, el subárbol de denominación excluye todos los nodos, excepto la base del subárbol y sus subordinados inmediatos.

A opción del expedidor del certificado, esta extensión puede ser crítica o no crítica. Se recomienda que se indique como crítica mediante banderas; de lo contrario, un usuario de certificado puede no comprobar que los certificados subsiguientes en un trayecto de certificación están situados en el espacio de nombre previsto por la CA expedidora.

Si esta extensión está presente y está indicada como crítica mediante banderas, el sistema que utiliza el certificado comprobará que el trayecto de certificación que se procesa concuerda con el valor en esta extensión.

12.4.2.3 Campo de restricciones de políticas

Este campo especifica las restricciones que pueden requerir identificación de política de certificados explícita o inhibir la correspondencia de políticas para el resto del trayecto de certificación. Este campo se define como sigue:

```
policyConstraints EXTENSION ::= {
    SYNTAX      PolicyConstraintsSyntax
    IDENTIFIED BY id-ce-policyConstraints }

PolicyConstraintsSyntax ::= SEQUENCE {
    requireExplicitPolicy      [0] SkipCerts OPTIONAL,
    inhibitPolicyMapping       [1] SkipCerts OPTIONAL }

SkipCerts ::= INTEGER (0..MAX)
```

Si el componente **requireExplicitPolicy** está presente, indica que, en todos los certificados a partir del expedido por una CA denominada en el trayecto de certificación hasta el fin del trayecto de certificación, es necesario que el certificado contenga, en la extensión de políticas de certificado, un identificador de política aceptable. Un identificador de política aceptable es el identificador de la política de certificado requerida por el usuario del trayecto de certificación o el identificador de una política que ha sido declarada equivalente a ésta mediante la correspondencia de políticas. La CA denominada es la CA que es el sujeto del certificado que contiene esta extensión (si el valor de **requireExplicitPolicy** es cero) o una CA que es el sujeto de un certificado subsiguiente en el trayecto de certificación (indicado por un valor no cero).

Si está presente el componente **inhibitPolicyMapping**, indica que en todos los certificados a partir de una CA denominada en el trayecto de certificación hasta el fin del trayecto de certificación, no se permite la correspondencia de políticas. La CA denominada es la CA que es el sujeto del certificado que contiene esta extensión (si el valor de **inhibitPolicyMapping** es cero) o una CA que es el sujeto de un certificado subsiguiente en el trayecto de certificación (indicado por un valor no cero).

Un valor del tipo **SkipCerts** indica el número de certificados en el trayecto de certificación que se han de saltar antes de que una restricción sea efectiva.

A opción del expedidor del certificado, esta extensión puede ser crítica o no crítica. Se recomienda que se indique como crítica mediante banderas; de lo contrario un usuario de certificado puede no interpretar correctamente la estipulación de la CA expedidora.

12.4.3 Procedimiento de procesamiento del trayecto de certificación

El procesamiento del trayecto de certificación se realiza en un sistema que tiene que utilizar la clave pública de una entidad final distante, por ejemplo, un sistema que está verificando una firma digital generada por una entidad distante. Las políticas de certificado, las constricciones básicas, las constricciones de nombre y las extensiones de constricciones de política han sido diseñadas para facilitar la implementación automática e independiente de la lógica del procesamiento del trayecto de certificación.

A continuación figura un esbozo de un procedimiento para validar trayectos de certificación. Una implementación será equivalente funcionalmente al comportamiento externo resultante de este procedimiento. El algoritmo utilizado por una implementación determinada para derivar las salidas correctas a partir de las entradas dadas no está normalizado.

Las entradas al procedimiento de procesamiento de trayecto de certificación son:

- a) un conjunto de certificados que comprenden un trayecto de certificación;
- b) un valor de clave pública o identificador público de confianza (si la clave está almacenada internamente en el módulo de procesamiento del trayecto de certificación), que se ha de utilizar para verificar el primer certificado en el trayecto de certificación;
- c) un *conjunto de políticas inicial* que comprende uno o más identificadores de políticas de certificado, que indican que cualquiera de estas políticas sería aceptable al usuario del certificado para los fines de procesamiento del trayecto de certificación; esta entrada puede tomar también el valor especial *cualquier política*;
- d) un valor de indicador de *política explícita inicial*, que indica si un identificador de política aceptable tiene que aparecer explícitamente en el campo de extensión de políticas de certificado de todos los certificados en el trayecto;
- e) un valor de indicador de *inhibición de correspondencia de políticas inicial*, que indica si la correspondencia de políticas está prohibida en el trayecto de certificación; y
- f) la fecha/hora actual (si no está disponible internamente en el módulo de procesamiento de trayecto de certificación).

Los valores indicados en c), d) y e) dependerán de los requisitos de política de la combinación de aplicaciones de usuario que tiene que utilizar la clave pública de la entidad final certificada.

Las salidas del procedimiento son:

- a) una indicación de éxito o fracaso de la validación del trayecto de certificación;
- b) si la validación fracasa, un código de diagnóstico que indica el motivo del fallo;
- c) si la validación tiene éxito, un conjunto de políticas constreñidas por las CA en el trayecto de certificación, junto con todos los calificadores para estas políticas encontradas en el trayecto de certificación, o el valor especial *cualquier política*. A menos que se devuelva *cualquier política*, el usuario del certificado sólo empleará el certificado de acuerdo con una de las políticas identificadas y procesará todos los calificadores para esa política presentes en el trayecto de certificación;
- d) si la validación tuvo éxito y c) devolvió el valor *cualquier política*, el conjunto de todos los calificadores de elementos de políticas encontrados en el trayecto de certificación;
- e) detalles de cualquier correspondencia de políticas que aparezcan en el procesamiento del trayecto de certificación.

NOTA – Si la validación tiene éxito, el sistema que utiliza certificado puede elegir no utilizar el certificado como resultado de los valores de calificadores de política u otra información en el certificado.

El procedimiento utiliza el siguiente conjunto de variables de estado:

- a) *conjunto de políticas constreñidas del usuario*: Un conjunto de identificadores de políticas de certificado que comprende las políticas actualmente consideradas aceptables para el usuario del certificado; esta variable de estado puede tomar también el valor especial *cualquier política*.
- b) *conjunto de políticas constreñidas por la autoridad*: Un conjunto de identificadores de políticas de certificado que comprende el conjunto de políticas actualmente considerado aceptable para las CA en el trayecto de certificación; esta variable de estado puede tomar también el valor especial *cualquier política*.
- c) *subárboles permitidos*: Un conjunto de especificaciones de subárbol que define subárboles dentro de los cuales deben aparecer todos los nombres de sujetos en certificados subsiguientes en el trayecto de certificación, o pueden tomar el valor especial *ilimitado*.

- d) *subárboles excluidos*: Un conjunto (posiblemente vacío) de especificaciones de subárbol (cada una de las cuales comprende un nombre de base de subárbol e indicadores de nivel máximo y mínimo) que define subárboles dentro de los cuales no puede aparecer ningún nombre de sujeto en un certificado subsiguiente en el trayecto de certificación.
- e) *indicador de política explícito*: Indica si una política aceptable tiene que ser explícitamente identificada en cada certificado.
- f) *indicador de inhibición de correspondencia de políticas*: Indica si se inhibe la correspondencia de políticas.
- g) *constricciones pendientes*: Detalles de constricciones de política explícita y/o de inhibición de correspondencia de políticas que han sido estipuladas pero que no han sido aún aplicadas. Hay dos indicadores de un bit denominados *política explícita pendiente* e *inhibición de correspondencia de políticas pendiente* y también, para cada uno, un entero denominado *salto de certificados* que da el número de certificados que hay que saltar aún antes de que se aplique la restricción.

El procedimiento conlleva un paso de inicialización, seguido por una serie de pasos de procesamiento de certificados. El paso de inicialización comprende:

- a) Inicializar la variable *conjunto de políticas constreñidas del usuario* al valor *conjunto de políticas inicial*.
- b) Inicializar la variable *conjunto de políticas constreñidas de la autoridad* al valor *cualquier política*.
- c) Inicializar la variable *subárboles permitidos a ilimitados*.
- d) Inicializar la variable *subárboles excluidos* a un conjunto vacío.
- e) Inicializar el *indicador de política explícita* al valor *política explícita inicial*.
- f) Inicializar el *indicador de correspondencia de políticas* al valor *inhibición de correspondencia de política inicial*.
- g) Inicializar los dos indicadores *constricciones pendientes* a no fijados.

Cada certificado se procesa en turno, comenzando con el certificado firmado que utiliza la clave pública de entrada. Se considera que el último certificado es el *certificado de fin*; cualesquiera otros certificados se consideran *certificados intermedios*.

Se aplican las siguientes comprobaciones a un certificado:

- a) Comprobar que la firma verifica que las fechas son válidas, que los nombres del sujeto del certificado y del expedidor del certificado encadenan correctamente y que el certificado no ha sido revocado.
- b) Para un certificado intermedio, si está presente en el certificado el campo de extensión de constricciones básico, comprobar que el componente **cA** está presente y está puesto a verdadero. Si el componente **pathLenConstraint** está presente, comprobar que el trayecto de certificación actual no viola esa restricción.
- c) Si el *indicador de política explícita* está fijado y el *conjunto de políticas constreñidas del usuario* no está fijado a *cualquier política*, comprobar que la extensión de políticas del certificado está presente y que por lo menos un miembro del *conjunto de políticas constreñidas del usuario* aparece en el campo de políticas del certificado.
- d) Si está presente la extensión de políticas del certificado y está indicada como crítica mediante banderas, calcular la intersección de las políticas en esa extensión y el *conjunto de políticas constreñidas de la autoridad* y poner el resultado como el nuevo valor de *conjunto de políticas constreñidas de la autoridad*. Comprobar que la intersección de *conjunto de políticas constreñidas de la autoridad* y *conjunto de políticas constreñidas del usuario* no está vacía.
- e) Comprobar que el nombre del sujeto está dentro del espacio de nombre dado por el valor de *subárboles permitidos* y no está dentro del espacio de nombre dado por el valor de *subárboles excluidos*.

Si falla alguna de las comprobaciones anteriores, el procedimiento termina, devolviendo una indicación de fallo y un código de motivo apropiado. Si no falla ninguna de las comprobaciones anteriores en el certificado de fin, el procedimiento termina, devolviendo una indicación de éxito junto con el conjunto de identificadores de política del *conjunto de políticas constreñidas*, los calificadores de elementos de política requeridos y los detalles de cualquier correspondencia de políticas que puedan haber aparecido.

Para un certificado intermedio, se realizan las siguientes acciones de grabación de constricción, para establecer correctamente las variables de estado para el procesamiento del siguiente certificado:

- a) Si la extensión **nameConstraints** con un componente **permittedSubtrees** está presente en el certificado, fijar la variable de estado *subárboles permitidos* a la intersección de su valor anterior y al valor indicado en la extensión del certificado.
- b) Si la extensión **nameConstraints** con un componente **excludedSubtrees** está presente en el certificado, fijar la variable de estado *subárboles excluidos* a la unión de su valor previo y del valor indicado en la extensión del certificado.
- c) Si indicador de *política explícita* no está fijado:
 - si el indicador *política explícita pendiente* está fijado, disminuir el correspondiente valor *salto de certificados* y, si este valor llega a cero, fijar *indicador de política explícita*;
 - si la constricción **requireExplicitPolicy** está presente en el certificado, hacer lo siguiente: para un valor **SkipCerts** de 0, fijar *el indicador de política explícita*; para cualquier otro valor de **SkipCerts**, fijar el indicador *política explícita pendiente* y fijar el correspondiente valor *salto de certificados* al valor **SkipCerts** o al valor anterior de *salto de certificados*, el que sea menor (si el indicador *política explícita pendiente* ya estaba fijado).
- d) Si el *indicador de inhibición de correspondencia de políticas* no está fijado:
 - procesar cualquier extensión de correspondencia de políticas con respecto a políticas en el *conjunto de políticas constreñidas del usuario* y añadir identificadores de política apropiados al *conjunto de políticas constreñidas del usuario*;
 - procesar cualquier extensión de correspondencia de políticas con respecto a políticas en el *conjunto de políticas constreñidas de la autoridad* y añadir identificadores de política apropiada al *conjunto de políticas constreñidas de la autoridad*;
 - si el *indicador inhibición de correspondencia de políticas pendiente* está fijado, disminuir el correspondiente valor de *salto de certificados* y, si este valor llega a cero, fijar el *indicador de inhibición de correspondencia de políticas*;
 - si la constricción **inhibitPolicyMapping** está presente en el certificado, hacer lo siguiente: para un valor de **SkipCerts** de 0, fijar el *indicador de inhibición de correspondencia de políticas*; para cualquier otro valor de **SkipCerts**, fijar el *indicador de inhibición de correspondencia de políticas pendiente*, y fijar el correspondiente valor de *salto de certificados* al valor de **SkipCerts** o al valor anterior de *salto de certificados*, el que sea menor (si el *indicador inhibición de correspondencia de políticas pendiente* estaba ya fijado).

12.5 Extensiones de la CRL básica

12.5.1 Requisitos

Los siguientes requisitos se relacionan con las CRL:

- a) Los usuarios de certificados tienen que ser capaces de seguir todas las CRL expedidas por un expedidor de CRL o punto de distribución CRL (véase 12.6) y ser capaces de detectar una CRL que falte en la secuencia. Por consiguiente se requieren números de secuencia de CRL.
- b) Es posible que algunos usuarios de CRL deseen responder diferentemente a una revocación, dependiendo del motivo de la revocación. Por consiguiente, es necesario que un asiento de CRL indique el motivo de la revocación.
- c) Es necesario que una CA sea capaz de suspender temporalmente la validez de un certificado y después revocarlo o restablecerlo. Entre los posibles motivos para esta acción cabe citar:
 - el deseo de disminuir la responsabilidad de una revocación errónea cuando una petición de revocación no está autenticada y hay información inadecuada para determinar si es válida;
 - otras necesidades comerciales, como la inhabilitación temporal del certificado de una entidad hasta una auditoría o investigación.

- d) Una CRL contiene, para cada certificado revocado, la fecha cuando la CA envió la revocación. Se puede obtener más información sobre cuándo se comprometió una clave real o sospechosa, y esta información puede ser valiosa para el usuario del certificado. La fecha de revocación es insuficiente para solucionar algunas controversias porque, suponiendo lo peor, todas las firmas expedidas durante el periodo de validez del certificado tienen que ser consideradas no válidas. Sin embargo, puede ser importante para un usuario que un documento firmado se reconozca como válido aún cuando la clave utilizada para firmar el mensaje fuese comprometida después de que se produjo la firma. Para facilitar la solución de este problema, un asiento de CRL puede incluir una segunda fecha que indique cuándo se supo o se sospechó que la clave privada estaba comprometida.

12.5.2 Campos de extensión de CRL y de asiento de CRL

Se definen los siguientes campos de extensión:

- a) *Número de CRL.*
- b) *Código de motivo.*
- c) *Código de instrucción de retención.*
- d) *Fecha de no validez.*

El campo de número de CRL se utilizará solamente como un campo de extensión de CRL y los otros campos se utilizarán solamente como campos de extensión de asientos de CRL.

12.5.2.1 Campo de número de CRL

Este campo de extensión de CRL transporta un número de secuencia que aumenta monótonamente para cada CRL emitida por un determinado expedidor de CRL a través de un atributo de directorio de CA dado o punto de distribución CRL. Permite a un usuario de la CRL detectar si las CRL expedidas antes de la que se está procesando fueron también vistas y procesadas. Este campo se define como sigue:

```
cRLNumber EXTENSION ::= {
    SYNTAX      CRLNumber
    IDENTIFIED BY id-ce-cRLNumber }
```

CRLNumber ::= INTEGER (0..MAX)

Esta extensión siempre es no crítica.

12.5.2.2 Campo de código de motivo

Este campo de extensión de asiento de CRL identifica un motivo para la revocación del certificado. El código de motivo puede ser utilizado por aplicaciones para decidir, sobre la base de la política local, cómo reaccionar a revocaciones enviadas. Este campo se define como sigue:

```
reasonCode EXTENSION ::= {
    SYNTAX      CRLReason
    IDENTIFIED BY id-ce-reasonCode }
```

```
CRLReason ::= ENUMERATED {
    unspecified          (0),
    keyCompromise        (1),
    cACompromise         (2),
    affiliationChanged    (3),
    superseded           (4),
    cessationOfOperation (5),
    certificateHold       (6),
    removeFromCRL        (8) }
```

Los siguiente valores de código de motivo indican por qué se revocó un certificado:

- **keyCompromise** se utiliza para revocar un certificado de entidad final; indica que se sabe o se sospecha que la clave privada del sujeto, u otros aspectos del sujeto validados en el certificado, han sido comprometidos;
- **cACompromise** se utiliza para revocar un certificado de CA; indica que se sabe o se sospecha que la clave privada del sujeto, u otros aspectos del sujeto validados en el certificado, han sido comprometidos;

- **affiliationChanged** indica que se ha modificado el nombre del sujeto u otra información en el certificado pero no hay motivos para sospechar que la clave privada ha sido comprometida;
- **superseded** indica que el certificado ha sido abrogado pero que no hay un motivo para sospechar que la clave privada ha sido comprometida;
- **cessationOfOperation** indica que el certificado ya no es necesario para la finalidad para la cual se expidió pero no hay motivos para sospechar que la clave privada ha sido comprometida.

Un certificado puede ser colocado en retención emitiendo un asiento de CRL con un código de motivo **certificateHold**. La notificación de retención del certificado puede incluir un código de instrucción de retención facultativo para transportar información adicional a los usuarios del certificado (véase 12.5.2.3). Cuando se ha emitido una retención, se puede tratar de una de las tres maneras siguientes:

- a) puede permanecer en la CRL sin ninguna otra acción, lo que hace que los usuarios rechacen las transacciones emitidas durante el periodo de retención; o
- b) se la puede sustituir por una revocación (final) para el mismo certificado, en cuyo caso el motivo será uno de los motivos normalizados para la revocación, la fecha de revocación será la fecha en la que el certificado fue colocado en retención, y no aparecerá el campo de extensión de código de instrucción facultativo, o
- c) se la puede liberar explícitamente y suprimir el asiento en la CRL.

El código de motivo **removeFromCRL** se ha de utilizar con las CRL delta (véase 12.6) solamente e indica que un asiento de CRL existente debe ser suprimido debido a expiración del certificado o liberación de la retención. Un asiento con este código de motivo se utilizará en las CRL delta para las cuales la CRL básica correspondiente contiene un asiento para el mismo certificado con el código de motivo **certificateHold**.

Esta extensión siempre es no crítica.

12.5.2.3 Campo de código de instrucción de retención

Este campo de extensión de asientos CRL prevé la inclusión de un identificador de instrucción registrado para indicar la acción que se ha de efectuar al encontrar un certificado retenido. Es aplicable solamente en un asiento que tiene el código de motivo **certificateHold**. Este campo se define como sigue:

holdInstructionCode EXTENSION ::= {
 SYNTAX **HoldInstruction**
 IDENTIFIED BY **id-ce-instructionCode** }

HoldInstruction ::= OBJECT IDENTIFIER

Esta extensión siempre es no crítica. En esta Especificación de directorio no se definen códigos de instrucción de retención normalizados.

NOTA – Como ejemplos de instrucciones de retención cabe citar "por favor, comuníquese con la CA" o "recupere el testigo del usuario".

12.5.2.4 Campo de fecha de no validez

Este campo de extensión de asiento de CRL indica la fecha en la cual se sabe o se sospecha que la clave privada fue comprometida o que el certificado debe considerarse no válido por otros motivos. Esta fecha puede ser anterior a la fecha de revocación en el asiento de CRL, que es la fecha en la cual la CA procesó la revocación. Este campo se define como sigue:

invalidityDate EXTENSION ::= {
 SYNTAX **GeneralizedTime**
 IDENTIFIED BY **id-ce-invalidityDate** }

Esta extensión siempre es no crítica.

NOTA 1 – La fecha en esta extensión no es, por sí misma, suficiente para el no repudio. Por ejemplo, esta fecha puede ser una fecha aconsejada por el tenedor de la clave privada, y es posible que esta persona alegue fraudulentamente que una clave estaba comprometida en algún momento del pasado, para repudiar una firma generada válidamente.

NOTA 2 – Cuando una revocación es enviada por primera vez por una CA en una CRL, la fecha de no validez puede preceder a la fecha de expedición de CRL anteriores. La fecha de revocación no debe preceder a la fecha de expedición de CRL anteriores.

12.6 Puntos de distribución de CRL y CRL delta

12.6.1 Requisitos

Como es posible que las listas de revocación sean grandes e inmanejables, se necesita la posibilidad de representar CRL parciales. Se necesitan diferentes soluciones para dos tipos diferentes de implementaciones que procesan las CRL.

El primer tipo de implementación es una estación individual, posiblemente en un testigo criptográfico adjunto. Es probable que estas implementaciones tengan capacidad limitada de almacenamiento de confianza, si tuvieran alguna. Por consiguiente, todas las CRL deben ser examinadas para determinar si son válidas, y después ver si el certificado es válido. Este procesamiento pudiera ser engorroso si la CRL es larga. Se requiere dividir las CRL para eliminar este problema en estas implementaciones.

El segundo tipo de implementación es un servidor de alta calidad de funcionamiento en el que se procesa un gran volumen de mensajes, por ejemplo, un servidor de procesamiento de transacciones. En este entorno, las CRL se procesan generalmente como una tarea de fondo, donde, después que la CRL es validada, el contenido de la misma se almacena localmente en una representación que acelera su examen, por ejemplo, un bit para cada certificado que indica si ha sido revocado. Esta representación se mantiene en almacenamiento de confianza. Este tipo de servidor requerirá en general CRL actualizadas para un gran número de CA. Como ya tiene una lista de certificados previamente revocados, sólo tiene que extraer una lista de los nuevos certificados revocados. Esta lista, denominada CRL delta, será más pequeña y se requerirán menos recursos para extraerla y procesarla que una CRL completa.

Los siguientes requisitos se relacionan con los puntos de distribución de CRL y las CRL delta:

- a) Para controlar los tamaños de CRL es necesario poder asignar subconjuntos del conjunto de todos los certificados emitidos por una CA a diferentes CRL. Esto se puede lograr asociando cada certificado con un punto de distribución de CRL que sea:
 - un asiento de directorio cuyo atributo contenga un asiento de revocación para ese certificado, si ha sido revocado; o
 - una ubicación, como una dirección de correo electrónico o un identificador de recursos uniformes Internet, a partir del cual se pueda obtener la CRL aplicable.
- b) Por motivos de funcionamiento, es deseable reducir el número de CRL que tienen que ser comprobadas cuando se validan múltiples certificados, por ejemplo, un trayecto de certificación. Esto se puede lograr teniendo un expedidor de CRL que firme y expida las CRL que contienen revocaciones de múltiples CA.
- c) Se necesitan distintas CRL que contengan certificados de CA revocados y certificados de entidad final revocados. Esto facilita el procesamiento del trayecto de certificación porque cabe esperar que las CRL para certificados de CA revocados sean muy cortas (usualmente vacías). Los atributos **authorityRevocationList** y **certificateRevocationList** se han especificado con este fin. Sin embargo, para que esta separación sea segura, es necesario disponer de un indicador en una CRL que identifique qué lista es. En los demás casos, no se podrá detectar la sustitución ilegítima de una lista por otra.
- d) Hay que prever que exista una CRL diferente para posibles situaciones de compromiso (cuando hay un gran riesgo de utilización indebida de claves privadas), además de la que incluye todas las terminaciones vinculantes de rutina (cuando no hay un gran riesgo de uso indebido de claves privadas).
- e) Hay que prever también CRL parciales (conocidas como CRL delta) que sólo contienen asientos para certificados que han sido revocados desde la expedición de una CRL básica.
- f) Si bien los requisitos a) a e) pueden conducir a mecanismos de revocación facultativos, cada CA emitirá, como un enfoque común, CRL completas con todos los certificados que emita.

12.6.2 Campos de extensión de certificado

La extensión de puntos de distribución de CRL se utilizará solamente como una extensión de certificado y puede ser utilizada en certificados de CA y en certificados de entidad final. Este campo identifica el punto o puntos de distribución de CRL a los cuales el usuario del certificado debe hacer referencia para indagar si el certificado ha sido revocado. Un usuario de certificado puede obtener una CRL de un punto de distribución aplicable o puede obtener una CRL completa vigente del asiento de directorio de la CA. Este campo se define como sigue:

```

cRLDistributionPoints EXTENSION ::= {
    SYNTAX      CRLDistPointsSyntax
    IDENTIFIED BY  id-ce-cRLDistributionPoints }

CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint

DistributionPoint ::= SEQUENCE {
    distributionPoint    [0]    DistributionPointName OPTIONAL,
    reasons              [1]    ReasonFlags OPTIONAL,
    cRLIssuer           [2]    GeneralNames OPTIONAL }

DistributionPointName ::= CHOICE {
    fullName              [0]    GeneralNames,
    nameRelativeToCRLIssuer [1]    RelativeDistinguishedName }

ReasonFlags ::= BIT STRING {
    unused                (0),
    keyCompromise         (1),
    cACompromise          (2),
    affiliationChanged    (3),
    superseded            (4),
    cessationOfOperation (5),
    certificateHold       (6) }

```

El componente **distributionPoint** identifica la ubicación de la cual se puede obtener la CRL. Si este componente está ausente, el nombre de punto de distribución es por defecto el nombre del expedidor de la CRL.

Cuando se utiliza la alternativa **fullName** o cuando se aplica el valor por defecto, el nombre del punto de distribución puede tener múltiples formas de nombre. El mismo nombre, por lo menos en una de sus formas de nombre, estará presente en el campo **distributionPoint** de la extensión del punto de distribución de expedición de la CRL. Un sistema que utiliza certificados no tiene que poder procesar todas las formas de nombre. Puede utilizar un punto de distribución a condición de que por lo menos una forma de nombre pueda ser procesada. Si ninguna forma de nombre puede ser procesada para un punto de distribución, un sistema que emplea certificados puede utilizar aún el certificado a condición de que se pueda obtener la información de revocación requerida de otra fuente, por ejemplo, otro punto de distribución o el asiento de directorio de la CA.

El componente **nameRelativeToCRLIssuer** se puede utilizar solamente si el punto de distribución de CRL tiene asignado un nombre de directorio que está directamente subordinado al nombre de directorio del expedidor de CRL. En este caso, el componente **nameRelativeToCRLIssuer** transporta el nombre distinguido relativo con respecto al nombre de directorio del expedidor de la CRL.

El componente **reasons** indica los motivos de revocación contenidos en esta CRL. Si el componente **reasons** está ausente, el correspondiente punto de distribución de CRL distribuye una CRL que contendrá un asiento para este certificado, si este certificado ha sido revocado, con independencia del motivo de revocación. En los demás casos, el valor **reasons** indica los motivos de revocación aducidos por el correspondiente punto de distribución de CRL.

El componente **cRLIssuer** identifica a la autoridad que expide y firma la CRL. Si este componente está ausente, el nombre del expedidor de la CRL es por defecto el nombre del expedidor del certificado.

A opción del expedidor del certificado, esta extensión puede ser crítica o no crítica. En aras de la interoperabilidad, se recomienda que se indique como no crítica mediante banderas. Si la extensión de punto de distribución de CRL es crítica, la CA asegurará que los puntos de distribución aducen códigos de motivo **keyCompromise** y/o **cACompromise**, el que sea aplicable.

Si esta extensión se indica como crítica mediante banderas, un sistema que utiliza certificados no utilizará el certificado sin extraer primero y comprobar una CRL de uno de los puntos de distribución denominados que aduce, por lo menos, el código del motivo **keyCompromise** (para un certificado de entidad final) o **cACompromise** (para un certificado de CA).

Si esta extensión se indica como no crítica mediante banderas y un sistema que utiliza certificados no reconoce el tipo de campo de extensión, el sistema sólo utilizará el certificado si:

- se puede adquirir y comprobar una CRL completa de la CA (el hecho de que la última CRL está completa es indicado por la ausencia de un campo de extensión de punto de distribución expedidor en la CRL);
- no se requiere comprobación de la revocación según la política local; o
- la comprobación de la revocación se realiza por otros medios.

NOTA – Es posible que haya CRL emitidas por más de un expedidor de CRL para el certificado. La coordinación de estos expedidores de CRL y la CA expedidora es un aspecto de la política de la CA.

12.6.3 Campos de extensión de CRL y de asiento de CRL

Se definen los siguientes campos de extensión de CRL o de asiento de CRL:

- a) *Punto de distribución expedidor.*
- b) *Expedidor de certificado.*
- c) *Indicador de CRL delta.*

El punto de distribución expedidor y el indicador de CRL delta se utilizarán solamente como extensiones de CRL. El expedidor de certificado se utilizará solamente como una extensión del asiento de CRL.

12.6.3.1 Campo de punto de distribución expedidor

Este campo de extensión de CRL identifica el punto de distribución de CRL para esta CRL particular, e indica si la CRL está limitada a revocaciones solamente para certificado de entidad final, solamente para certificado de CA, o solamente para un conjunto limitado de motivos. La CRL está firmada por la clave del emisor de la CRL. Los puntos de distribución de CRL no tienen su propio par de claves. Sin embargo, para una CRL distribuida por el directorio, la CRL está almacenada en el asiento del punto de distribución de CRL, que puede no ser el asiento de directorio del expedidor de la CRL. Este campo se define como sigue:

```
issuingDistributionPoint EXTENSION ::= {
    SYNTAX      IssuingDistPointSyntax
    IDENTIFIED BY  id-ce-issuingDistributionPoint }

IssuingDistPointSyntax ::= SEQUENCE {
    distributionPoint      [0] DistributionPointName OPTIONAL,
    onlyContainsUserCerts  [1] BOOLEAN DEFAULT FALSE,
    onlyContainsCACerts    [2] BOOLEAN DEFAULT FALSE,
    onlySomeReasons        [3] ReasonFlags OPTIONAL,
    indirectCRL            [4] BOOLEAN DEFAULT FALSE }
```

El componente **distributionPoint** contiene el nombre del punto de distribución en una o más formas de nombre. Si este campo está ausente, la CRL contendrá asientos para todos los certificados no expirados revocados expedidos por el expedidor de CRL.

Si **onlyContainsUserCerts** es verdadero, la CRL sólo contiene revocaciones para certificados de entidad final. Si **onlyContainsCACerts** es verdadero, la CRL sólo contiene revocaciones para certificados de CA. Si **onlySomeReasons** está presente, la CRL sólo contiene revocaciones para el motivo o motivos identificados, y en los demás casos la CRL contiene revocaciones para todos los motivos.

Si **indirectCRL** es verdadero, la CRL puede contener notificaciones de revocación de otras CA distintas al expedidor de la CRL. La CA particular responsable de cada asiento es la indicada por la extensión de asiento de CRL del expedidor del certificado en ese asiento o de acuerdo con las reglas descritas en 12.6.3.2. En esta CRL, es responsabilidad del expedidor de la CRL asegurar que la CRL está completa porque contiene todos los asientos de revocación, y que concuerda con los indicadores **onlyContainsUserCerts**, **onlyContainsCACerts** y **onlySomeReasons**, de todas las CA que identifican a este expedidor de CRL en sus certificados.

Para las CRL distribuidas por el directorio, se aplican las siguientes reglas sobre la utilización de atributos. Una CRL que tiene **onlyContainsCACerts** fijado será distribuida por el atributo **authorityRevocationList** del punto de distribución asociado o, si no se identifica ningún punto de distribución, por el atributo **authorityRevocationList** del asiento del expedidor de la CRL. En los demás casos, la CRL será distribuida por el atributo **certificateRevocationList** del punto de distribución asociado o, si no se identifica ningún punto de distribución, por el atributo **certificateRevocationList** del asiento de CA.

Esta extensión siempre es crítica. Un usuario de certificado que no comprende esta extensión no puede suponer que la CRL contiene una lista completa de certificados revocados de la CA identificada. Las CRL que no contienen extensiones críticas deben contener todos los asientos de CRL vigentes para la CA expedidora, incluidos los asientos para todos los certificados de usuario y certificados de CA revocados.

NOTA 1 – Los medios por los cuales la información de revocación es comunicada por las CA a los expedidores de CRL está fuera del alcance de esta Recomendación | Norma Internacional.

NOTA 2 – Si una CA expide, de su propio asiento de directorio (es decir, no de un punto de distribución de CRL denominado separadamente), una CRL con **onlyContainsUserCerts** o **onlyContainsCACerts** fijados, la CA debe asegurar que todos los certificados cubiertos por esta CRL contienen la extensión **basicConstraints**.

12.6.3.2 Campo de expedidor de certificado

Esta extensión de asiento de CRL identifica al expedidor del certificado asociado con un asiento en una CRL indirecta, es decir, una CRL que tiene el indicador **indirectCRL** fijado en su extensión de punto de distribución expedidor. Si esta extensión no está presente en el primer asiento en una CRL indirecta, el expedidor de certificado es por defecto el expedidor de la CRL. En los asientos siguientes en una CRL indirecta, si esta extensión no está presente, el expedidor del certificado para el asiento es el mismo que para el asiento precedente. Este campo se define como sigue:

certificateIssuer EXTENSION ::= {

SYNTAX **GeneralNames**

IDENTIFIED BY **id-ce-certificateIssuer }**

Esta extensión siempre es crítica. Si una implementación pasa por alto esta extensión, no podrá atribuir correctamente asientos de CRL a los certificados.

12.6.3.3 Campo de indicador de CRL delta

Este campo de extensión de CRL identifica que una CRL es solamente una CRL delta. Este campo se define como sigue:

deltaCRLIndicator EXTENSION ::= {

SYNTAX **BaseCRLNumber**

IDENTIFIED BY **id-ce-deltaCRLIndicator }**

BaseCRLNumber ::= CRLNumber

El valor del tipo **BaseCRLNumber** identifica al número de CRL de la CRL básica que se utilizó en el punto de partida de la generación de esta CRL delta, es decir, esta CRL delta contiene los cambios entre la CRL básica y la CRL completa emitidas junto con esta CRL delta. La CA debe decidir si ha de proporcionar las CRL delta. Sin embargo, no se expedirá una CRL delta sin que se haya expedido una CRL completa correspondiente al mismo tiempo. El valor del número de CRL transportado en el campo de extensión de número de CRL (si está presente) será idéntico para la CRL delta y la correspondiente CRL completa.

El usuario de CRL que construye una CRL mantenida localmente a partir de las CRL delta considerará que la CRL construida está incompleta y es inutilizable si las dos condiciones siguientes son verdaderas:

- el valor del **CRLNumber** de la CRL delta recibida es un número mayor que el **CRLNumber** de la última CRL delta procesada; y
- el valor de **BaseCRLNumber** de la CRL delta recibida ha cambiado con respecto al **BaseCRLNumber** de la última CRL delta procesada.

Esta extensión siempre es crítica. Un usuario de certificado que no comprende la utilización de las CRL delta no debe emplear una CRL que contenga esta extensión, pues la CRL puede no estar completa como el usuario espera. Las CRL que no contienen extensiones críticas deben contener todos los asientos de CRL vigentes para la CA expedidora, incluidos los asientos para todos los certificados de usuario y los certificados de CA revocados.

NOTA – La CA debe decidir si distribuye o no las CRL expedidas entre dos CRL básicas. Por ejemplo, puede ser política de la CA distribuir las CRL básicas por CD-ROM y las CRL delta por un servicio en línea como es el directorio. Aunque la CA haya expedido su CRL con la CRL delta asociada, puede ser política de las CA que el usuario construya la CRL vigente aplicando la CRL delta a la CRL básica mantenida en el CD-ROM.

12.6.4 Tipo de atributo para las CRL delta

Se define el siguiente tipo de atributo para mantener una CRL delta en un asiento de directorio de la CA.

```
deltaRevocationList ATTRIBUTE ::= {
  WITH SYNTAX      CertificateList
  EQUALITY MATCHING RULE certificateListExactMatch
  ID               id-at-deltaRevocationList }
```

12.7 Reglas de concordancia

Esta Especificación de directorio define reglas de concordancia para utilizarlas con atributos del tipo **Certificate**, **CertificatePair**, **CertificateList**, y **SupportedAlgorithm**, respectivamente.

12.7.1 Concordancia exacta de certificados

La regla de concordancia exacta de certificados compara la equivalencia de un valor presentado con un valor de atributo del tipo **Certificate**. Selecciona únicamente un certificado.

```
certificateExactMatch MATCHING-RULE ::= {
  SYNTAX      CertificateExactAssertion
  ID          id-mr-certificateExactMatch }

CertificateExactAssertion ::= SEQUENCE {
  serialNumber      CertificateSerialNumber,
  issuer            Name }
```

Esta regla de concordancia devuelve VERDADERO si los componentes en el valor de atributo concuerdan con los del valor presentado.

12.7.2 Concordancia de certificados

La regla de concordancia de certificados compara un valor presentado con un valor de atributo del tipo **Certificate**. Selecciona uno o más certificados sobre la base de diversas características.

```
certificateMatch MATCHING-RULE ::= {
  SYNTAX      CertificateAssertion
  ID          id-mr-certificateMatch }

CertificateAssertion ::= SEQUENCE {
  serialNumber      [0] CertificateSerialNumber    OPTIONAL,
  issuer            [1] Name                        OPTIONAL,
  subjectKeyIdentifier [2] SubjectKeyIdentifier      OPTIONAL,
  authorityKeyIdentifier [3] AuthorityKeyIdentifier  OPTIONAL,
  certificateValid    [4] Time                      OPTIONAL,
  privateKeyValid     [5] GeneralizedTime           OPTIONAL,
  subjectPublicKeyAlgID [6] OBJECT IDENTIFIER        OPTIONAL,
  keyUsage            [7] KeyUsage                  OPTIONAL,
  subjectAltName       [8] AltNameType               OPTIONAL,
  policy              [9] CertPolicySet              OPTIONAL,
  pathToName          [10] Name                     OPTIONAL }
```

```
AltNameType ::= CHOICE {
  builtinNameForm  ENUMERATED {
    rfc822Name      (1),
    dNSName         (2),
    x400Address     (3),
    directoryName   (4),
    ediPartyName    (5),
    uniformResourceIdentifier (6),
    iPAddress       (7),
    registeredId    (8) },
  otherNameForm    OBJECT IDENTIFIER }
```

```
CertPolicySet ::= SEQUENCE (1..MAX) OF CertPolicyId
```

Esta regla de concordancia devuelve VERDADERO si todos los componentes que están presentes en el valor presentado concuerdan con los componentes correspondientes del valor de atributo, como sigue:

- a) **serialNumber** concuerda si el valor de este componente en el valor de atributo equivale al del valor presentado;

- b) **issuer** concuerda si el valor de este componente en el valor de atributo equivale al del valor presentado;
- c) **subjectKeyIdentifier** concuerda si el valor de este componente en el valor de atributo almacenado equivale al del valor presentado; no hay concordancia si el valor de atributo almacenado no contiene la extensión de identificador de clave de sujeto;
- d) **authorityKeyIdentifier** concuerda si el valor de este componente en el valor de atributo almacenado equivale al del valor presentado; no hay concordancia si el valor de atributo almacenado no contiene extensión de identificador de clave de autoridad o si no todos los componentes en el valor presentado están presentes en el valor de atributo almacenado;
- e) **certificateValid** concuerda si el valor presentado está dentro del periodo de validez del valor de atributo almacenado;
- f) **privateKeyValid** concuerda si el valor presentado está dentro del periodo indicado por la extensión de periodo de utilización de clave privada del valor de atributo almacenado o si no hay extensión de periodo de utilización de clave privada en el valor de atributo almacenado;
- g) **subjectPublicKeyAlgID** concuerda si es igual al componente **algorithm** del **algorithmIdentifier** del componente **subjectPublicKeyInformation** del valor de atributo almacenado;
- h) **keyUsage** concuerda si todos los bits fijados en el valor presentado están también fijados en la extensión de utilización de clave en el valor de atributo almacenado, o si no hay extensión de utilización de clave en el valor de atributo almacenado;
- i) **subjectAltName** concuerda si el valor de atributo almacenado contiene la extensión de nombre alternativo de sujeto con un componente **AltNames** del mismo tipo de nombre indicado en el valor presentado;
- j) **policy** concuerda si todos los elementos de política identificados en uno de los valores presentados están contenidos en el conjunto de **policyElementIds** en cualquiera de los valores **policyInformation** en la extensión de políticas de certificado en el valor de atributo almacenado; no hay concordancia si no hay extensión de políticas de certificado en el valor de atributo almacenado;
- k) **pathToName** concuerda a menos que el certificado tenga una extensión de constricciones de nombre que inhibe la construcción de un trayecto de certificación al valor de nombre presentado.

12.7.3 Concordancia exacta de pares de certificados

Esta regla de concordancia exacta de pares de certificado compara la equivalencia de un valor presentado con un valor de atributo del tipo **CertificatePair**. Selecciona únicamente un par de certificados cruzados.

```
certificatePairExactMatch MATCHING-RULE ::= {
    SYNTAX      CertificatePairExactAssertion
    ID          id-mr-certificatePairExactMatch }

CertificatePairExactAssertion ::= SEQUENCE {
    forwardAssertion  [0] CertificateExactAssertion OPTIONAL,
    reverseAssertion  [1] CertificateExactAssertion OPTIONAL }
( WITH COMPONENTS    {..., forwardAssertion PRESENT} |
  WITH COMPONENTS    {..., reverseAssertion PRESENT} )
```

Esta regla de concordancia devuelve VERDADERO si los componentes que están presentes en los componentes **forwardAssertion** y **reverseAssertion** del valor presentado concuerdan con los componentes correspondientes de los componentes **forward** y **reverse**, respectivamente, en el valor de atributo almacenado.

12.7.4 Concordancia de pares de certificados

La regla de concordancia de pares de certificados compara un valor presentado con un valor de atributo del tipo **CertificatePair**. Selecciona uno o más pares de certificados sobre la base de diversas características del certificado directo o inverso del par.

```
certificatePairMatch MATCHING-RULE ::= {
    SYNTAX      CertificatePairAssertion
    ID          id-mr-certificatePairMatch }
```

```

CertificatePairAssertion ::= SEQUENCE {
    forwardAssertion  [0] CertificateAssertion OPTIONAL,
    reverseAssertion  [1] CertificateAssertion OPTIONAL }
( WITH COMPONENTS      {..., forwardAssertion PRESENT} |
  WITH COMPONENTS      {..., reverseAssertion PRESENT} )

```

Esta regla de concordancia devuelve VERDADERO si todos los componentes que están presentes en los componentes **forwardAssertion** y **reverseAssertion** del valor presentado concuerdan con los correspondientes componentes de los componentes **forward** y **reverse**, respectivamente, en el valor de atributo almacenado, utilizando las reglas indicadas en 12.7.2.

12.7.5 Concordancia exacta de listas de certificados

La regla de concordancia exacta de listas de certificados compara la equivalencia de un valor presentado con un valor de atributo del tipo **CertificateList**. Selecciona únicamente una CRL.

```

certificateListExactMatch MATCHING-RULE ::= {
    SYNTAX      CertificateListExactAssertion
    ID          id-mr-certificateListExactMatch }

```

```

CertificateListExactAssertion ::= SEQUENCE {
    issuer          Name,
    thisUpdate      Time,
    distributionPoint DistributionPointName OPTIONAL }

```

La regla devuelve VERDADERO si los componentes en el valor de atributo almacenado concuerdan con los del valor presentado. Si el componente **distributionPoint** está presente, debe concordar por lo menos con una forma de nombre.

12.7.6 Concordancia de listas de certificados

La regla de concordancia de listas de certificados compara un valor presentado con un valor de atributo del tipo **CertificateList**. Selecciona una o más CRL sobre la base de diversas características.

```

certificateListMatch MATCHING-RULE ::= {
    SYNTAX      CertificateListAssertion
    ID          id-mr-certificateListMatch }

```

```

CertificateListAssertion ::= SEQUENCE {
    issuer          Name      OPTIONAL,
    minCRLNumber    [0] CRLNumber OPTIONAL,
    maxCRLNumber    [1] CRLNumber OPTIONAL,
    reasonFlags      ReasonFlags OPTIONAL,
    dateAndTime      Time      OPTIONAL,
    distributionPoint [2] DistributionPointName OPTIONAL }

```

La regla de concordancia devuelve VERDADERO si todos los componentes que están presentes en el valor presentado concuerdan con los componentes correspondientes del valor de atributo almacenado, como sigue:

- issuer** concuerda si el valor de este componente en el valor de atributo equivale al del valor presentado;
- minCRLNumber** concuerda si este valor es menor o igual al valor en la extensión de número de CRL del valor de atributo almacenado; no hay concordancia si el valor de atributo almacenado no contiene extensión de número de CRL;
- maxCRLNumber** concuerda si su valor es mayor o igual al valor en la extensión de número de CRL del valor de atributo almacenado; no hay concordancia si el valor de atributo almacenado no contiene extensión de número de CRL;
- reasonFlags** concuerda si cualquiera de los bits que están fijados en el valor presentado están fijados también en los componentes **onlySomeReasons** de la extensión de punto de distribución expedidor del valor de atributo almacenado; no hay concordancia si el valor de atributo almacenado no contiene extensión de punto de distribución expedidor;
- dateAndTime** concuerda si el valor es igual o superior al valor en el componente **thisUpdate** del valor de atributo almacenado y es inferior al valor en el componente **nextUpdate** del valor de atributo almacenado; no hay concordancia si el valor de atributo almacenado no contiene el componente **nextUpdate**;

- f) **distributionPoint** concuerda si el valor de atributo almacenado contiene una extensión de punto de distribución expedidor y el valor de este componente en el valor presentado equivale al valor correspondiente, por lo menos en una forma de nombre, en esa extensión.

12.7.7 Concordancia de identificadores de algoritmo

La regla de concordancia de identificadores de algoritmo compara la equivalencia de un valor presentado con un valor de atributo del tipo **SupportedAlgorithm**.

```
algorithmIdentifierMatch MATCHING-RULE ::= {
  SYNTAX      AlgorithmIdentifier
  ID          id-mr-algorithmIdentifierMatch }
```

La regla devuelve VERDADERO si el valor presentado equivale al componente **algorithmIdentifier** del valor de atributo almacenado.

13 Obtención de atributos certificados

Las identidades de usuario están vinculadas a sus certificados de clave pública para fines de autenticación e identificación. Cada certificado de clave pública transporta la información para realizar ciertas funciones criptográficas. Los atributos certificados asociados con un sujeto, tales como autorizaciones, pueden ser transportados en una estructura separada, definida como un certificado de atributo (**AttributeCertificate**).

13.1 Certificados de atributos

Un certificado de atributos es una estructura distinta de un certificado de clave pública X.509 del sujeto. Un sujeto puede tener múltiples certificados de atributos asociados con cada uno de sus certificados de clave pública. No es necesario que la misma autoridad de certificación cree el certificado de clave pública y el certificado o certificados de atributos para un usuario; de hecho, la separación de tareas frecuentemente impondrá otra cosa. Sin embargo, al definir la política local se deben considerar las repercusiones de realizar múltiples validaciones del trayecto de certificado. Los intercambios entre un sujeto y la autoridad de certificación que trata la generación de certificados de atributos están fuera del alcance de esta Recomendación | Norma Internacional.

AttributeCertificate ::= SIGNED {**AttributeCertificateInfo**}

```
AttributeCertificateInfo ::= SEQUENCE {
  version    Version DEFAULT v1,
  subject    CHOICE {
    baseCertificateID [0] IssuerSerial, -- associated with a Public Key Certificate
    subjectName       [1] GeneralNames }, -- associated with a name
  issuer      GeneralNames, -- CA issuing the attribute certificate
  signature   AlgorithmIdentifier,
  serialNumber CertificateSerialNumber,
  attrCertValidityPeriod AttCertValidityPeriod,
  attributes  SEQUENCE OF Attribute,
  issuerUniqueID UniqueIdentifier OPTIONAL,
  extensions  Extensions OPTIONAL }
```

```
IssuerSerial ::= SEQUENCE {
  issuer      GeneralNames,
  serial      CertificateSerialNumber,
  issuerUID   UniqueIdentifier OPTIONAL }
```

```
AttCertValidityPeriod ::= SEQUENCE {
  notBeforeTime GeneralizedTime,
  notAfterTime  GeneralizedTime }
```

El número de **version** diferencia entre distintas versiones del certificado de atributos. Para esta Recomendación | Norma Internacional, el número de versión será **v1**.

El campo **subject** transporta la identidad del sujeto del certificado, referenciada por el **número de serie** asociado con el certificado de clave pública X.509 del sujeto o por el **subjectName**.

El campo **issuer** es el nombre de la autoridad de certificación que creó el certificado de atributo.

El campo **serialNumber** es el número de serie que identifica de manera única al certificado de atributo.

El campo **signature** identifica el algoritmo criptográfico utilizado para firmar digitalmente el certificado de atributo.

El campo **attCertValidityPeriod** transporta el periodo de tiempo durante el cual se considera que el certificado de atributo es válido, expresado en formato de tiempo generalizado.

El campo **attributes** contiene cualquier atributo de directorio (no restringido a los atributos almacenados dentro de un sistema de directorio, sino que debe ser reconocido dentro del dominio) relacionado con el sujeto.

IssuerUniqueID identifica de manera única al expedidor del certificado de atributo en aquellos casos en que el nombre del expedidor no es suficiente.

El campo **extensions** permite añadir nuevos campos al certificado de atributo.

13.2 Atributo certificado de atributos

Se puede utilizar un certificado de atributo para certificar ciertos valores de atributo y vincularlos a un certificado X.509 o directamente a un nombre de sujeto.

```
attributeCertificateAttribute ATTRIBUTE ::= {
    WITH SYNTAX                AttributeCertificate
    EQUALITY MATCHING-RULE     certificateExactMatch
    ID                         id-at-attributeCertificate }
```

13.3 Regla de concordancia de certificado de atributos

La regla de concordancia de certificado de atributo compara un valor presentado con un valor de atributo del **tipo certificado de atributo**. Estas reglas de concordancia permiten una concordancia más compleja que la **concordancia exacta de certificados**.

```
attributeCertificateMatch MATCHING-RULE ::= {
    SYNTAX AttributeCertificateAssertion
    ID      id-mr-attributeCertificateMatch }

AttributeCertificateAssertion ::= SEQUENCE {
    subject
    [0] CHOICE {
        baseCertificateID [0] IssuerSerial,
        subjectName       [1] GeneralNames} OPTIONAL,
    issuer [1] GeneralNames OPTIONAL,
    attCertValidity [2] GeneralizedTime OPTIONAL,
    attType [3] SET OF AttributeType OPTIONAL}
    -- At least one component of the sequence must be present
```

La regla de concordancia devuelve VERDADERO si todos los componentes que están presentes en el valor presentado concuerdan con los correspondientes componentes del valor de atributo, como sigue:

- baseCertificateID** concuerda si es igual al componente **IssuerSerial** del valor de atributo almacenado;
- subjectName** concuerda si el valor de atributo almacenado contiene la extensión de nombre con el mismo tipo de nombre indicado en el valor presentado;
- issuer** concuerda si el valor de atributo almacenado contiene el componente de nombre del mismo tipo de nombre indicado en el valor presentado;
- attCertValidity** concuerda si está dentro del periodo de validez especificado del valor de atributo almacenado;
- para cada **attType** en el valor presentado, hay un atributo de ese tipo presente en el componente **attributes** del valor almacenado.

13.4 Trayectos de certificados de atributos

Un usuario puede obtener uno o más certificados de atributo de una o más autoridades de certificados. Cada certificado de atributo lleva el nombre de la autoridad de certificado que lo expidió. El siguiente tipo de datos ASN.1 se puede utilizar para representar un trayecto de certificado de atributo:

```
AttributeCertificationPath ::= SEQUENCE {
    attributeCertificate AttributeCertificate,
    acPath SEQUENCE OF ACPPathData OPTIONAL }
```

```

ACPathData ::= SEQUENCE {
    certificate      [0] Certificate OPTIONAL,
    attributeCertificate [1] AttributeCertificate OPTIONAL }

```

13.5 Lista de revocación de certificados de atributos

Los certificados de atributos pueden ser revocados antes de su expiración si los atributos ya no son pertinentes. Los certificados de atributos pueden ser revocados sin afectar al certificado de la clave pública del usuario. Las listas de revocación se mantienen dentro de los asientos como un atributo del tipo "**AttributeCertificateRevocationList**". Este tipo de atributo se define como sigue:

```

attributeCertificateRevocationList    ATTRIBUTE ::= {
    WITH SYNTAX    CertificateList
    ID              id-at-attributeCertificateRevocationList}

```

NOTA – Todas las enmiendas utilizadas para las listas de revocación de certificados de la versión 2 son aplicables a las listas de revocación de certificados de atributo.

Anexo A

Marco de autenticación en ASN.1

(Este anexo es parte integrante de esta Recomendación | Norma Internacional)

Este anexo incluye todas las definiciones de tipo, valor y clase de objeto de información ASN.1 contenidas en esta especificación de directorio, en la forma de dos módulos ASN.1 **AuthenticationFramework** y **CertificateExtensions**.

AuthenticationFramework {joint-iso-ccitt ds(5) module(1) authenticationFramework(7) 3}

DEFINITIONS ::=

BEGIN

-- EXPORTS All --

-- The types and values defined in this module are exported for use in the other ASN.1 modules contained
 -- within the Directory Specifications, and for the use of other applications which will use them to access
 -- Directory services. Other applications may use them for their own purposes, but this will not constrain
 -- extensions and modifications needed to maintain or improve the Directory service.

IMPORTS

id-at, informationFramework, upperBounds, selectedAttributeTypes, basicAccessControl,
 certificateExtensions

FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 3}

Name, ATTRIBUTE

FROM InformationFramework informationFramework

ub-user-password

FROM UpperBounds upperBounds

AuthenticationLevel

FROM BasicAccessControl basicAccessControl

UniqueIdentifier, octetStringMatch

FROM SelectedAttributeTypes selectedAttributeTypes

certificateExactMatch, certificatePairExactMatch, certificateListExactMatch

FROM CertificateExtensions certificateExtensions ;

-- basic certificate definition

```
Certificate ::= SIGNED { SEQUENCE {
  version          [0] Version DEFAULT v1,
  serialNumber     CertificateSerialNumber,
  signature         AlgorithmIdentifier,
  issuer           Name,
  validity         Validity,
  subject          Name,
  subjectPublicKeyInfo SubjectPublicKeyInfo,
  issuerUniqueIdentifier [1] IMPLICIT UniqueIdentifier OPTIONAL,
                      -- if present, version must be v2 or v3
  subjectUniqueIdentifier [2] IMPLICIT UniqueIdentifier OPTIONAL
                      -- if present, version must be v2 or v3
  extensions       [3] Extensions OPTIONAL
                      -- If present, version must be v3 -- } }
```

Version ::= INTEGER { v1(0), v2(1), v3(2) }

CertificateSerialNumber ::= INTEGER

AlgorithmIdentifier ::= SEQUENCE {

algorithm ALGORITHM.&id ({SupportedAlgorithms}),

parameters ALGORITHM.&Type ({SupportedAlgorithms}{ @algorithm}) OPTIONAL }

-- Definition of the following information object set is deferred, perhaps to standardized
 -- profiles or to protocol implementation conformance statements. The set is required to
 -- specify a table constraint on the parameters component of AlgorithmIdentifier.

SupportedAlgorithms ALGORITHM ::= { ... }

Validity ::= SEQUENCE {
 notBefore,

Time
 notAfter Time }

SubjectPublicKeyInfo ::= SEQUENCE {
 algorithm AlgorithmIdentifier,
 subjectPublicKey BIT STRING }

Time ::= CHOICE {
 utcTime UTCTime,
 generalizedTime GeneralizedTime }

Extensions ::= SEQUENCE OF Extension

-- For those extensions where ordering of individual extensions within the SEQUENCE is significant, the
-- specification of those individual extensions shall include the rules for the significance of the order therein

Extension ::= SEQUENCE {
 extnId EXTENSION.&id ({ExtensionSet}),
 critical BOOLEAN DEFAULT FALSE,
 extnValue OCTET STRING
 -- contains a DER encoding of a value of type &ExtnType
 -- for the extension object identified by extnId -- }

ExtensionSet EXTENSION ::= { ... }

EXTENSION ::= CLASS {
 &id OBJECT IDENTIFIER UNIQUE,
 &ExtnType }

WITH SYNTAX {
 SYNTAX &ExtnType
 IDENTIFIED BY &id }

-- other certificate constructs

Certificates ::= SEQUENCE {
 userCertificate Certificate,
 certificationPath ForwardCertificationPath OPTIONAL}

ForwardCertificationPath ::= SEQUENCE OF CrossCertificates

CrossCertificates ::= SET OF Certificate

CertificationPath ::= SEQUENCE {
 userCertificate Certificate,
 theCACertificates SEQUENCE OF CertificatePair OPTIONAL}

CertificatePair ::= SEQUENCE {
 forward [0] Certificate OPTIONAL,
 reverse [1] Certificate OPTIONAL
 -- at least one of the pair shall be present -- }

-- Certificate Revocation List (CRL)

CertificateList ::= SIGNED { SEQUENCE {
 version Version OPTIONAL,
 -- if present, version must be v2
 signature AlgorithmIdentifier,
 issuer Name,
 thisUpdate Time,
 nextUpdate Time OPTIONAL,
 revokedCertificates SEQUENCE OF SEQUENCE {
 userCertificate CertificateSerialNumber,
 revocationDate Time,
 crlEntryExtensions Extensions OPTIONAL } OPTIONAL,
 crlExtensions [0] Extensions OPTIONAL }}

-- attribute certificate

AttributeCertificationPath ::= SEQUENCE {
 attributeCertificate AttributeCertificate,
 acPath SEQUENCE OF ACPPathData OPTIONAL }

```

ACPathData ::= SEQUENCE {
    certificate      [0] Certificate OPTIONAL,
    attributeCertificate [1] AttributeCertificate OPTIONAL }

attributeCertificate ATTRIBUTE ::= {
    WITH SYNTAX      AttributeCertificate
    EQUALITY MATCHING-RULE attributeCertificateMatch
    ID               id-at-attributeCertificate }

AttributeCertificate ::= SIGNED {AttributeCertificateInfo}

AttributeCertificateInfo ::= SEQUENCE {
    version          Version DEFAULT v1,
    subject CHOICE {
        baseCertificateID [0] IssuerSerial, -- associated with a Public Key Certificate
        subjectName       [1] GeneralNames }, -- associated with a name
    issuer           GeneralNames, -- CA issuing the attribute certificate
    signature        AlgorithmIdentifier,
    serialNumber     CertificateSerialNumber,
    attCertValidityPeriod AttCertValidityPeriod,
    attributes       SEQUENCE OF Attribute,
    issuerUniqueID   UniqueIdentifier OPTIONAL,
    extensions       Extensions OPTIONAL }

IssuerSerial ::= SEQUENCE {
    issuer          GeneralNames,
    serial          CertificateSerialNumber,
    issuerUID       UniqueIdentifier OPTIONAL}

AttCertValidityPeriod ::= SEQUENCE{
    notBeforeTime   GeneralizedTime,
    notAfterTime    GeneralizedTime }

attributeCertificateMatch MATCHING-RULE ::= {
    SYNTAX      AttributeCertificateAssertion
    ID         id-mr-attributeCertificateMatch }

AttributeCertificateAssertion ::= SEQUENCE {
    subject
    [0] CHOICE {
        baseCertificateID [0] IssuerSerial,
        subjectName       [1] Name} OPTIONAL,
    issuer [1] Name OPTIONAL,
    attCertValidity
    [2] GeneralizedTime OPTIONAL,
    attType [3] SET OF AttributeType OPTIONAL}
-- At least one component of the sequence must be present
-- attribute types --

userPassword ATTRIBUTE ::= {
    WITH SYNTAX      OCTET STRING (SIZE (0..ub-user-password))
    EQUALITY MATCHING RULE octetStringMatch
    ID               id-at-userPassword }

userCertificate ATTRIBUTE ::= {
    WITH SYNTAX      Certificate
    EQUALITY MATCHING RULE certificateExactMatch
    ID               id-at-userCertificate}

cACertificate ATTRIBUTE ::= {
    WITH SYNTAX      Certificate
    EQUALITY MATCHING RULE certificateExactMatch
    ID               id-at-cACertificate }

crossCertificatePair ATTRIBUTE ::= {
    WITH SYNTAX      CertificatePair
    EQUALITY MATCHING RULE certificatePairExactMatch
    ID               id-at-crossCertificatePair }

authorityRevocationList ATTRIBUTE ::= {
    WITH SYNTAX      CertificateList
    EQUALITY MATCHING RULE certificateListExactMatch
    ID               id-at-authorityRevocationList }

```

```

certificateRevocationList ATTRIBUTE ::= {
    WITH SYNTAX    CertificateList
    EQUALITY MATCHING RULE certificateListExactMatch
    ID             id-at-certificateRevocationList }

attributeCertificateRevocationList ATTRIBUTE ::= {
    WITH SYNTAX    CertificateList
    ID             id-at-attributeCertificateRevocationList}
-- information object classes --

ALGORITHM ::= TYPE-IDENTIFIER
-- parameterized types --

ENCRYPTED-HASH { ToBeSigned } ::= BIT STRING ( CONSTRAINED BY {
    -- must be the result of applying a hashing procedure to the DER-encoded octets --
    -- of a value of -- ToBeSigned -- and then applying an encipherment procedure to those octets -- })

ENCRYPTED { ToBeEnciphered } ::= BIT STRING ( CONSTRAINED BY {
    -- must be the result of applying an encipherment procedure --
    -- to the BER-encoded octets of a value of -- ToBeEnciphered})

SIGNATURE { ToBeSigned } ::= SEQUENCE {
    algorithmIdentifier AlgorithmIdentifier,
    encrypted           ENCRYPTED-HASH { ToBeSigned }}

SIGNED { ToBeSigned } ::= SEQUENCE {
    toBeSigned          ToBeSigned,
    COMPONENTS OF      SIGNATURE { ToBeSigned }}
-- object identifier assignments --

id-at-userPassword      OBJECT IDENTIFIER ::= {id-at 35}
id-at-userCertificate   OBJECT IDENTIFIER ::= {id-at 36}
id-at-cACertificate     OBJECT IDENTIFIER ::= {id-at 37}
id-at-authorityRevocationList OBJECT IDENTIFIER ::= {id-at 38}
id-at-certificateRevocationList OBJECT IDENTIFIER ::= {id-at 39}
id-at-crossCertificatePair OBJECT IDENTIFIER ::= {id-at 40}
id-at-attributeCertificate OBJECT IDENTIFIER ::= {id-at 58}

END

-- Certificate Extensions module

CertificateExtensions {joint-iso-ccitt ds(5) module(1) certificateExtensions(26) 0}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- EXPORTS ALL --

IMPORTS
    id-at, id-ce, id-mr, informationFramework, authenticationFramework,
        selectedAttributeTypes, upperBounds
    FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1)
        usefulDefinitions(0) 3}
    Name, RelativeDistinguishedName, ATTRIBUTE, Attribute,
        MATCHING-RULE FROM InformationFramework informationFramework
    CertificateSerialNumber, CertificateList, AlgorithmIdentifier,
        EXTENSION
    FROM AuthenticationFramework authenticationFramework
    DirectoryString
    FROM SelectedAttributeTypes selectedAttributeTypes
    ub-name
    FROM UpperBounds upperBounds
    ORAddress
    FROM MTSAbstractService {joint-iso-ccitt mhs(6) mts(3)
        modules(0) mts-abstract-service(1) version-1994 (0) } ;
-- Unless explicitly noted otherwise, there is no significance to the ordering
-- of components of a SEQUENCE OF construct in this Specification.
-- Key and policy information extensions --

```

```

authorityKeyIdentifier EXTENSION ::= {
    SYNTAX      AuthorityKeyIdentifier
    IDENTIFIED BY id-ce-authorityKeyIdentifier }

AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier          [0] KeyIdentifier          OPTIONAL,
    authorityCertIssuer    [1] GeneralNames            OPTIONAL,
    authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL }
( WITH COMPONENTS { ..., authorityCertIssuer PRESENT,
    authorityCertSerialNumber PRESENT } |
    WITH COMPONENTS { ..., authorityCertIssuer ABSENT,
    authorityCertSerialNumber ABSENT } )

KeyIdentifier ::= OCTET STRING

subjectKeyIdentifier EXTENSION ::= {
    SYNTAX      SubjectKeyIdentifier
    IDENTIFIED BY id-ce-subjectKeyIdentifier }

SubjectKeyIdentifier ::= KeyIdentifier

keyUsage EXTENSION ::= {
    SYNTAX      KeyUsage
    IDENTIFIED BY id-ce-keyUsage }

KeyUsage ::= BIT STRING {
    digitalSignature      (0),
    nonRepudiation        (1),
    keyEncipherment       (2),
    dataEncipherment      (3),
    keyAgreement          (4),
    keyCertSign           (5),
    cRLSign               (6),
    encipherOnly          (7),
    decipherOnly          (8) }

extKeyUsage EXTENSION ::= {
    SYNTAX      SEQUENCE SIZE (1..MAX) OF KeyPurposeId
    IDENTIFIED BY id-ce-extKeyUsage }

KeyPurposeId ::= OBJECT IDENTIFIER

privateKeyUsagePeriod EXTENSION ::= {
    SYNTAX      PrivateKeyUsagePeriod
    IDENTIFIED BY id-ce-privateKeyUsagePeriod }

PrivateKeyUsagePeriod ::= SEQUENCE {
    notBefore      [0] GeneralizedTime OPTIONAL,
    notAfter       [1] GeneralizedTime OPTIONAL }
( WITH COMPONENTS { ..., notBefore PRESENT } |
    WITH COMPONENTS { ..., notAfter PRESENT } )

certificatePolicies EXTENSION ::= {
    SYNTAX      CertificatePoliciesSyntax
    IDENTIFIED BY id-ce-certificatePolicies }

CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
    policyIdentifier  CertPolicyId,
    policyQualifiers SEQUENCE SIZE (1..MAX) OF
        PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
    policyQualifierId CERT-POLICY-QUALIFIER.&id
        ({SupportedPolicyQualifiers}),
    qualifier         CERT-POLICY-QUALIFIER.&Qualifier
        ({SupportedPolicyQualifiers}{@policyQualifierId})
        OPTIONAL }

SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { ... }

```

```

CERT-POLICY-QUALIFIER ::= CLASS {
    &id      OBJECT IDENTIFIER UNIQUE,
    &Qualifier OPTIONAL }
WITH SYNTAX {
    POLICY-QUALIFIER-ID    &id
    [QUALIFIER-TYPE &Qualifier] }

policyMappings EXTENSION ::= {
    SYNTAX      PolicyMappingsSyntax
    IDENTIFIED BY id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
    issuerDomainPolicy      CertPolicyId,
    subjectDomainPolicy     CertPolicyId }

supportedAlgorithms ATTRIBUTE ::= {
    WITH SYNTAX SupportedAlgorithm
    EQUALITY MATCHING RULE algorithmIdentifierMatch
    ID id-at-supportedAlgorithms }

SupportedAlgorithm ::= SEQUENCE {
    algorithmIdentifier      AlgorithmIdentifier,
    intendedUsage            [0] KeyUsage OPTIONAL,
    intendedCertificatePolicies [1] CertificatePoliciesSyntax OPTIONAL }
-- Certificate subject and certificate issuer attributes extensions --

subjectAltName EXTENSION ::= {
    SYNTAX      GeneralNames
    IDENTIFIED BY id-ce-subjectAltName }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName          [0] INSTANCE OF OTHER-NAME,
    rfc822Name         [1] IA5String,
    dNSName            [2] IA5String,
    x400Address        [3] ORAddress,
    directoryName      [4] Name,
    ediPartyName       [5] EDIPartyName,
    uniformResourceIdentifier [6] IA5String,
    iPAddress          [7] OCTET STRING,
    registeredID       [8] OBJECT IDENTIFIER }

OTHER-NAME ::= TYPE-IDENTIFIER

EDIPartyName ::= SEQUENCE {
    nameAssigner [0] DirectoryString {ub-name} OPTIONAL,
    partyName    [1] DirectoryString {ub-name} }

issuerAltName EXTENSION ::= {
    SYNTAX      GeneralNames
    IDENTIFIED BY id-ce-issuerAltName }

subjectDirectoryAttributes EXTENSION ::= {
    SYNTAX      AttributesSyntax
    IDENTIFIED BY id-ce-subjectDirectoryAttributes }

AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute
-- Certification path constraints extensions --

basicConstraints EXTENSION ::= {
    SYNTAX      BasicConstraintsSyntax
    IDENTIFIED BY id-ce-basicConstraints }

BasicConstraintsSyntax ::= SEQUENCE {
    cA      BOOLEAN DEFAULT FALSE,
    pathLenConstraint INTEGER (0..MAX) OPTIONAL }

nameConstraints EXTENSION ::= {
    SYNTAX      NameConstraintsSyntax
    IDENTIFIED BY id-ce-nameConstraints }

```

NameConstraintsSyntax ::= SEQUENCE {
 permittedSubtrees [0] GeneralSubtrees OPTIONAL,
 excludedSubtrees [1] GeneralSubtrees OPTIONAL }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
 base GeneralName,
 minimum [0] BaseDistance DEFAULT 0,
 maximum [1] BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)

policyConstraints EXTENSION ::= {
 SYNTAX PolicyConstraintsSyntax
 IDENTIFIED BY id-ce-policyConstraints }

PolicyConstraintsSyntax ::= SEQUENCE {
 requireExplicitPolicy [0] SkipCerts OPTIONAL,
 inhibitPolicyMapping [1] SkipCerts OPTIONAL }

SkipCerts ::= INTEGER (0..MAX)

CertPolicySet ::= SEQUENCE (1..MAX) OF CertPolicyId

-- Basic CRL extensions --

cRLNumber EXTENSION ::= {
 SYNTAX CRLNumber
 IDENTIFIED BY id-ce-cRLNumber }

CRLNumber ::= INTEGER (0..MAX)

reasonCode EXTENSION ::= {
 SYNTAX CRLReason
 IDENTIFIED BY id-ce-reasonCode }

CRLReason ::= ENUMERATED {
 unspecified (0),
 keyCompromise (1),
 cACompromise (2),
 affiliationChanged (3),
 superseded (4),
 cessationOfOperation (5),
 certificateHold (6),
 removeFromCRL (8) }

instructionCode EXTENSION ::= {
 SYNTAX HoldInstruction
 IDENTIFIED BY id-ce-instructionCode }

HoldInstruction ::= OBJECT IDENTIFIER

invalidityDate EXTENSION ::= {
 SYNTAX GeneralizedTime
 IDENTIFIED BY id-ce-invalidityDate }

-- CRL distribution points and delta-CRL extensions --

cRLDistributionPoints EXTENSION ::= {
 SYNTAX CRLDistPointsSyntax
 IDENTIFIED BY id-ce-cRLDistributionPoints }

CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint

DistributionPoint ::= SEQUENCE {
 distributionPoint [0] DistributionPointName OPTIONAL,
 reasons [1] ReasonFlags OPTIONAL,
 cRLIssuer [2] GeneralNames OPTIONAL }

DistributionPointName ::= CHOICE {
 fullName [0] GeneralNames,
 nameRelativeToCRLIssuer [1] RelativeDistinguishedName }

```

ReasonFlags ::= BIT STRING {
    unused                (0),
    keyCompromise         (1),
    caCompromise          (2),
    affiliationChanged     (3),
    superseded             (4),
    cessationOfOperation  (5),
    certificateHold        (6) }

issuingDistributionPoint EXTENSION ::= {
    SYNTAX      IssuingDistPointSyntax
    IDENTIFIED BY id-ce-issuingDistributionPoint }

IssuingDistPointSyntax ::= SEQUENCE {
    distributionPoint      [0] DistributionPointName OPTIONAL,
    onlyContainsUserCerts [1] BOOLEAN DEFAULT FALSE,
    onlyContainsCACerts   [2] BOOLEAN DEFAULT FALSE,
    onlySomeReasons       [3] ReasonFlags OPTIONAL,
    indirectCRL           [4] BOOLEAN DEFAULT FALSE }

certificateIssuer EXTENSION ::= {
    SYNTAX      GeneralNames
    IDENTIFIED BY id-ce-certificateIssuer }

deltaCRLIndicator EXTENSION ::= {
    SYNTAX      BaseCRLNumber
    IDENTIFIED BY id-ce-deltaCRLIndicator }

BaseCRLNumber ::= CRLNumber

deltaRevocationList ATTRIBUTE ::= {
    WITH SYNTAX      CertificateList
    EQUALITY MATCHING RULE certificateListExactMatch
    ID id-at-deltaRevocationList }

-- Matching rules --

certificateExactMatch MATCHING-RULE ::= {
    SYNTAX      CertificateExactAssertion
    ID          id-mr-certificateExactMatch }

CertificateExactAssertion ::= SEQUENCE {
    serialNumber      CertificateSerialNumber,
    issuer            Name }

certificateMatch MATCHING-RULE ::= {
    SYNTAX      CertificateAssertion
    ID          id-mr-certificateMatch }

CertificateAssertion ::= SEQUENCE {
    serialNumber      [0] CertificateSerialNumber OPTIONAL,
    issuer            [1] Name OPTIONAL,
    subjectKeyIdentifier [2] SubjectKeyIdentifier OPTIONAL,
    authorityKeyIdentifier [3] AuthorityKeyIdentifier OPTIONAL,
    certificateValid     [4] Time OPTIONAL,
    privateKeyValid      [5] GeneralizedTime OPTIONAL,
    subjectPublicKeyAlgID [6] OBJECT IDENTIFIER OPTIONAL,
    keyUsage             [7] KeyUsage OPTIONAL,
    subjectAltName        [8] AltNameType OPTIONAL,
    policy               [9] CertPolicySet OPTIONAL,
    pathToName           [10] Name OPTIONAL }

AltNameType ::= CHOICE {
    builtinNameForm  ENUMERATED {
        rfc822Name      (1),
        dNSName         (2),
        x400Address     (3),
        directoryName   (4),
        ediPartyName    (5),
        uniformResourceIdentifier (6),
        iPAddress        (7),
        registeredId    (8) },
    otherNameForm    OBJECT IDENTIFIER }

```

```
certificatePairExactMatch MATCHING-RULE ::= {
    SYNTAX      CertificatePairExactAssertion
    ID          id-mr-certificatePairExactMatch }
```

```
CertificatePairExactAssertion ::= SEQUENCE {
    forwardAssertion  [0] CertificateExactAssertion OPTIONAL,
    reverseAssertion  [1] CertificateExactAssertion OPTIONAL }
( WITH COMPONENTS    {..., forwardAssertion PRESENT} |
  WITH COMPONENTS    {..., reverseAssertion PRESENT} )
```

```
certificatePairMatch MATCHING-RULE ::= {
    SYNTAX      CertificatePairAssertion
    ID          id-mr-certificatePairMatch }
```

```
CertificatePairAssertion ::= SEQUENCE {
    forwardAssertion  [0] CertificateAssertion OPTIONAL,
    reverseAssertion  [1] CertificateAssertion OPTIONAL }
( WITH COMPONENTS    {..., forwardAssertion PRESENT} |
  WITH COMPONENTS    {..., reverseAssertion PRESENT} )
```

```
certificateListExactMatch MATCHING-RULE ::= {
    SYNTAX      CertificateListExactAssertion
    ID          id-mr-certificateListExactMatch }
```

```
CertificateListExactAssertion ::= SEQUENCE {
    issuer           Name,
    thisUpdate       Time,
    distributionPoint DistributionPointName OPTIONAL }
```

```
certificateListMatch MATCHING-RULE ::= {
    SYNTAX      CertificateListAssertion
    ID          id-mr-certificateListMatch }
```

```
CertificateListAssertion ::= SEQUENCE {
    issuer           Name      OPTIONAL,
    minCRLNumber     [0]      CRLNumber  OPTIONAL,
    maxCRLNumber     [1]      CRLNumber  OPTIONAL,
    reasonFlags      ReasonFlags  OPTIONAL,
    dateAndTime       Time      OPTIONAL,
    distributionPoint [2]      DistributionPointName OPTIONAL }
```

```
algorithmIdentifierMatch MATCHING-RULE ::= {
    SYNTAX      AlgorithmIdentifier
    ID          id-mr-algorithmIdentifierMatch }
```

-- Object identifier assignments --

id-at-supportedAlgorithms	OBJECT IDENTIFIER ::= {id-at 52}
id-at-deltaRevocationList	OBJECT IDENTIFIER ::= {id-at 53}
id-ce-subjectDirectoryAttributes	OBJECT IDENTIFIER ::= {id-ce 9}
id-ce-subjectKeyIdentifier	OBJECT IDENTIFIER ::= {id-ce 14}
id-ce-keyUsage	OBJECT IDENTIFIER ::= {id-ce 15}
id-ce-privateKeyUsagePeriod	OBJECT IDENTIFIER ::= {id-ce 16}
id-ce-subjectAltName	OBJECT IDENTIFIER ::= {id-ce 17}
id-ce-issuerAltName	OBJECT IDENTIFIER ::= {id-ce 18}
id-ce-basicConstraints	OBJECT IDENTIFIER ::= {id-ce 19}
id-ce-cRLNumber	OBJECT IDENTIFIER ::= {id-ce 20}
id-ce-reasonCode	OBJECT IDENTIFIER ::= {id-ce 21}
id-ce-instructionCode	OBJECT IDENTIFIER ::= {id-ce 23}
id-ce-invalidityDate	OBJECT IDENTIFIER ::= {id-ce 24}
id-ce-deltaCRLIndicator	OBJECT IDENTIFIER ::= {id-ce 27}
id-ce-issuingDistributionPoint	OBJECT IDENTIFIER ::= {id-ce 28}
id-ce-certificateIssuer	OBJECT IDENTIFIER ::= {id-ce 29}
id-ce-nameConstraints	OBJECT IDENTIFIER ::= {id-ce 30}
id-ce-cRLDistributionPoints	OBJECT IDENTIFIER ::= {id-ce 31}
id-ce-certificatePolicies	OBJECT IDENTIFIER ::= {id-ce 32}
id-ce-policyMappings	OBJECT IDENTIFIER ::= {id-ce 33}
-- deprecated	OBJECT IDENTIFIER ::= {id-ce 34}
id-ce-authorityKeyIdentifier	OBJECT IDENTIFIER ::= {id-ce 35}
id-ce-policyConstraints	OBJECT IDENTIFIER ::= {id-ce 36}

id-ce-extKeyUsage	OBJECT IDENTIFIER ::= {id-ce 37}
id-mr-certificateExactMatch	OBJECT IDENTIFIER ::= {id-mr 34}
id-mr-certificateMatch	OBJECT IDENTIFIER ::= {id-mr 35}
id-mr-certificatePairExactMatch	OBJECT IDENTIFIER ::= {id-mr 36}
id-mr-certificatePairMatch	OBJECT IDENTIFIER ::= {id-mr 37}
id-mr-certificateListExactMatch	OBJECT IDENTIFIER ::= {id-mr 38}
id-mr-certificateListMatch	OBJECT IDENTIFIER ::= {id-mr 39}
id-mr-algorithmIdentifierMatch	OBJECT IDENTIFIER ::= {id-mr 40}

-- *The following OBJECT IDENTIFIERS are not used by this Specification:*

-- {id-ce 2}, {id-ce 3}, {id-ce 4}, {id-ce 5}, {id-ce 6}, {id-ce 7},
 -- {id-ce 8}, {id-ce 10}, {id-ce 11}, {id-ce 12}, {id-ce 13},
 -- {id-ce 22}, {id-ce 25}, {id-ce 26}

END

Anexo B

Requisitos de seguridad²⁾

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

Muchas aplicaciones OSI, servicios definidos por el UIT-T y servicios no definidos por el UIT-T tendrán exigencias de seguridad. Tales exigencias se explican por la necesidad de proteger la transferencia de información contra una serie de peligros potenciales.

B.1 Peligros

Algunos peligros comúnmente conocidos son:

- a) *Intercepción de identidad* – La identidad de uno o más de los usuarios que participan en una comunicación se observa con el fin de detectar todo uso incorrecto.
- b) *Usurpación de identidad, suplantación, impostura o mascarada (masquerade)* – Pretensión de un usuario de ser otro diferente para ganar acceso a información u obtener privilegios adicionales.
- c) *Reactuación o reproducción (Replay)* – Registración y posterior reactuación de una comunicación en alguna fecha posterior.
- d) *Intercepción de datos* – Observación de datos de un usuario durante una comunicación, por un usuario no autorizado.
- e) *Manipulación* – Reemplazo, inserción, supresión u ordenación incorrecta de datos de usuario durante una comunicación, por un usuario no autorizado.
- f) *Repudio* – Negación por un usuario de haber participado en una comunicación, o parte de ella.
- g) *Denegación de servicio* – Prevención o interrupción de una comunicación, o demora de operaciones críticas en el tiempo.

NOTA 1 – Este peligro para la seguridad es más general y depende de la aplicación individual o de la intención de la perturbación no autorizada y, por consiguiente, no está explícitamente dentro del ámbito del marco de autenticación.

- h) *Encaminamiento incorrecto* – Encaminamiento incorrecto de un trayecto de comunicación previsto de un usuario a otro.

NOTA 2 – El encaminamiento incorrecto ocurrirá naturalmente en las capas 1 a 3 de OSI, por lo que está fuera del ámbito del marco de autenticación. Sin embargo, puede ser posible evitar las consecuencias del encaminamiento incorrecto utilizando servicios apropiados de seguridad como los suministrados dentro del marco de autenticación.

- i) *Análisis de tráfico* – Observación de información sobre una comunicación entre usuarios (por ejemplo, ausencia/presencia, frecuencia, sentido de transmisión, secuencia, tipo, cantidad, etc.).

NOTA 3 – Los peligros de análisis de tráfico no están naturalmente limitados a una capa OSI determinada. Por consiguiente el análisis de tráfico está generalmente fuera del ámbito del marco de autenticación. Sin embargo, el análisis de tráfico puede ser protegido parcialmente generando tráfico adicional ininteligible (tráfico de relleno), usando datos cifrados o aleatorios.

B.2 Servicios de seguridad

Para la protección contra los peligros conocidos deben prestarse diversos servicios de seguridad. Los servicios de seguridad proporcionados por el marco de autenticación se efectúan por medio de los mecanismos de seguridad descritos en B.3.

- a) *Autenticación de entidad par* – Este servicio proporciona una corroboración de que un usuario en una determinada instancia de comunicación es el que se anuncia como tal. Pueden solicitarse dos servicios diferentes de autenticación de identidad par:
 - *autenticación de entidad simple* (ya sea autenticación de entidad de *origen de datos* o autenticación de entidad de *receptor de datos*);
 - *autenticación mutua*, donde ambos usuarios comunicantes se autentican el uno al otro.

²⁾ Para mayor información, véase ISO 7498-2.

Cuando se solicita un servicio de autenticación de entidad par, los dos usuarios acuerdan si sus identidades serán protegidas o no.

El servicio de autenticación de entidad par es soportado por el marco de autenticación. Puede ser usado para proteger contra la usurpación de identidad y la reactuación, concernientes a las identidades de los usuarios.

- b) *Control de acceso* – Este servicio puede usarse para proteger contra el uso no autorizado de recursos. El servicio de control de acceso es proporcionado por el directorio u otra aplicación y no es por consiguiente un asunto del marco de autenticación.
- c) *Confidencialidad de datos* – Este servicio puede usarse para suministrar protección de los datos contra una revelación no autorizada. El servicio de confidencialidad de datos está soportado por el marco de autenticación. El mismo puede usarse para proteger contra intercepción de datos.
- d) *Integridad de datos* – Este servicio suministra prueba de la integridad de los datos en una comunicación. El servicio de integridad de datos está soportado por el marco de autenticación. Puede usarse para detectar y proteger contra la manipulación.
- e) *No repudio* – Este servicio suministra la prueba de la integridad y del origen de los datos – ambos en una relación infalsificable – que pueden ser verificados por un tercero en cualquier momento.

B.3 Mecanismos de seguridad

Los mecanismos de seguridad que se describen aquí permiten efectuar los servicios de seguridad descritos en B.2.

- a) *Intercambio de autenticación* – Hay dos grados de mecanismos de autenticación suministrados por el marco de autenticación:
 - *Autenticación simple* – Se basa en que el originador suministre su nombre y contraseña, los cuales son comprobados por el receptor.
 - *Autenticación fuerte* – Se basa en el uso de técnicas criptográficas para proteger el intercambio de información de validación. En el marco de autenticación, la autenticación fuerte se basa en un esquema asimétrico.

El mecanismo de intercambio de autenticación se usa para soportar el servicio de autenticación de entidad par.

- b) *Cifrado* – El marco de autenticación contempla el cifrado de datos durante la transferencia. Pueden usarse esquemas simétricos o asimétricos. El intercambio necesario de claves se realiza o bien dentro de un intercambio de autenticación precedente o fuera de línea en cualquier momento antes de la comunicación que se va a hacer. Este último caso está fuera del ámbito del marco de autenticación. El mecanismo de cifrado soporta el servicio de confidencialidad de datos.
- c) *Integridad de los datos* – Este mecanismo implica el cifrado de una cadena comprimida de los datos pertinentes a transmitir. Junto con los datos ordinarios, este mensaje se le envía al receptor. El receptor repite la compresión y el cifrado ulterior de los datos ordinarios y compara el resultado con el creado por el originador para probar la integridad.

El mecanismo de integridad de datos puede ser suministrado por cifrado de los datos ordinarios comprimidos ya sea por un esquema asimétrico o por un esquema simétrico. (Con el esquema simétrico, la compresión y el cifrado de los datos pudieran ser procesados simultáneamente.) El mecanismo no es suministrado explícitamente por el marco de autenticación. Sin embargo, se suministra totalmente como una parte del mecanismo de firma digital (véase más adelante) usando un esquema asimétrico.

El mecanismo de integridad de datos soporta el servicio de integridad de datos. También soporta parcialmente el servicio de no-repudio (ese servicio también necesita el mecanismo de firma digital para que sus requisitos se cumplan plenamente).

- d) *Firma digital* – Este mecanismo implica el cifrado, por medio de la clave privada del originador, de una cadena comprimida de los datos pertinentes que se van a transferir. La firma digital, junto con los datos ordinarios se envía al receptor. Similarmente al caso del mecanismo de integridad de datos, este mensaje es procesado por el receptor para probar la integridad. El mecanismo de firma digital también prueba la autenticidad del originador y la relación inequívoca entre el originador y los datos que se transfirieron.

El marco de autenticación soporta el mecanismo de firma digital usando un esquema asimétrico.

El mecanismo de signatura digital soporta el servicio de integridad de datos y también el servicio de no-repudio.

B.4 Peligros contra los que protegen los servicios de seguridad

El cuadro B.1 indica los peligros de seguridad contra los que cada servicio de seguridad puede proteger. La presencia de un punto grueso ("•") indica que un cierto servicio de seguridad ofrece protección contra cierto peligro.

Cuadro B.1 – Peligros y protección

Peligros	Servicios			
	Autenticación de entidad	Confidencialidad de datos	Integridad de datos	No repudio
Intercepción de identidad	• (si se requiere)			
Intercepción de datos		•		
Usurpación	•			
Reactuación	• (identidad)		• (datos)	•
Manipulación			•	
Repudio				•

B.5 Negociación de servicios y mecanismos de seguridad

La provisión de prestaciones de seguridad durante una instancia de comunicación requiere la negociación del contexto en el cual se requieren los servicios de seguridad. Esto implica el acuerdo en el tipo de mecanismos de seguridad y de parámetros que son necesarios para suministrar tales servicios de seguridad. Los procedimientos que se requieren para negociar los mecanismos y parámetros pueden o bien ser llevados a cabo como una parte integrante del procedimiento normal de establecimiento de conexión, o como un proceso separado. Los detalles precisos de estos procedimientos para la negociación no se especifican en este anexo.

Anexo C

Introducción a la criptografía de claves públicas³⁾

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

En los sistemas criptográficos convencionales, la clave usada para cifrar la información por el originador de un mensaje secreto es la misma usada por el receptor legítimo para descifrar el mensaje.

En los criptosistemas de claves públicas (PKCS), sin embargo, las claves vienen en pares; una de las cuales se usa para el cifrado y la otra para el descifrado. Cada par de claves se asocia con un usuario particular X. Una de las claves, conocida como la clave pública (X_p) se conoce públicamente, y puede ser usada por cualquier usuario para cifrar datos. Solamente X, quien posee la clave privada complementaria (X_s), puede descifrar los datos. (Esto se representa por la notación $D = X_s[X_p[D]]$.) Es computacionalmente irrealizable derivar la clave privada a partir del conocimiento de la clave pública. Cualquier usuario puede entonces comunicar una información la cual solamente X puede hallar, cifrándola bajo X_p . Por extensión, dos usuarios pueden comunicar en secreto, usando cada uno la clave pública del otro para cifrar los datos, como se muestra en la figura C.1.

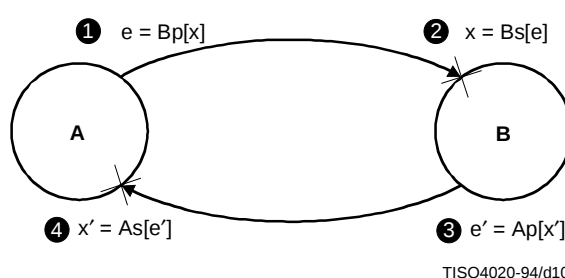


Figura C.1 – Uso de un PKCS para intercambiar información secreta

El usuario A tiene la clave pública A_p y la clave privada A_s , y el usuario B tiene otro conjunto de claves, B_p y B_s . A y B conocen cada uno la clave pública, pero no la clave privada del otro. A y B pueden por consiguiente intercambiar información secreta entre ellos siguiendo los pasos siguientes (ilustrados en la figura C.1).

- 1) A desea enviar alguna información secreta x a B. A por consiguiente cifra x bajo la clave de cifrado de B y envía la información cifrada e a B. Esto se representa por:

$$e = B_p[x]$$

- 2) B puede ahora descifrar este cifrado e para obtener la información x usando la clave secreta de descifrado B_s . Obsérvese que B es el único poseedor de B_s , y debido a que esta clave puede que nunca sea revelada o enviada, es imposible para cualquier otra parte obtener la información x . La posesión de B_s determina la identidad de B. La operación de descifrado se representa por:

$$x = B_s[e], \text{ o } x = B_s[B_p[x]]$$

- 3) B puede ahora, análogamente, enviar alguna información secreta, x' , a A, bajo la clave de cifrado de A, A_p :

$$e' = A_p[x']$$

- 4) A obtiene x' descifrando e' :

$$x' = A_s[e'], \text{ o } x' = A_s[A_p[x']]$$

Por este medio, A y B han intercambiado la información secreta x y x' . Esta información no puede ser obtenida por ningún otro que A y B, siempre que sus claves privadas no sean reveladas.

³⁾ Para más información, véase:

DIFFIE (W.) y HELLMAN (M. E.): New Directions in Cryptography, *IEEE Transactions on Information Theory*, IT-22, N.º 6 (Noviembre 1976).

Un intercambio tal puede servir para verificar sus identidades, así como para transferir la información secreta entre las partes. Específicamente, A y B se identifican por su posesión de las claves secretas de descifrado, A_s y B_s respectivamente. A puede determinar si B está en posesión de la clave secreta de descifrado, B_s , haciendo retornar parte de su información x en el mensaje x' de B. Esto le indica a A que la comunicación está teniendo lugar con el propietario de B_s . B puede, de manera similar, probar la identidad de A.

Es una propiedad de algunos PKCS que los pasos de descifrado y cifrado puedan invertirse, como en $D = X_p[X_s[D]]$. Esto permite que una información que pudiera haber sido originada solamente por X sea legible por cualquier usuario (que esté en posesión de X_p). Esto puede usarse por consiguiente al certificar la fuente de información, y es la base para las firmas digitales. Solamente los PKCS que tienen esta propiedad (permeabilidad) son apropiados para uso en este marco de autenticación. En el anexo D se describe uno de estos algoritmos.

Anexo D

El criptosistema⁴⁾ de claves públicas RSA⁵⁾

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

D.1 Alcance y campo de aplicación

Está fuera del alcance de este anexo discutir el algoritmo RSA en su totalidad. Sin embargo, se da una breve descripción sobre el método, el cual se basa en el uso de exponenciación modular.

D.2 Definiciones

Los siguientes términos se definen en este campo.

D.2.1 clave pública: El par de parámetros formado por el exponente público y el módulo aritmético.

NOTA – El elemento de datos ASN.1 **subjectPublicKey**, definido como **BIT STRING** (véase el anexo A) debe interpretarse en el caso de los sistemas RSA como si fuese del tipo:

SEQUENCE {INTEGER, INTEGER}

donde el primer entero es el módulo aritmético y el segundo es el exponente público. La secuencia se representa por medio de las reglas de codificación básica ASN.1.

D.2.2 clave privada: El par de parámetros formado por el exponente secreto y el módulo aritmético.

D.3 Símbolos y abreviaturas

A los efectos de este anexo se utilizan los siguientes símbolos y abreviaturas:

X, Y	Bloques de datos que son aritméticamente menores que el módulo
n	Módulo aritmético
e	Exponente público
d	Exponente secreto
p, q	Números primos cuyo producto forma el módulo aritmético (n) NOTA – Se prefiere utilizar dos números primos; sin embargo, no se excluye el uso de un módulo con tres o más factores primos.
lcm	Mínimo común múltiplo
mod n	Módulo aritmético n

⁴⁾ El criptosistema especificado en este anexo fue creado por Rivest-Shamir-Adleman, y se conoce generalmente por algoritmo *RSA*.

⁵⁾ Para más información, véase:

Aspectos generales:

RIVEST (R.L.) SHAMIR (A.) y ADLEMAN (L.): – A Method for Obtaining Digital Signatures and Public-key Cryptosystems, *Communications of the ACM*, 21, 2, 120-126, Febrero 1978.

Referencia de generación de claves:

GORDON (J.): Strong RSA Keys, *Electronics Letters*, 20, 5, 514-516.

Referencia de descifrado:

QUISQUATER (J.J.) y COUVREUR (C.): Fast Decipherment Algorithm for RSA Public-key Cryptosystems, *Electronics Letters*, 18, 21, 905-907, 14 octubre 1982.

D.4 Descripción

Este algoritmo asimétrico usa la función de potenciación para la transformación de bloques de datos de la siguiente forma:

- $Y = X^e \bmod n$ con $0 \leq X < n$;
- $X = Y^d \bmod n$ con $0 \leq Y < n$,

que puede ser satisfecha, por ejemplo, por:

- $ed \bmod \text{lcm}(p-1, q-1) = 1$; o
- $ed \bmod (p-1)(q-1) = 1$.

Para efectuar este proceso, un bloque de datos debe interpretarse como un entero. Esto se obtiene considerando que el bloque completo de datos es una secuencia ordenada de bits (por ejemplo, de longitud λ) donde el primer bit es el bit de orden más elevado del primer octeto del bloque de datos. El entero se forma entonces como la suma de los bits después de darle un peso de $2^{\lambda-1}$ al primer bit, y dividiendo el peso por 2 para cada bit ulterior (el último bit tiene un peso de 1).

La longitud del bloque de datos debe ser el mayor número de octetos que contienen menos bits que el módulo. Los bloques incompletos deben ser rellenados de cualquier manera deseada. Se puede añadir cualquier número de bloques de relleno adicionales.

D.5 Requisitos de seguridad

D.5.1 Longitudes de las claves

Se reconoce que la longitud aceptable de la clave es probable que cambie con el tiempo, en función del costo y la disponibilidad del hardware, el tiempo necesario, los avances en las técnicas y el nivel de seguridad requerido. Se recomienda adoptar inicialmente para la longitud de n un valor de 512 bits, pero este aspecto *queda en estudio*.

D.5.2 Generación de claves

La seguridad de RSA se basa en la dificultad de factorizar n . Hay muchos algoritmos para realizar esta operación, y para obstaculizar el uso de cualquier técnica actualmente conocida, los valores p y q tienen que ser escogidos cuidadosamente, de acuerdo a las reglas siguientes (por ejemplo, véase la nota de pie de página 5, "Referencia de generación de claves"):

- a) deben ser escogidos al azar;
- b) deben ser grandes;
- c) deben ser números primos;
- d) $|p-q|$ debe ser grande;
- e) $(p+1)$ tendrá un factor primo grande;
- f) $(q+1)$ tendrá un factor primo grande;
- g) $(p-1)$ tendrá un factor primo grande, por ejemplo, r ;
- h) $(q-1)$ tendrá un factor primo grande, por ejemplo, s ;
- i) $(r-1)$ tendrá un factor primo grande;
- j) $(s-1)$ tendrá un factor primo grande.

Después de generar las claves pública y privada " X_p " y " X_s ", como se define en las cláusulas 3 y 4, constituidas por d , e y n , los valores p y q junto con todos los otros datos producidos tales como el producto $(p-1)(q-1)$ y los factores primos grandes deben ser preferiblemente destruidos. Sin embargo, el mantener p y q localmente puede mejorar el rendimiento en la descifrado por un factor de dos a cuatro. La decisión de mantener p y q se considera un asunto local (véase la nota de pie de página 5, "Referencia de descifrado").

Se tiene que asegurar que $e > \log_2(n)$. Si no se hace esto, la simple operación de tomar la e -ésima raíz de un bloque de texto cifrado revelará el texto en lenguaje ordinario.

D.6 Exponente público

El exponente público (e) podría ser común al entorno total, para minimizar la longitud de esa parte de la clave pública que, en efecto, tiene que ser distribuida, para reducir la capacidad de transmisión requerida y la complejidad de la transformación (véase la nota 1).

El exponente e debe ser suficientemente grande, pero hasta un punto tal que la exponenciación pueda ser realizada eficientemente con respecto al tiempo de procesamiento y a la capacidad de almacenamiento. Por consiguiente se recomienda que el exponente e sea el Número Fermat F_4 (véase la nota 2).

$$F_4 = 2^{2^4} + 1$$

= 65537 decimal, y

= 1 0000 0000 0000 0001 binario

NOTA 1 – Aunque el módulo n y el exponente e son públicos, el módulo no debe ser la parte que es común a un grupo de usuarios. El conocimiento del módulo " n ", del exponente público " e " y del exponente secreto " d " es suficiente para determinar la factorización de " n ". Por tanto, si el módulo fuera común, todo el mundo podría deducir sus factores, y todo el mundo podría averiguar el exponente secreto de todos los demás.

NOTA 2 – El exponente fijo tiene que ser grande y primo pero también tiene que permitir un procesamiento eficiente. El número Fermat F_4 cumple estos requisitos, por ejemplo, la autenticación necesita solamente 17 multiplicaciones y es en promedio 30 veces más rápida que el descifrado.

D.7 Conformidad

Aunque este anexo especifica un algoritmo para las funciones pública y secreta, no define el método para efectuar los cálculos; por consiguiente, pueden existir distintos productos que cumplan con este anexo y sean mutuamente compatibles.

Anexo E

Funciones de troceo (Hash)

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

Se desaconseja la función de troceo cuadrado-mod (square-mod hash) descrita en este anexo en la primera edición de la presente Especificación de directorio.

E.1 Requisitos de las funciones de troceo

Para utilizar una función de troceo como una función unidireccional segura es condición indispensable que no sea posible obtener fácilmente el mismo resultado de troceo a partir de diferentes combinaciones del mensaje de entrada.

Una función de troceo fuerte cumplirá los siguientes requisitos:

- a) La función de troceo será unidireccional, es decir, dado un resultado de troceo posible cualquiera, no se podrá construir computacionalmente un mensaje de entrada que al ser sometido a una función de troceo, dé este resultado.
- b) La función de troceo tiene que estar libre de colisiones, es decir, no se podrá construir computacionalmente dos mensajes de entrada distintos que al ser sometidos a una función de troceo den el mismo resultado.

Anexo F

Peligros contra los que ofrece protección el método de autenticación fuerte

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

El método de autenticación fuerte que se describe en esta Especificación de directorio ofrece protección contra los peligros como se describe en el anexo B para la autenticación fuerte.

Además, hay una gama de peligros potenciales que son específicos del propio método de autenticación fuerte. Estos peligros son:

- *Comprometer la clave secreta del usuario* – Uno de los principios básicos de autenticación fuerte es que la clave secreta del usuario permanezca segura. Un número de métodos prácticos están disponibles para que el usuario mantenga su clave privada en una forma que ofrezca la seguridad adecuada. Las consecuencias de esta situación se limitan a un trastorno de la comunicación en que interviene ese usuario.
- *Comprometer la clave privada de la CA* – El hecho de que la clave privada de una CA permanezca segura es también un principio básico de la autenticación fuerte. La seguridad física y los métodos "necesidad de conocer" se aplican. Las consecuencias de esta situación se limitan a un trastorno de la comunicación en que interviene cualquier usuario certificado por esa CA.
- *Inducir a error a la CA para que cree un certificado no válido* – El hecho de que las CA funcionen fuera de línea da cierta protección. Recae sobre la CA el trabajo de comprobar que las credenciales fuertes contempladas son válidas, antes de crear un certificado. Las consecuencias de esta situación se limitan a un trastorno de la comunicación en que interviene el usuario para el cual se creó el certificado, y cualquiera afectado por el certificado no válido.
- *Colusión entre una CA deshonesto y un usuario* – Un ataque de este tipo hará fracasar este método. Esto podría constituir una traición a la confianza depositada en la CA. Las consecuencias de una CA deshonesto se limitan a un trastorno de la comunicación en que interviene cualquier usuario certificado por esa CA.
- *Falsificación de un certificado* – El método de autenticación fuerte protege contra la falsificación de un certificado consiguiendo que lo firme la CA. El método depende del mantenimiento del secreto de la clave privada de la CA.
- *Falsificación de un token* – El método de autenticación fuerte protege contra la falsificación de un token haciendo que lo firme el emisor. El método depende del mantenimiento del secreto de la clave privada del emisor.
- *Reactuación de un token* – Los métodos de autenticación unidireccionales y bidireccionales protegen contra la reactuación de un token por medio de la inclusión de una indicación de tiempo en el token. El método tridireccional lo hace por medio de la comprobación de los números aleatorios.
- *Ataque al sistema criptográfico* – Los adelantos conseguidos en la teoría de los números, basados en las nuevas técnicas computacionales se reflejan en una mayor probabilidad de eficaces criptoanálisis de los sistemas; de ahí que sea razonable pensar en claves de mayor longitud.

Anexo G

Confidencialidad de los datos

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

G.1 Introducción

El proceso de confidencialidad de los datos puede iniciarse después de que las claves necesarias para el cifrado hayan sido intercambiadas. Esto pudiera efectuarse por un intercambio previo de autenticación tal como se describe en la cláusula 9 o por algún otro proceso de intercambio de claves; esto último está fuera del alcance de esta Especificación de directorio.

La confidencialidad de los datos puede ofrecerse ya sea por la aplicación de un esquema de cifrado asimétrico o simétrico.

G.2 Confidencialidad de los datos por cifrado asimétrico

En este caso, la confidencialidad de los datos se obtiene cuando un originador cifra los datos que va a enviar usando la clave pública del receptor previsto: el receptor los descifrára usando su clave privada.

G.3 Confidencialidad de los datos por cifrado simétrico

En este caso la confidencialidad de los datos se logra mediante un algoritmo de cifrado simétrico. Su selección está fuera del ámbito del marco de autenticación.

Donde un intercambio de autenticación de acuerdo a la cláusula 9 se ha llevado a cabo por las dos partes interesadas, se puede derivar una clave para el uso de un algoritmo simétrico. La selección de claves privadas depende de la transformación que se utilice. Las partes tienen que estar seguras de que son claves fuertes. Esta Especificación de directorio no indica cómo se hace esta selección, aunque es evidente que esto debería ser acordado por las partes interesadas, o especificado en otras normas.

Anexo H

Definición de referencia de los identificadores de objeto para algoritmo

(Este anexo es parte integrante de esta Recomendación | Norma Internacional)

Este anexo define los identificadores de objeto asignados a los algoritmos de autenticación y criptación, en ausencia de un registro formal. Se tiene la intención de utilizar esos registros cuando estén disponibles. Las definiciones se presentan en forma del módulo ASN.1, "AlgorithmObjectIdentifiers".

```

AlgorithmObjectIdentifiers {joint-iso-ccitt ds(5) module(1) algorithmObjectIdentifiers(8) 3}
DEFINITIONS ::=
BEGIN

-- EXPORTS All --
-- The types and values defined in this module are exported for use in the other ASN.1 modules contained
-- within the Directory Specifications, and for the use of other applications which will use them to access
-- Directory services. Other applications may use them for their own purposes, but this will not constrain
-- extensions and modifications needed to maintain or improve the Directory service.

IMPORTS
    algorithm, authenticationFramework
    FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 3}
ALGORITHM
    FROM AuthenticationFramework authenticationFramework ;
-- categories of object identifier --

encryptionAlgorithm    OBJECT IDENTIFIER ::=      {algorithm 1}
hashAlgorithm          OBJECT IDENTIFIER ::=      {algorithm 2}
signatureAlgorithm     OBJECT IDENTIFIER ::=      {algorithm 3}
-- synonyms --

id-ea OBJECT IDENTIFIER ::=      encryptionAlgorithm
id-ha OBJECT IDENTIFIER ::=      hashAlgorithm
id-sa OBJECT IDENTIFIER ::=      signatureAlgorithm
-- algorithms --

rsa ALGORITHM          ::=      {
    KeySize
    IDENTIFIED BY id-ea-rsa }
KeySize                ::=      INTEGER
-- object identifier assignments --

id-ea-rsa              OBJECT IDENTIFIER ::=      {id-ea 1}
-- the following object identifier assignments reserve values assigned to deprecated functions

id-ha-sqMod-n          OBJECT IDENTIFIER ::=      {id-ha 1}
id-sa-sqMod-nWithRSA   OBJECT IDENTIFIER ::=      {id-sa 1}

END

```

Anexo I

Bibliografía

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

KENT (S.): Privacy Enhancement for Internet Electronic Mail, Part II: Certificate-Based Key Management, RFC 1422, *Internet Activities Board*, 1993.

Naciones Unidas, EDIFACT Security Implementation Guidelines, *Consejo Económico y Social* de las Naciones Unidas, TRADE/WP.4/R.1026, 7 de febrero de 1994. (Se incorporará en la ISO 9735 EDIFACT Syntax.)

Anexo J

Ejemplos de la utilización de constricciones del trayecto de certificación

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

J.1 Ejemplo 1: Utilización de constricciones básicas

Supongamos que Widget Corporation desea certificar recíprocamente la CA central de Acme Corporate Group, pero sólo desea que la comunidad Widget utilice certificados de entidad final expedidos por esa CA, no certificados expedidos por otras CA certificadas por esa CA.

Widget Corporation podrá satisfacer este requisito expidiendo un certificado para la CA central de Acme, que incluya el siguiente valor de campo de extensión:

Valor de campo de constricciones básicas:

{ cA TRUE, pathLenConstraint 0 }

J.2 Ejemplo 2: Utilización de constricciones de nombre

Supongamos que Widget Corporation desea certificar recíprocamente la CA central de Acme Corporation Group, pero sólo desea que la comunidad Widget utilice certificados de Acme para sujetos que satisfacen los siguientes criterios:

- en Acme Inc., en Estados Unidos de América, todos los sujetos son aceptables salvo los sujetos en el departamento de compras;
- en EuroAcme, en Francia, sólo los sujetos que están subordinados inmediatamente a la sede EuroAcme son aceptables (esto incluye a las personas que informan directamente a la sede pero excluye a los que informan a organizaciones subordinadas); y
- en Acme Ltd., en el Reino Unido, todos los sujetos son aceptables salvo los que informan a organizaciones que están subordinadas a la Unidad de Organización e Investigación y Desarrollo (esto incluye a las personas que informan directamente a Investigación y Desarrollo pero excluye a los que informan a unidades subordinadas de Investigación y Desarrollo).

Widget Corporation podrá satisfacer estos requisitos expidiendo un certificado para la CA central de Acme, que incluya los siguientes valores de campo de extensión:

Valor de campo de constricciones básicas:

{ cA TRUE }

Value of Name Constraints Field:

**{ permittedSubtrees {{base --Country=US, Org=Acme Inc.--},
 {base --Country=France, Org=EuroAcme--, maximum 1},
 {base --Country=UK, Org=Acme Ltd.--}},
 excludedSubtrees {{base --Country=US, Org=Acme Inc., Org. Unit=Purchasing--},
 {base --Country=UK Org=Acme Ltd., Org. Unit=R&D--, minimum 2}}}**

J.3 Ejemplo 3: Utilización de constricciones de correspondencia de políticas y de política

Supongamos que se requiere la siguiente certificación recíproca entre los Gobiernos de Canadá y de Estados Unidos de América:

- a) una CA del Gobierno de Canadá desea certificar la utilización de firmas del Gobierno de Estados Unidos de América con respecto a una política canadiense denominada *Can/Us-Trade*;
- b) el Gobierno de Estados Unidos de América tiene una política denominada *US/Can-Trade*, que el Gobierno canadiense está dispuesto a considerar equivalente a su política *Can/US-Trade*;
- c) el Gobierno de Canadá desea aplicar salvaguardas que requieren que todos los certificados de Estados Unidos de América indiquen explícitamente apoyo de la política y que inhiban la correspondencia con otras políticas dentro del dominio de Estados Unidos de América.

Una CA del Gobierno canadiense podrá expedir un certificado para una CA del Gobierno de Estados Unidos de América con los siguientes valores de campo de extensión:

Valor de campo de políticas de certificado:

{{ policyIdentifier -- *object identifier for Can/US-Trade* -- }}

Value of Policy Mappings Field:

**{{ issuerDomainPolicy -- *object identifier for Can/US-Trade* -- ,
subjectDomainPolicy -- *object identifier for US/Can-Trade* -- }}**

Value of PolicyConstraints Field:

**{{ policySet { -- *object identifier for Can/US-Trade* -- }, requireExplicitPolicy (0),
inhibitPolicyMapping (0)}}**

Anexo K

Enmiendas y corrigenda

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

Esta edición de la presente Especificación de directorio incluye las siguientes enmiendas:

- Enmienda 1 para extensiones de certificados.
- Enmienda 2 para mejora de la seguridad operacional del directorio.

Esta edición de la presente Especificación de directorio abarca los siguientes corrigenda técnicos que subsanan defectos indicados en los siguientes informes de defectos (algunas partes de algunas de los siguientes corrigenda técnicos pueden haber sido incluidos en las enmiendas que formaron esta edición de la presente Especificación del directorio):

- corrigendum técnico 1 (que incluye el informe de defectos 128);
- corrigendum técnico 2 (que incluye el informe de defectos 077, 078, 083, 084);
- corrigendum técnico 3 (que incluye el informe de defectos 080, 092, 100, 177);
- corrigendum técnico 1 de la versión 97 (que incluye los informes de defectos 183, 194).

Reemplazada por una versión más reciente

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	Transmisiones de señales radiofónicas, de televisión 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
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación