Unión Internacional de Telecomunicaciones

UIT-T X.509

(08/2005)

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

SERIE X: REDES DE DATOS, COMUNICACIONES DE SISTEMAS ABIERTOS Y SEGURIDAD

Directorio

Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Marcos para certificados de claves públicas y atributos

Recomendación UIT-T X.509

Unión Internacional de Telecomunicaciones

RECOMENDACIONES UIT-T DE LA SERIE X

REDES DE DATOS, COMUNICACIONES DE SISTEMAS ABIERTOS Y SEGURIDAD

REDES PÚBLICAS DE DATOS	
Servicios y facilidades	X.1–X.19
Interfaces	X.20–X.49
Transmisión, señalización y conmutación	X.50–X.89
Aspectos de redes	X.90–X.149
Mantenimiento	X.150–X.179
Disposiciones administrativas	X.180–X.199
INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Modelo y notación	X.200–X.209
Definiciones de los servicios	X.210–X.219
Especificaciones de los protocolos en modo conexión	X.220–X.229
Especificaciones de los protocolos en modo sin conexión	X.230–X.239
Formularios para declaraciones de conformidad de implementación de protocolo	X.240–X.259
Identificación de protocolos	X.260–X.269
Protocolos de seguridad	X.270–X.279
Objetos gestionados de capa	X.280–X.289
Pruebas de conformidad	X.290–X.299
INTERFUNCIONAMIENTO ENTRE REDES	
Generalidades	X.300–X.349
Sistemas de transmisión de datos por satélite	X.350–X.369
Redes basadas en el protocolo Internet	X.370–X.379
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400–X.499
DIRECTORIO	**X.500–X.599**
GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS	
Gestión de redes	X.600–X.629
Eficacia	X.630–X.639
Calidad de servicio	X.640–X.649
Denominación, direccionamiento y registro	X.650–X.679
Notación de sintaxis abstracta uno	X.680–X.699
GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Marco y arquitectura de la gestión de sistemas	X.700–X.709
Servicio y protocolo de comunicación de gestión	X.710–X.719
Estructura de la información de gestión	X.720–X.729
Funciones de gestión y funciones de arquitectura de gestión distribuida abierta	X.730–X.799
SEGURIDAD	X.800–X.849
APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Compromiso, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.889
Aplicaciones genéricas de la notación de sintaxis abstracta uno	X.890–X.899
PROCESAMIENTO DISTRIBUIDO ABIERTO	X.900–X.999
SEGURIDAD DE LAS TELECOMUNICACIONES	X.1000–

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

NORMA INTERNACIONAL ISO/CEI 9594-8
RECOMENDACIÓN UIT-T X.509

Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Marcos para certificados de claves públicas y atributos

Resumen

La presente Recomendación | Norma Internacional define un marco para certificados de clave pública y para certificados de atributo. Otros conjuntos de normas pueden utilizar estos marcos para perfilar su aplicación a infraestructuras de clave pública (PKI) y a infraestructuras de gestión de privilegios (PMI). Asimismo, la presente Recomendación | Norma Internacional define un marco para la prestación de servicios de autenticación por el directorio a sus usuarios. Describe dos niveles de autenticación: autenticación simple que utiliza 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 para proporcionar servicios seguros.

Orígenes

La Recomendación UIT-T X.509 fue aprobada el 29 de agosto de 2005 por la Comisión de Estudio 17 (2005-2008) del UIT-T por el procedimiento de la Recomendación UIT-T A.8. Se publica también un texto idéntico como Norma Internacional ISO/CEI 9594-8.

PREFACIO

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

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

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

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

NOTA

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

La observancia de esta Recomendación es voluntaria. Ahora bien, la Recomendación puede contener ciertas disposiciones obligatorias (para asegurar, por ejemplo, la aplicabilidad o la interoperabilidad), por lo que la observancia se consigue con el cumplimiento exacto y puntual de todas las disposiciones obligatorias. La obligatoriedad de un elemento preceptivo o requisito se expresa mediante las frases "tener que, haber de, hay que + infinitivo" o el verbo principal en tiempo futuro simple de mandato, en modo afirmativo o negativo. El hecho de que se utilice esta formulación no entraña que la observancia se imponga a ninguna de las partes.

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

ÍNDICE

Introducción

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. 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 certificados de clave pública. Dicho marco incluye la especificación de objetos de datos utilizada para presentar los propios certificados así como notificaciones de revocación para certificados expedidos en los que ya no se debe confiar. El marco de certificados de clave pública definido en esta Especificación, aunque define algunos componentes críticos de la infraestructura de claves públicas (PKI), no define una PKI en su totalidad. Sin embargo, esta Especificación proporciona las bases sobre las cuales se construirán las PKI y sus especificaciones.

Asimismo, esta Recomendación | Norma Internacional define un marco para certificados de atributo. Dicho marco incluye la especificación de objetos de datos utilizados para representar los propios certificados así como notificaciones de revocación para certificados expedidos en los que ya no se debe confiar. El marco de certificados de atributo definido en esta Especificación, aunque define algunos componentes críticos de la infraestructura de gestión de privilegios (PMI), no define una PMI en su totalidad. Sin embargo, esta Especificación proporciona las bases sobre las cuales se construirán las PMI y sus especificaciones.

También se definen objetos de información para alojar objetos de PKI y de PMI en el directorio y para comparar valores presentados con valores almacenados.

Esta Recomendación | Norma Internacional define también un marco para la prestación de servicios de autenticación por el directorio a sus usuarios.

Esta Recomendación | Norma Internacional proporciona los marcos básicos sobre los que otros grupos de normas y foros industriales pueden definir perfiles industriales. Muchas de las características definidas como optativas en dichos marcos pueden ser de uso obligatorio en ciertos entornos mediante los perfiles. Esta quinta edición revisa y mejora técnicamente la cuarta edición de esta Recomendación | Norma Internacional, pero no la sustituye. Las implementaciones pueden seguir alegando conformidad con la cuarta edición. Sin embargo, en algún punto, no se soportará la cuarta edición (es decir, los defectos informados ya no serán resueltos). Se recomienda que las implementaciones se conformen con esta quinta edición lo antes posible.

Esta quinta edición especifica las versiones 1, 2 y 3 de los certificados de clave pública y versiones 1 y 2 de las listas de revocación de certificados. Esta edición también especifica la versión 2 de los certificados de atributo.

La función de extensibilidad se empieza a ofrecer en una versión precedente con la versión 3 del certificado de clave pública y con la versión 2 de la lista de revocación de certificados, y se ha incorporado en el certificado del atributo desde el principio. Esta función se especifica en la cláusula 7. Previsiblemente, esta función se podrá utilizar para introducir mejoras en esta edición, en su caso, sin que sea necesario producir una nueva versión.

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

El anexo B, que es parte integrante de esta Recomendación | Norma Internacional, proporciona reglas para generar y procesar listas de revocación de certificados.

El anexo C, que no es parte integrante de esta Recomendación | Norma Internacional, proporciona ejemplos de expedición de CRL delta.

El anexo D, que no es parte integrante de esta Recomendación | Norma Internacional, proporciona ejemplos de definición de política de privilegios y de atributos de privilegio.

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

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

El anexo H, que no es parte integrante de esta Recomendación | Norma Internacional, ofrece orientación sobre las aplicaciones habilitadas en la PKI relativas al procesamiento de la política de certificados durante el proceso de validación del trayecto de certificados.

El anexo I, que no es parte integrante de esta Recomendación | Norma Internacional, ofrece orientación sobre el uso del bit contentCommitment en la extensión del certificado keyUsage.

El anexo J, que no es parte integrante de esta Recomendación | Norma Internacional, contiene una lista alfabética de las definiciones de elementos de información definidos en esta Especificación.

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: Marcos para certificados de claves públicas y atributos

SECCIÓN 1 – GENERAL

1 Alcance

Esta Recomendación | Norma Internacional trata algunos de los requisitos de seguridad en los ámbitos de autenticación, y otros servicios de seguridad mediante la introducción de un conjunto de marcos sobre los que se pueden basar servicios completos. De forma específica, esta Recomendación | Norma Internacional define marcos para:

- certificados de clave pública;
- certificados de atributo;
- servicios de autenticación.

El marco de certificado de clave pública definido en esta Recomendación | Norma Internacional incluye la definición de los objetos de información para la infraestructura de claves públicas (PKI, *public key infrastructure*), incluidos los certificados de clave pública y la lista de revocación de certificados (CRL, *certificate revocation list*). El marco de certificados de atributo incluye la definición de los objetos de información para la infraestructura de gestión de privilegios (PMI, *privilege management infrastructure*), incluidos los certificados de atributo y la lista de revocación de certificados de atributo (ACRL, *attribute certificate revocation list*). Esta Especificación también proporciona el marco para expedir, gestionar, utilizar y revocar certificados. Se incluye un mecanismo de extensibilidad en los formatos definidos para ambos tipos de certificado y para todos los esquemas de lista de revocación. Esta Recomendación | Norma Internacional también incluye un conjunto de extensiones normalizadas para cada uno de ellos, lo que se espera que sea útil en general en algunas aplicaciones de PKI y PMI. En esta Recomendación | Norma Internacional se incluyen los componentes de esquema, incluidos clases de objeto, tipos de atributo y reglas de concordancia para almacenar objetos PKI y PMI en el directorio. Se espera que otros grupos de normas (por ejemplo ISO TC 68, IETF etc.) definan otros elementos PKI y PMI más allá de estos marcos, tales como protocolos de gestión de claves y de certificados, protocolos de explotación, certificados adicionales y extensiones CRL.

El esquema de autenticación definido en esta Recomendación | Norma Internacional es genérico y se puede aplicar a una variedad de aplicaciones y de entornos.

El directorio utiliza certificados de clave pública y certificados de atributo. Asimismo en esta Recomendación | Norma Internacional se define el marco para la utilización de estas facilidades por el directorio. La tecnología de claves públicas, incluidos los certificados, la utiliza el directorio para permitir autenticación fuerte, explotación firmada y/o criptada y para el almacenamiento de datos firmados y/o criptados en el directorio. El directorio puede utilizar certificados de atributo para permitir el control de acceso basado en reglas. Aunque el marco para esto se incluye en esta Especificación, la definición completa de la utilización de directorio de estos marcos de estos marcos y de los servicios asociados proporcionados por el directorio y sus componentes se define en el conjunto completo de Especificaciones de directorio.

En el marco de servicios de autenticación, esta Recomendación | Norma Internacional también:

- especifica la forma de la información de autenticación contenida por el directorio;
- describe como se puede obtener la información de autenticación a partir del directorio;
- enuncia los supuestos formulados en cuanto a la formación y al emplazamiento de esta 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 como 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 entidad 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 se debe utilizar autenticación fuerte para proporcionar servicios seguros. No se pretende con ello establecer un marco general; no obstante, puede ser de uso general para aplicaciones en las que estas técnicas se consideren adecuadas.

La autenticación (y otros servicios de seguridad) sólo pueden suministrarse dentro del contexto de una política de seguridad definida. 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 utilizado por las aplicaciones para obtener credenciales del directorio es el protocolo de acceso al directorio (DAP, *directory access protocol),* especificado en la Rec. UIT-T X.519 | ISO/CEI 9594-5.

2 Referencias normativas

Las siguientes Recomendaciones y Normas Internacionales contienen disposiciones que, mediante la referencia 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 citadas a continuación. Los miembros de la CEI y de la ISO mantienen registros de las Normas Internacionales actualmente vigentes. La Oficina de Normalización de las Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones UIT-T actualmente vigentes.

2.1 Recomendaciones | Normas Internacionales idénticas

- Recomendación UIT-T X.411 (1999) | ISO/CEI 10021-4:2003, *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 (2005) | ISO/CEI 9594-1:2005, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Visión de conjunto de conceptos, modelos y servicios.*
- Recomendación UIT-T X.501 (2005) | ISO/CEI 9594-2:2005, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Modelos*.
- Recomendación UIT-T X.511 (2005) | ISO/CEI 9594-3:2005, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Definición del servicio abstracto*.
- Recomendación UIT-T X.518 (2005) | ISO/CEI 9594-4:2005, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Procedimientos para operación distribuida*.
- Recomendación UIT-T X.519 (2005) | ISO/CEI 9594-5:2005, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Especificaciones de protocolo*.
- Recomendación UIT-T X.520 (2005) | ISO/CEI 9594-6:2005, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Tipos de atributos seleccionados*.
- Recomendación UIT-T X.521 (2005) | ISO/CEI 9594-7:2005, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Clases de objetos seleccionadas*.
- Recomendación UIT-T X.525 (2005) | ISO/CEI 9594-9:2005, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Replicación*.
- Recomendación UIT-T X.530 (2005) | ISO/CEI 9594-10:2005, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Utilización de la gestión de sistemas para la administración del directorio*.
- Recomendación UIT-T X.660 (2004) | ISO/CEI 9834-1:2005, *Tecnología de la información – Interconexión de sistemas abiertos – Procedimientos para la operación de autoridades de registro para interconexión de sistemas abiertos: Procedimientos generales y arcos superiores del árbol de identificadores de objetos de ASN.1*.
- Recomendación UIT-T X.680 (2002) | ISO/CEI 8824-1:2002, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica*.

- Recomendación UIT-T X.681 (2002) | ISO/CEI 8824-2:2002, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de objetos de información*.
- Recomendación UIT-T X.682 (2002) | ISO/CEI 8824-3:2002, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de constricciones*.
- Recomendación UIT-T X.683 (2002) | ISO/CEI 8824-4:2002, *Tecnología de la información – Notación de sintaxis abstracta uno: Parametrización de especificaciones de la notación de sintaxis abstracta uno*.
- Recomendación UIT-T X.690 (2002) | ISO/CEI 8825-1:2002, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de las reglas de codificación básica, de las reglas de codificación canónica y de las reglas de codificación distinguida*.
- Recomendación UIT-T X.691 (2002) | ISO/CEI 8825-2:2002, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de las reglas de codificación compactada*.
- Recomendación UIT-T X.812 (1995) | ISO/CEI 101181-3:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad para sistemas abiertos: Marco de control de acceso*.
- Recomendación UIT-T X.813 (1996) | ISO/CEI 10181-4:1997, *Tecnología de la información – Interconexión de sistemas abiertos – Marcos de seguridad para sistemas abiertos: Marco de no rechazo*.
- 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 de 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*.

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. CCITT X.800 | 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) firma digital;
j) cifrado;
k) clave;
l) contraseña;
m) autenticación de entidad par;
n) simétrico (cifrado).

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

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

3.3.1 certificado de atributo (AC, *attribute certificate*): Estructura de datos, firmada digitalmente por una autoridad de atributo, que vincula algunos valores de atributo con información de identificación de su titular.

3.3.2 autoridad de atributo (AA, *attribute authority*): Autoridad que asigna privilegios expidiendo certificados de atributo.

3.3.3 lista de revocación de autoridad de atributo (AARL, *attribute authority revocation list*): Lista de revocación que contiene una lista de referencias para certificados de atributo expedidos por las AA, que la autoridad expedidora ya no considera válidos.

3.3.4 lista de revocación de certificado de atributo (ACRL, *attribute certificate revocation list*): Lista de revocación que contiene una lista de referencias para certificados de atributo que la autoridad expedidora ya no considera válidos.

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

3.3.6 autoridad: Entidad responsable de la expedición de certificados. En esta Especificación se definen dos tipos; la autoridad de certificación que expide certificados de clave pública y la autoridad de atributo que expide certificados de atributo.

3.3.7 certificado de autoridad: Certificado expedido a una autoridad (por ejemplo, puede ser a una autoridad de certificación o a una autoridad de atributo).

3.3.8 lista de revocación de certificados básica; CRL básica: CRL que se utiliza como base en la generación de una dCRL.

3.3.9 certificado de autoridad de certificación; certificado de CA: Certificado para una CA expedido por otra CA.

3.3.10 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 certificado 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.

3.3.11 declaración de práctica de certificación (CPS, *certification practice statement*): Declaración de las prácticas que aplica una autoridad de certificación para la expedición de certificados.

3.3.12 lista de revocación de certificados (CRL, *certificate revocation list*): Lista firmada que indica un conjunto de certificados que el expedidor de certificados ya no considera válidos. Además del término genérico CRL, se definen algunos tipos de CRL específicos para CRL que tratan ámbitos particulares.

3.3.13 usuario de certificado: Entidad que necesita conocer, con certidumbre, los atributos y o la clave pública de otra entidad.

3.3.14 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.

3.3.15 sistema que utiliza el certificado: Implementación de las funciones definidas en esta Especificación de directorio que son utilizadas por un usuario de certificado.

3.3.16 validación de certificado: Proceso para asegurar que un certificado era válido en un momento determinado, con posible inclusión de la construcción y el procesamiento de un trayecto de certificación, y que asegura que todos los certificados en dicho trayecto eran válidos (es decir, no habían caducado ni estaban revocados) en un determinado momento.

3.3.17 autoridad de certificación (CA, *certification authority*): Autoridad a la cual uno o más usuarios han confiado la creación y asignación de certificados de clave pública. Facultativamente, la autoridad de certificación puede crear las claves de los usuarios.

3.3.18 lista de revocación de autoridad de certificación (CARL, *certification authority revocation list*): Lista de revocación que incluye una lista de certificados de claves públicas expedidos por autoridades de certificación, a las que el expedidor del certificado ya no considera válidos.

3.3.19 trayecto de certificación: Secuencia ordenada de certificados de clave pública 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.

3.3.20 punto de distribución de lista de revocación de certificados: Asiento de directorio u otra fuente de distribución para las CRL; una CRL distribuida a través de un punto de distribución de CRL puede contener asientos de revocación sólo para un subconjunto del conjunto total de certificados expedidos por una autoridad de certificación o puede contener asientos de revocación para múltiples autoridades de certificación.

3.3.21 certificado cruzado: Clave pública o certificado de atributo en el que el expedidor y el sujeto/titular son respectivamente dos CA o dos AA diferentes. Las CA y las AA expiden respectivamente certificados cruzados a otras CA o AA como mecanismo para autorizar la existencia de la CA sujeto (por ejemplo, en una jerarquía estricta) o para reconocer la existencia de la CA sujeto o la AA titular (por ejemplo, en un modelo fiduciario distribuido). La estructura del certificado cruzado se utiliza en ambos casos.

3.3.22 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.

3.3.23 confidencialidad de los datos: Este servicio se puede utilizar para obtener la protección de los datos frente a buscadores no autorizados. El servicio de confidencialidad de datos está soportado por un marco de autenticación. Se puede utilizar para la protección contra la interceptación de datos.

3.3.24 delegación: Envío de un privilegio desde una entidad que tiene dicho privilegio a otra entidad.

3.3.25 trayecto de delegación: Secuencia ordenada de certificados que, junto con la autenticación de una identidad de asertor de privilegios, puede ser procesada para verificar la autenticidad de un privilegio de asertor de privilegios.

3.3.26 lista de revocación de certificados-delta (dCRL, *delta-CRL*): Lista de revocación de certificados parcial que contiene únicamente asientos para certificados cuyo estado de revocación ha sido modificado después de la expedición de la lista de revocación de certificados básica referenciada.

3.3.27 entidad final: Sujeto del certificado de clave pública que utiliza su clave privada para otros fines distintos que firmar certificados, o titular de certificado de atributo que utiliza sus atributos para obtener acceso a un recurso, o entidad que es una parte confiante.

3.3.28 lista de revocación de certificados de atributo de entidad final (EARL, *end-entity attribute certificate revocation list*): Lista de revocación que contiene una lista de certificados de atributo, expedida a los titulares que no sean también autoridades de atributo, que el expedidor de certificados ya no considera válidos.

3.3.29 lista de revocación de certificados de clave pública de entidad final (EPRL, *end-entity public-key certificate revocation list*): Lista de revocación que contiene una lista de certificados de clave pública, expedidos a sujetos que no sean también autoridades de certificación, que el expedidor de certificados ya no considera válidos.

3.3.30 variables ambientales: Aquellos aspectos de política necesarios para una decisión de autorización, que no estén contenidos en estructuras de estadística, pero de los que un verificador de privilegios disponga mediante algún medio local (por ejemplo, hora del día o balance de cuentas corrientes).

3.3.31 lista de revocación de certificados completa: Lista de revocación completa que contiene asientos para todos los certificados que han sido revocados en un ámbito determinado.

3.3.32 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.

3.3.33 titular: Entidad a la que se ha delegado algún privilegio ya sea directamente a partir de la fuente de autoridad o indirectamente a través de otra autoridad de atributo.

3.3.34 lista de revocación de certificados indirecta (iCRL, indirect CRL): Lista de revocación que contiene por lo menos información de revocación sobre certificados expedidos por autoridades distintas de la que expidió esta lista de revocación de certificados.

3.3.35 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).

3.3.36 método objeto: Acción que puede ser invocada en un recurso (por ejemplo, un sistema de archivos puede haber leído, escrito, ejecutado métodos objeto).

3.3.37 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 fácil computacionalmente.

3.3.38 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 certificación 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.

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

3.3.40 privilegio: Atributo o propiedad asignado a una entidad por una autoridad.

3.3.41 asertor de privilegios: Titular de un privilegio que utiliza su certificado de atributo o su certificado de clave pública para aseverar un privilegio.

3.3.42 infraestructura de gestión de privilegios (PMI, *privilege management infrastructure*): Infraestructura capaz de soportar la gestión de privilegios como soporte de un servicio de autorización completo y en relación con una infraestructura de claves públicas.

3.3.43 política de privilegios: Política que destaca condiciones para los verificadores de privilegios con el fin de proporcionar o realizar servicios relacionados con asertores de privilegios cualificados. La política de privilegios relaciona atributos asociados con el servicio, así como atributos asociados con asertores de privilegios.

3.3.44 verificador de privilegios: Entidad que verifica certificados a partir de una política de privilegios.

3.3.45 clave pública: (En un critposistema de claves públicas) clave de un par de claves de usuario que es conocida públicamente.

3.3.46 certificado de clave pública (PKC, *public-key certificate*): Clave pública de un usuario, junto con alguna otra información, hecha infalsificable por firma digital con la clave privada de la autoridad de certificación que la emitió.

3.3.47 infraestructura de claves públicas (PKI, *public key infrastucture*): Infraestructura capaz de soportar la gestión de claves públicas para los servicios de autenticación, criptación, integridad, o no repudio.

3.3.48 parte confiante: Usuario o agente que se fía de los datos de un certificado al tomar decisiones.

3.3.49 certificado de asignación de cometido: Certificado que contiene el atributo de cometido y asigna uno o más cometidos al sujeto/titular del certificado.

3.3.50 certificado de especificación de cometido: Certificado que contiene la asignación de privilegios a un cometido.

3.3.51 sensibilidad: Característica de un recurso que presupone su valor o importancia.

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

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

3.3.54 certificado de atributo autoexpedido: Certificado de atributo (AC) en el que el expedidor y el sujeto son la misma autoridad de atributo. Una autoridad de atributo podría utilizar un AC autoexpedido, por ejemplo, para publicar información de políticas.

3.3.55 certificado autoexpedido: Certificado de clave pública en el que el expedidor y el sujeto son la misma autoridad de certificación (CA). Una CA podría utilizar certificados autoexpedidos, por ejemplo, durante una operación de renovación de clave para pasar la confianza de la clave antigua a la clave nueva.

3.3.56 certificado autofirmado: Constituye un caso especial de certificados autoexpedidos en los que la clave privada utilizada por la autoridad de certificación (CA) para firmar el certificado corresponde a la clave pública que está certificada en el certificado. Una CA podría utilizar un certificado autofirmado, por ejemplo, para anunciar su clave pública u otra información sobre sus operaciones.

NOTA – La utilización de certificados autoexpedidos y autofirmados expedidos por entidades distintas a las autoridades de certificación queda fuera del alcance de esta Recomendación | Norma Internacional.

3.3.57 fuente de autoridad (SOA, *source of authority*): Autoridad de atributo en la que confía un verificador de privilegios para un recurso determinado como la autoridad última en asignar un conjunto de privilegios.

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

3.3.59 fiduciario: En general, se puede decir que una entidad acepta como "fiduciaria" a una segunda entidad cuando aquella (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 tendrá que estar segura de que puede confiar en que la autoridad de certificación crea solamente certificados válidos y fiables.

3.3.60 ancla de confianza: Se trata de un conjunto de la siguiente información adicional a la clave pública: identificador de algoritmo, parámetros de clave pública (si se aplican), nombre distinguido del titular de la clave privada asociada (es decir, la autoridad de certificación sujeto) y facultativamente un periodo de validez. El ancla de confianza puede presentarse en la forma de un certificado autofirmado. Un sistema que utiliza un certificado puede confiar en un ancla de confianza y puede aplicarla para validar certificados en los trayectos de certificación.

4 Abreviaturas

A los efectos de la presente Recomendación | Norma Internacional se aplican las siguientes siglas

AA	Autoridad de atributos (*attribute authority*)
AARL	Lista de revocación de autoridades de atributo (*attribute authority revocation list*)
AC	Certificado de atributos (*attribute certificate*)
ACRL	Lista de revocación de certificados de atributo (*attribute certificate revocation list*)
CA	Autoridad de certificación (*certification authority*)
CARL	Lista de revocación de autoridades de certificación (*certification authority revocation list*)
CRL	Lista de revocación de certificados (*certificate revocation list*)
dCRL	Lista de revocación de certificados delta (*delta certificate revocation list*)
DIB	Base de información de directorio (*directory information base*)
DIT	Árbol de información de directorio (*directory information tree*)
DSA	Agente de sistema de directorio (*directory system agent*)
DUA	Agente de usuario de directorio (*directory user agent*)
EARL	Lista de revocación de certificados de atributo de entidad final *(end-entity attribute certificate revocation list*)
EPRL	Lista de revocación de certificados de clave pública de entidad final (*end-entity public-key certificate revocation list*)
iCRL	Lista de revocación de certificados indirecta (*indirect cetificate revocation list*)
OCSP	Protocolo en línea del estado de certificado (o*n-line certificate status protocol*)
PKC	Certificado de clave pública (*public key certificate*)
PKCS	Criptosistema de claves públicas (*public key cryptosystem*)
PKI	Infraestructura de claves públicas (*public key infrastructure*)
PMI	Infraestructura de gestión de privilegios (*privilege management infrastructure*)
SOA	Fuente de autoridad (*source of authority*)

5 Convenios

Con pequeñas excepciones, esta Especificación de directorio se ha preparado con arreglo a las *Reglas de representación del texto común de UIT-T | ISO/CEI* de noviembre de 2001.

El término "Especificación de directorio" (como en "esta Especificación de directorio") se entenderá en el sentido de esta Rec. UIT-T X.509 | ISO/CEI 9594-8. El término "Especificaciones de directorio" se entenderá que designa a las Recomendaciones de la serie X.500 y a todas las partes de ISO/CEI 9594.

Esta Especificación de directorio utiliza el término *sistemas de la primera edición* para hacer referencia a los sistemas conformes a la primera edición 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 segunda edición* para hacer referencia a los sistemas conformes a la segunda edición 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. Esta Especificación de directorio utiliza el término *sistemas de la tercera edición* para hacer referencia a los sistemas conformes a la tercera edición de las Especificaciones de directorio, es decir, la edición de 1997 de las Recomendaciones UIT-T de la serie X.500 y la edición de ISO/CEI 9594:1998. Esta Especificación de directorio utiliza el término *sistemas de la cuarta edición* para referirse a los sistemas conformes a la cuarta edición de las Especificaciones de directorio, por ejemplo, las ediciones 2001 de las Recs. UIT-T X.500, X.501, X.511, X.518, X.519, X.520, X.521, X.525 y X.530, la edición 2000 de la Rec. UIT-T X.509 y las partes 1 a 10 de la edición ISO/CEI 9594:2001.

Esta Especificación de directorio utiliza el término *sistemas de la quinta edición* para referirse a los sistemas conformes a la quinta edición de las Especificaciones de directorio, es decir, las ediciones 2005 de las Recs. UIT-T X.500, X.501, X.509, X.511, X.518, X.519, X.520, X.521, X.525 y X.530 y las partes 1-10 de la edición ISO/CEI 9594:2005.

Esta Especificación de directorio presenta la notación ASN.1 con caracteres del tipo Helvetica en negritas. Cuando los tipos y valores ASN.1 aparecen en texto normal, se diferencian del texto normal presentándolos en el tipo Helvetica 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 Times en negrita. Los permisos de control de acceso se presentan en el tipo Times en cursivas.

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

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

Cuadro 1 – Notación

Notación	Significado
Xp	Clave pública de un usuario X.
Xs	Clave privada de X.
Xp[I]	Cifrado de alguna información, I, mediante la clave pública de X.
Xs[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(...n veces...(X)))
$X_1<<X_2>>$	Certificado del usuario X_2 emitido por la autoridad de certificación X_1.
$X_1<<X_2>>$ $X_2<<X_3>>$	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 $X1<<X_{n+1}>>$. Por ejemplo, la posesión de A<<B>>B<<C>> confiere la misma capacidad que A<<C>>, a saber, la aptitud para hallar Cp dada Ap.
$X_1p \circ X_1<<X_2>>$	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: Ap ° A<<B>> B<<C>> denota la operación de usar la clave pública de A para obtener la clave pública de B, Bp, de su certificado, seguido por el uso de Bp para desenvolver el certificado de C. El resultado de la operación es la clave pública de C, Cp.
A→B	Un trayecto de certificación de A a B, formado por una cadena de certificados, que comienza por CA(A)<<CA^2(A)>> y termina por CA(B)<<B>>.
NOTA – Cuando se introducen las notaciones, los símbolos X, X1, X2, etc., aparecen en lugar de los nombres de los usuarios, mientras que el símbolo I aparece en lugar de una información arbitraria.	

6 Visión de conjunto de los marcos

Esta Especificación define un marco para obtener una clave pública de una entidad y confiar en ella con el fin de que dicha entidad cripte información que ella misma debe descriptar, o para verificar la firma digital de dicha entidad. El marco incluye la emisión de un certificado de clave pública por una autoridad de certificación (CA, *certification authority*) y la validación de dicho certificado por el usuario del certificado. La validación incluye:

- el establecimiento un trayecto fiduciario de certificados entre el usuario del certificado y el sujeto del certificado;
- la verificación de las firmas digitales de cada certificado en el trayecto; y
- la verificación de todos los certificados a lo largo de dicho trayecto (es decir, verificar que no estaban caducados ni revocados en un momento determinado).

Esta Especificación define un marco para obtener atributos de privilegio de una entidad y confiar en ellos con el fin de determinar si están o no autorizados a acceder a un recurso determinado. El marco incluye la emisión de un certificado por una autoridad de atributo (AA, *attribute authority*) y la validación de dicho certificado por un verificador de privilegios. La validación incluye:

- el que se asegure que los privilegios en el certificado son suficientes cuando se les compara con la política de privilegios;
- el establecimiento de un trayecto de delegación fiduciario de certificados si es preciso;
- la verificación de la firma digital de cada certificado en el trayecto;
- el que se asegure que cada expedidor estaba autorizado a delegar privilegios; y
- el validar que los certificados no están caducados o no han sido revocados por sus expedidores.

Aunque PKI y PMI son infraestructuras separadas y se pueden establecer de forma independiente una de la otra, están relacionadas. Esta Especificación recomienda que se identifique a los titulares y a los expedidores de certificados de atributo en los propios certificados de atributo mediante punteros en sus correspondientes certificados de clave pública. La autenticación de los expedidores y de los titulares de certificados de atributo, para asegurar que las entidades que reclaman privilegios y expiden privilegios son las que dicen ser, se realiza utilizando los procesos normales de la PKI para autenticar identidades. Este proceso de autenticación no está duplicado en el marco de certificados de atributo.

6.1 Firmas digitales

Las firmas digitales se utilizan tanto en la PKI como en la PMI como el mecanismo mediante el cual la autoridad que expide un certificado certifica la vinculación en el certificado. En la PKI la firma digital de la CA expedidora en un certificado de clave pública certifica la vinculación entre el material de clave pública y el sujeto del certificado. En la PMI la firma digital de la AA expedidora certifica la vinculación entre los atributos (privilegios) y el titular del certificado. Esta subcláusula describe firmas digitales en general. Las secciones 2 y 3 de esta Especificación tratan de la utilización de firmas digitales específicamente en las PKI y PMI.

En esta subcláusula no se pretende especificar una norma para firmas digitales en general, sino especificar los medios para firmar los testigos en la PKI, la PMI y en el directorio.

La información (info) se firma añadiéndole un sumario cifrado de la información. El sumario se produce por medio de una función de troceo unidireccional, mientras que el cifrado se lleva a cabo usando la clave privada del firmante (véase la figura 1). Así:

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

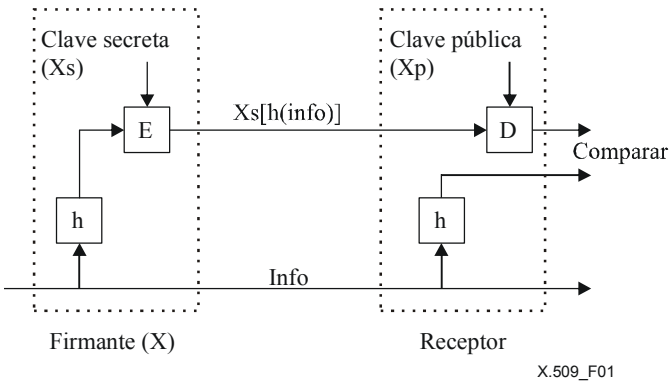


Figura 1 – Firmas digitales

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

El receptor de información firmada verifica la firma:

– aplicando la función de troceo unidireccional a la información;

– comparando el resultado con el obtenido descifrando la firma mediante la clave pública del firmante.

Esta Especificación no impone una sola función de troceo unidireccional para uso en firmado. Se pretende que el marco sea aplicable a cualquier función de troceo adecuada, y que por consiguiente soporte 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 tendrán que soportar la misma función de troceo 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 de troceo y el algoritmo de criptación utilizados para computar la firma digital.

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

```
ENCRYPTED { ToBeEnciphered } ::=   BIT STRING ( CONSTRAINED BY {
   -- shall 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 (2002) | ISO/CEI 8825-1:2002) 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 sincronización 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 requerir 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 elementos de datos se forma criptando una transformación abreviada o "troceada" del elemento, y puede describirse utilizando la siguiente macro ASN.1:

```
HASH {ToBeHashed}          ::=   SEQUENCE {
   algorithmIdentifier     AlgorithmIdentifier,
   hashValue               BIT STRING ( CONSTRAINED BY {
      -- shall be the result of applying a hashing procedure to the DER-encoded octets --
      -- of a value of -- ToBeHashed } ) }

ENCRYPTED-HASH { ToBeSigned }    ::=        BIT STRING ( CONSTRAINED BY {
   -- shall 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 se asocia 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 la Rec. UIT-T X.690 (2002) | ISO/CEI 8825-1:2002 con las siguientes limitaciones:

a) se utilizará la forma definida de codificación de longitud, codificada en el mínimo número de octetos;

b) para los tipos cadena, no se utilizará la forma construida de codificación;

c) si el valor de un tipo es su valor por defecto, deberá estar ausente;

d) los componentes de un tipo conjunto deberán codificarse en orden ascendente de su valor de rótulo;

e) los componentes de un tipo conjunto-de se codificarán en orden ascendente de su valor de octeto;

f) si el valor de un tipo booleano es verdadero, el octeto de contenido de la codificación deberá fijarse a "FF"16;

g) todo bit no utilizado en el octeto final de la codificación de un valor cadena de bits, si existe, deberá fijarse a cero;

h) el tipo real se codificará de una manera tal que no se utilicen las bases 8, 10 y 16, y el factor de escala binario será cero;

i) la codificación de un tiempo UTC se efectuará conforme a lo especificado en la Rec. UIT-T X.690 (2002) | ISO/CEI 8825-1:2002;

j) la codificación de un tiempo generalizado se efectuará conforme a lo especificado en la Rec. UIT-T X.690 (2002) | ISO/CEI 8825-1:2002.

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.

SECCIÓN 2 – MARCO DE CERTIFICADOS DE CLAVE PÚBLICA

El marco de certificados de clave pública definido aquí se utiliza para aplicaciones con requisitos de autenticación, integridad, confidencialidad y de no repudio.

La vinculación de una clave pública a una entidad la proporciona una autoridad mediante una estructura de datos firmados digitalmente denominada certificado de clave pública. El formato de los certificados de clave pública se define aquí, incluidos un mecanismo de extensibilidad y un conjunto de extensiones de certificado específicas. Si, por cualquier razón, una autoridad revoca un certificado de clave pública expedido con anterioridad, es preciso que los usuarios sean capaces de reconocer que se ha producido dicha revocación, de forma que no utilicen un certificado poco seguro. Las listas de revocación constituyen un esquema que se puede utilizar para notificar a los usuarios las revocaciones. Aquí se define el formato de las listas de revocación, incluidos un mecanismo de extensibilidad y un conjunto de extensiones de lista de revocación. Tanto en el caso del certificado como en el de la lista de revocación, otros entes pueden también definir extensiones adicionales que sean útiles a sus entornos específicos.

Un sistema que utiliza certificados de clave pública necesita validar un certificado antes de utilizar dicho certificado para una aplicación. Aquí también se definen procedimientos para realizar dicha validación, incluidos la verificación e integridad del propio certificado, su estado de revocación y su validez en relación con el uso que se pretende.

El directorio utiliza certificados de clave pública en la prestación de servicios de seguridad que incluyen:

- autenticación fuerte en los componentes de directorio y entre ellos;
- autenticación, integridad y confidencialidad de operaciones de directorio; así como
- integridad y autenticación de datos almacenados.

7 Claves públicas y certificados de clave pública

Con el fin de que un usuario sea capaz de confiar en una clave pública para otro usuario, por ejemplo, para autenticar la identidad de dicho usuario, la clave pública será obtenida de una fuente de confianza. Dicha fuente, denominada una autoridad de certificación (CA) certifica una clave pública, expidiendo un certificado de clave pública, que vincula la clave pública a la entidad que posee la clave privada correspondiente. Los procedimientos utilizados por una CA para asegurar que una entidad posee de hecho una clave privada y otros procedimientos relacionados con la expedición de certificados de clave pública están fuera del ámbito de esta Especificación. El certificado, cuya forma se especifica más adelante en esta cláusula, tiene las propiedades siguientes:

- cualquier usuario con acceso a la clave pública de la autoridad de certificación puede recuperar la clave pública que fue certificada;
- ninguna parte distinta de la autoridad de certificación puede modificar el certificado sin que se pueda detectar (los certificados no se pueden falsificar).

Debido a que los certificados no se pueden falsificar, se pueden publicar situándolos en el directorio sin necesidad de realizar ningún esfuerzo especial para protegerlos.

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

Una autoridad de certificación genera el certificado de un usuario firmando (véase 6.1) un conjunto de información, que incluye el nombre distinguido del usuario, una clave pública, así como un *identificador único* facultativo que contiene información adicional sobre el usuario. La forma exacta del contenido del identificador único no se especifica aquí y se deja a la autoridad de certificación pero podría ser, por ejemplo, un identificador de objeto, un certificado, una fecha o alguna otra forma de certificación en la validez del nombre distinguido. De forma específica, el certificado de un usuario con nombre distinguido A e identificador único UA, generado por la autoridad de certificación denominada CA y un identificador único UCA, tiene la forma siguiente:

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

donde V es una versión el certificado, SN es el número de serie del certificado, AI es el identificador del algoritmo autorizado para firmar el certificado, UCA es el identificador único facultativo de la CA, UA es el identificador único facultativo del usuario A, T^A indica el periodo de validez del certificado constituido por 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 información sobre el estado del certificado, es decir, publicará datos de revocación. Puesto que se supone que T^A se modificará en periodos no inferiores a 24 horas, se espera que los sistemas utilizarán el tiempo universal coordinado como base de tiempos de referencia. Cualquier usuario con el conocimiento de la CAp

podrá comprobar la validez de la firma en el certificado. Para representar certificados se pude utilizar el siguiente tipo de datos ASN.1:

```
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 shall be v2 or v3
    subjectUniqueIdentifier [2]     IMPLICIT UniqueIdentifier OPTIONAL,
                                    -- if present, version shall be v2 or v3
    extensions              [3]     Extensions OPTIONAL
                                    -- If present, version shall 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 de utilizar el valor de **Time** en cualquier operación de comparación, por ejemplo, como parte de una regla de concordancia en una búsqueda, y en el caso de que se haya elegido la sintaxis de **Time** como el tipo **UTCTime**, el valor del campo de año de dos cifras se debe racionalizar en un valor de año de cuatro cifras como sigue:

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

NOTA 2 – La utilización de **GeneralizedTime** puede impedir el interfuncionamiento con implementaciones que no tienen la posibilidad de elegir entre **UTCTime** o **GeneralizedTime**. Cuando se puede utilizar **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 ningún caso se utilizará **UTCTime** para representar fechas más allá del año 2049.

version es la versión del certificado codificado. Si está presente el componente **extensions** en el certificado, la versión será v3. Si el componente **issuerUniqueIdentifier** o **subjectUniqueIdentifier** estará presente la versión será v2 o v3.

serialNumber es un número entero asignado por la CA. El valor de **serialNumber** tendrá que ser único para cada certificado expedido por una determinada CA (es decir, el nombre del expedidor y el número de serie identifican a un único certificado).

signature contiene el identificador de algoritmo para el algoritmo y la función de troceo utilizados por la CA para firmar el certificado (por ejemplo, md5WithRSAEncryption, sha-1WithRSAEncryption, id-dsa-with-sha1, etc.)

issuer identifica la entidad que ha firmado y expedido el certificado.

validity es el intervalo del tiempo durante el cual la CA garantiza que mantendrá información sobre el estado del certificado.

subject identifica la entidad asociada con la clave pública que se encuentra en el campo de clave pública del sujeto.

subjectPublicKeyInfo se utiliza para encaminar la clave pública que se está certificando y para identificar el algoritmo del cual esta clave pública es un ejemplar de (por ejemplo, rsaEncryption, dhpublicnumber, id-dsa, etc.)

issuerUniqueIdentifier se utiliza para identificar unívocamente a un expedidor en el caso de reutilizar un nombre.

subjectUniqueIdentifier se utiliza para identificar unívocamente un sujeto en el caso de reutilizar un nombre.

NOTA 3 – En situaciones en las que la autoridad de denominación podría reasignar un nombre distinguido a un usuario diferente, las CA pueden utilizar el identificador único para distinguir entre ejemplares reutilizados. 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 para registrar usuarios.

El campo **extensions** permite la adición de nuevos campos a la estructura sin modificar la definición de ASN.1. Un campo de extensión está constituido por un identificador de extensión, una bandera de criticidad y una codificación de un valor de datos de un tipo ASN.1 asociado con la extensión identificada. Para aquellas extensiones en las que es importante la ordenación de extensiones individuales dentro de **SEQUENCE**, la especificación de dichas extensiones individuales incluirá las reglas sobre la importancia de su orden interno. Cuando una implementación que está procesando un certificado no reconoce una extensión, si la bandera de criticidad es **FALSO**, puede ignorar dicha extensión. Si la bandera de criticidad es **VERDADERO**, extensiones no reconocidas harán que la estructura se considere no válida, es decir, en un certificado, una extensión crítica no reconocida provocará el fracaso de la validación de una firma que utiliza dicho certificado. Cuando una implementación que utiliza el certificado reconoce y es capaz de procesar una extensión, la procesará independientemente del valor de la bandera de criticidad. Obsérvese que una extensión indicada como no crítica provocará un comportamiento discordante entre los sistemas que utilizan el certificado y procesan esta extensión, y los sistemas que utilizan el certificado pero no reconocen la extensión y la ignoran.

Si aparecen elementos desconocidos en la extensión, y la extensión no se indica como crítica, dichos elementos desconocidos se ignorarán de conformidad con las reglas de extensibilidad indicadas en 12.2.2 de la Rec. UIT-T X.519 | ISO/CEI 9594-5.

La CA tiene tres opciones para con una extensión:

i) puede excluir la extensión del certificado;

ii) puede incluir la extensión e indicarla como no crítica;

iii) puede incluir la extensión e indicarla como crítica.

Un motor de validación puede ejercer dos posibles acciones con respecto a una extensión:

i) puede ignorar la extensión y aceptar el certificado (sin que cambie nada más);

ii) puede procesar la extensión y aceptar o rechazar el certificado, según el contenido de la extensión y las condiciones en las que se realiza el procesamiento (por ejemplo, los valores actuales de las variables de procesamiento de trayecto).

Algunas extensiones sólo pueden indicarse como críticas. En estos casos, un motor de validación que entiende esta extensión la procesa, y la aceptación o el rechazo del certificado depende (al menos en parte) del contenido de la extensión. Un motor de validación que no entiende la extensión rechaza el certificado.

Algunas extensiones sólo pueden indicarse como no críticas. En estos casos, un motor de validación que entiende esta extensión la procesa, y la aceptación o el rechazo del certificado depende (al menos en parte) del contenido de la extensión. Un motor de validación que no entiende la extensión acepta el certificado (a menos que haya otros factores diferentes de esta extensión que obliguen a rechazarla).

Algunas extensiones pueden indicarse como críticas o no críticas. En estos casos, un motor de validación que entiende esta extensión la procesa y la aceptación o el rechazo del certificado depende (al menos en parte) del contenido de la extensión, independientemente de la bandera de criticidad. Un motor de validación que no entiende una extensión acepta el certificado si dicha extensión está indicada como no crítica (a menos que haya factores diferentes de esta extensión que obliguen a rechazarla) y rechaza el certificado si la extensión está indicada como crítica.

Cuando una CA considera la inclusión de una extensión en un certificado, lo hace con la esperanza de que se cumplirá su intención siempre que sea posible. Si es necesario considerar el contenido de la extensión para determinar la confianza en el certificado, la CA indicaría la extensión como crítica. Se hace así sabiendo que cualquier sistema de validación que no procese la extensión rechazará el certificado (esto tal vez limite el conjunto de aplicaciones que pueden verificar el certificado). La CA podría indicar algunas extensiones como no críticas para permitir la compatibilidad hacia atrás con aplicaciones de validación que no pueden procesar las extensiones. Cuando la necesidad de compatibilidad hacia atrás e interoperabilidad con aplicaciones de validación que no pueden procesar las extensiones sea más vital que la capacidad de la CA para imponer las extensiones, estas extensiones de criticidad facultativa se indicarían como no críticas. Es muy probable que las CA indiquen las extensiones opcionalmente críticas como no críticas durante un periodo de transición, hasta tanto que las aplicaciones de procesamiento de certificados del verificador se transformen en unas que puedan procesar las extensiones.

Se pueden definir extensiones específicas en las Recomendaciones UIT-T | Normas Internacionales o lo puede hacer cualquier organización que tenga esa necesidad. Se definirá el identificador de objeto que identifica una extensión de conformidad con la Rec. UIT-T X.660 | ISO/CEI 9834-1. En la cláusula 8 de esta Especificación de directorio se definen las extensiones normalizadas para certificados.

Se utiliza la siguiente clase de objeto para definir extensiones específicas:

```
EXTENSION ::= CLASS {
    &id     OBJECT IDENTIFIER UNIQUE,
    &ExtnType }
WITH SYNTAX {
    SYNTAX              &ExtnType
    IDENTIFIED BY       &id }
```

Hay dos tipos primarios de certificados de clave pública, certificados de entidad final y certificados de CA.

Un certificado de entidad final es un certificado expedido por una CA a un sujeto que no es un expedidor de otros certificados de clave pública.

Un certificado de CA es un certificado expedido por una CA a un sujeto que es una CA y que, por lo tanto, es capaz de expedir certificados de clave pública. Los certificados de CA pueden categorizarse en los tipos siguientes:

- Certificado autoexpedido – Certificado en el que el expedidor y el sujeto son la misma CA. Una CA podría utilizar certificados autoexpedidos, por ejemplo, durante una operación de renovación de clave para pasar la confianza de la clave antigua a la clave nueva.
- Certificado autofirmado – Constituye un caso especial de certificados autoexpedidos en los que la clave privada utilizada por la CA para firmar el certificado corresponde a la clave pública que está certificada en el certificado. Una CA podría utilizar un certificado autofirmado, por ejemplo, para anunciar su clave pública u otra información sobre sus operaciones.
- Certificado cruzado – Certificado en el que el expedidor y el sujeto son dos CA diferentes. Las CA expiden certificados a otras CA, ya sea como mecanismo para autorizar la existencia de la CA sujeto (por ejemplo, en una jerarquía estricta) o para reconocer la existencia de una CA sujeto (por ejemplo, en un modelo fiduciario distribuido). La estructura del certificado cruzado se utiliza en ambos casos. En algunas situaciones, los requisitos en conflicto o superpuestos de las constricciones, tales como las constricciones de nombre, pueden tener necesidad de una autoridad de certificación (CA) para que expida más de un certificado cruzado a otra CA.

El asiento de directorio de cada usuario A, que está participando en una autenticación fuerte, contiene el certificado o certificados de A. Este tipo de certificados lo genera una autoridad de certificación de A, que es una entidad del DIT. Una autoridad de certificación A, que puede no ser única, se denomina CA(A), o simplemente CA si se sobreentiende A. La clave pública de A puede entonces ser encontrada por cualquier usuario que conozca la clave pública de CA. El hecho de descubrir claves públicas es por tanto recursivo.

Si el usuario A, al intentar obtener la clave pública del usuario B, ya ha logrado la clave pública de CA(B), entonces el proceso está completo. Para permitir que A obtenga la clave pública de CA(B), el asiento de directorio de cada autoridad de certificación, X, contiene algunos certificados. Estos certificados son de dos tipos. En primer lugar hay certificados directos de X generados por otras autoridades de certificación. En segundo lugar hay certificados inversos generados por el propio X que son las claves públicas certificadas de otras autoridades de certificación. La existencia de estos certificados permite a los usuarios construir trayectos de certificación desde un punto a otro.

Una lista de certificados necesaria para permitir a un determinado usuario obtener la clave pública de otro se conoce como un *trayecto de certificación*. Cada elemento de la lista es un certificado de una autoridad de certificación del elemento siguiente en la lista. Un trayecto de certificación desde A a B (anotado como A→B):

- se inicia con un certificado generado por la CA(A), es decir, CA(A)<<X1>> para determinada entidad X1;

- continúa con otros certificados Xi<<Xi+1>>;
- finaliza con el certificado de B.

Los campos **issuer** (**titular**) y **subject** (**sujeto**) de cada certificado se emplean, en parte, para identificar un trayecto válido. Para cada par de certificados adyacentes en un trayecto de certificación válido, el valor del campo **subject** en un certificado debe concordar con el valor del campo **issuer** en el certificado subsiguiente. Además, el valor del campo **issuer** en el primer certificado debe concordar con el nombre distinguido (DN, *distinguished name*) del ancla de confianza. Cuando se comprueba la validez de un trayecto de certificación sólo se utilizan los nombres en estos campos. Los nombres en las extensiones del certificado no se emplean para esta finalidad. Un trayecto de certificación forma lógicamente una cadena continua de puntos fiduciarios 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→B y B→A puede variar. Una manera de facilitar esto es disponer de una jerarquía de autoridades de certificación, que puede coincidir o no con una parte o la totalidad de la jerarquía DIT. La ventaja de esto consiste en que los usuarios que tiene autoridades de certificación en la jerarquía pueden establecer un trayecto de certificación entre ellos, utilizando el directorio sin ninguna información previa. Para lograr esto, cada CA puede almacenar un certificado y un certificado inverso designados como correspondientes a una CA superior. La regla de concordancia **distinguishedNameMatch**, que se define en 13.5.2 de la Rec. UIT-T X.501|ISO/CEI 9594-2, debería utilizarse para comparar el nombre distinguido (DN) en el campo **issuer** de un certificado con el DN en el campo **subject** de otro certificado.

Un usuario puede obtener uno o más certificados de una o más autoridades de certificación. Cada certificado incluye el nombre de la autoridad de certificación expedidora. Se pueden utilizar los siguientes datos ASN.1 para representar certificados y un trayecto de certificación:

```
Certificates                ::=  SEQUENCE {
    userCertificate             Certificate,
    certificationPath           CertPath OPTIONAL }

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

Además, se puede utilizar el siguiente tipo de datos ASN.1 para representar el trayecto de certificación directo. Este componente incluye el trayecto de certificación que puede señalar hacia el originador.

```
CertPath  ::=  SEQUENCE OF CrossCertificates

CrossCertificates           ::=        SET OF Certificate

PkiPath   ::=  SEQUENCE OF Certificate
```

PkiPath es útil para representar un trayecto de certificación. En la secuencia, el orden de los certificados es tal que el sujeto del primer certificado es el titular del segundo certificado, etc.

Cada certificado de un trayecto de certificación debe ser único. Ningún certificado podrá aparecer más de una vez en un valor del componente **theCACertificates** de **CertificationPath,** ni en un valor de **Certificate** en el componente **CrossCertificates** de **CertPath** ni en un valor de **Certificate** en **PkiPath**.

7.1 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 de este marco. 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" 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:

a) 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.

b) El par de claves es generado por una tercera entidad. La tercera entidad tendrá 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.

c) 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.

7.2 Creación de certificados de clave pública

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

a) una autoridad de certificación tendrá que cerciorarse de la identidad de un usuario antes de crear un certificado para él;

b) una autoridad de certificación no expedirá certificados para dos usuarios con el mismo nombre.

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:

a) 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.

b) Si se emplea el medio de generación de los pares de claves de 7.1 b) o 7.1 c), la clave privada de usuario será transferida al usuario de manera segura.

c) Si se emplea el medio de generación de pares claves descrito en 7.1 a) o 7.1 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 de clave pública 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.

7.3 Validez de certificados

La autoridad que expide certificados (de clave pública o de atributo) también asume la responsabilidad de indicar la validez de los certificados que expide. En general, los certificados están sujetos a posibles revocaciones subsiguientes. Esta revocación y la notificación de la revocación las puede realizar directamente la misma autoridad que expidió el certificado o indirectamente otra autoridad debidamente autorizada por la autoridad que expidió el certificado. Una autoridad que emite certificados debe declarar, posiblemente mediante una disposición publicada de sus prácticas, mediante los propios certificados o mediante algún otro medio identificado, si:

– los certificados no se pueden revocar; o

– los certificados se pueden revocar directamente por la misma autoridad que expidió el certificado; o

– la autoridad que expide el certificado autoriza a una entidad diferente para que realice la revocación.

Las autoridades que revocan certificados deben declarar, mediante algún medio similar, qué mecanismo o mecanismos pueden utilizar las partes confiantes para obtener información sobre el estado de las revocaciones en certificados expedidos por dicha autoridad. Esta Especificación define un mecanismo de lista de revocación de certificados (CRL) pero no impide la utilización de mecanismos alternativos. Un mecanismo alternativo es el protocolo de estado de certificado en línea (OCSP, *online certificate status protocol*) que se describe en la norma RFC 2560[1] del IETF. Al aplicar este protocolo, una parte confiante (cliente) puede solicitar el estado de revocación de un certificado del servidor OCSP. El servidor puede utilizar CRL u otros mecanismos para comprobar el estado del certificado y responder al cliente en consecuencia. Si las partes confiantes pueden emplear OCSP para comprobar el estado de un certificado, la

[1] Norma RFC 2560 del IETF, X.509 *Internet Public Key Infrastructure Online Certificate Status Protocol (OCSP)*, junio de 1999.

norma RFC 3280[2] del IETF contiene una extensión de certificado (Acceso a la información de la autoridad) que podría incluirse en esos certificados y proporcionaría suficiente información para acceder a un servidor OCSP adecuado. Las partes confiantes comprueban la información del estado de revocación, según convenga, para todos los certificados considerados en el procedimiento de procesamiento de trayecto descrito en la cláusula 10 y en el procedimiento de procesamiento de trayecto por delegación descrito en la cláusula 16 para validar un certificado.

Sólo una autoridad de certificación (CA) que esté autorizada para expedir CRL, puede delegar esa autorización a otra entidad. Si se lleva a cabo, debe ser verificable en el momento de una verificación de certificado/CRL. La extensión **cRLDistributionPoints** puede utilizarse para esta finalidad. El campo **cRLIssuer** de esta extensión será rellenado con el nombre o nombres de las entidades, distintas del propio titular del certificado, autorizadas para expedir CRL relativas al estado de revocación del certificado en cuestión.

Los certificados, incluidos los certificados de clave pública así como los certificados de atributo, tendrán asociada cierta duración, al final de la cual caduca. A fin de asegurar la continuidad del servicio, la autoridad garantizará el suministro oportuno de certificados de sustitución que reemplacen a los caducados o próximos a caducar. La fecha de notificación de revocación es la fecha/hora en la que la notificación de revocación para un certificado aparece por primera vez en una CRL, independientemente de si se trata de una CRL básica o dCRL. En la CRL, la fecha de notificación de revocación es el valor contenido en el campo **thisUpdate**. La fecha de revocación es la fecha/hora en la que la CA revocó realmente el certificado, lo que puede ser diferente de la primera vez en que aparece en una CRL. En la CRL, la fecha de revocación es el valor contenido en el componente **revocationDate**. La fecha de no validez es la fecha/hora en la que se sabe o supone que la clave privada fue comprometida o que, por el contrario, el certificado debe considerarse no válido. Esta fecha puede ser anterior a la fecha de revocación. En la **CRL**, la fecha de no validez es el valor contenido en la extensión de inserción **invalidityDate**.

Hay dos cuestiones conexas:

- La validez de los certificados puede 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 la autoridad tenga que instalar y distribuir un gran número de certificados que pudieran agotarse en la misma fecha de caducidad.
- Los certificados caducados serán sacados del directorio. Es cuestión de política de seguridad y de responsabilidad de la autoridad 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 autoridad, o si se supone que el certificado de la autoridad ha quedado comprometido. La autoridad debe dar a conocer la revocación de un certificado de usuario o certificado de autoridad y se pondrá a disposición un nuevo certificado según proceda. La autoridad podrá entonces informar al poseedor del certificado sobre su revocación mediante algún procedimiento independiente.

Una autoridad que expide y revoca posteriormente certificados:

a) puede ser requerida para que mantenga un registro auditor de sus eventos de revocación para todos los tipos de certificados expedidos por dicha autoridad (por ejemplo, certificados de clave pública, certificados de atributo expedidos a entidades finales así como a otras autoridades);

b) proporcionará información del estado de revocación a las partes confiantes utilizando las CRL, protocolos de estado de certificados en línea o algún otro mecanismo para la publicación de la información de estado de revocación;

c) si utiliza CRL, mantendrá y publicará las CRL incluso cuando las listas de certificados revocados estén vacías;

d) si se emplean sólo CRL separadas, expedirá un conjunto completo de CRL separadas que cubran el conjunto completo de certificados cuyo estado de revocación será comunicado aplicando el mecanismo CRL. Por lo tanto, el conjunto completo de CRL separadas será equivalente a una CRL completa para el mismo conjunto de certificados, si el titular de la CRL no estaba utilizando CRL separadas.

Las partes confiantes pueden utilizar algunos mecanismos para ubicar la información de estado de revocación suministrada por una autoridad. Por ejemplo, puede existir un puntero en el propio certificado que dirige a la parte confiante a una ubicación en la que se suministra la información de revocación. Puede existir un puntero en una lista de revocación que dirige a la parte confiante a una ubicación diferente. La parte confiante puede ubicar información de revocación en un almacén (por ejemplo un directorio) o por otros medios que están fuera del ámbito de esta Especificación (por ejemplo, configurados localmente).

---

[2] Norma RFC 3280 del IETF, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, abril de 2002

El mantenimiento de asientos de directorio, afectados por las listas de revocación, de la autoridad, es responsabilidad del directorio y de 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.

Si se publican listas de revocación en el directorio, se mantienen en inserciones como atributos de los tipos siguientes:

- Lista de revocación de certificados.
- Lista de revocación de autoridades.
- Lista de revocación delta.
- Lista de revocación de certificados de atributo.
- Lista de revocación de autoridades de atributo.

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

version es la versión de la lista de revocación codificada. Si está presente el componente **extensions,** indicado como crítico mediante banderas, en la lista de revocación, la versión será v2. Si no está presente ningún componente **extensions,** indicado como crítico mediante banderas, en la lista de revocación, la versión puede estar ausente o presente como v2.

signature contiene el identificador de algoritmo para el algoritmo utilizado por la autoridad para firmar la lista de revocación.

issuer identifica la entidad que ha firmado y expedido la lista de revocación.

thisUpdate es la fecha/hora en la que se expidió esta lista de revocación.

nextUpdate, si está presente, indica la fecha/hora en la que será expedida la próxima lista de revocación de esta serie. La lista de revocación siguiente podría expedirse antes de la fecha indicada, pero no se expedirá en ningún caso después de la hora indicada.

revokedCertificates identifica certificados que han sido revocados. Los certificados revocados se identifican mediante sus números de serie. En caso de no revocarse en ninguno de los certificados cubiertos por esta CRL, se recomienda encarecidamente omitir el parámetro **revokedCertificates** de la CRL, en lugar de incluirlo en una **SEQUENCE** vacía.

crlExtensions, si está presente, contiene una o más extensiones CRL.

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

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

NOTA 3 – Si cualesquiera extensiones incluidas en una **CertificateList** 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 podrá estar ausente. Si **version** está ausente, esto puede permitir que una implementación que soporta únicamente la versión 1 de las CRL siga utilizando la CRL, si en su examen de la secuencia **revokedCertificates** de la CRL no encuentra una extensión. Una implementación que soporta la versión 2 (o una versión superior) de las CRL en ausencia de versión, puede también 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,** 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** 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 rama de certificados), esa extensión se indicará como crítica en el campo **crlExtensions** con independencia de donde 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 8 de esta Especificación de directorio.

Si aparecen elementos desconocidos en la extensión, y la extensión no está indicada como crítica, se ignorarán dichos elementos desconocidos de conformidad con las reglas de extensibilidad indicadas en 12.2.2 de la Rec. UIT-T X.519 | ISO/CEI 9594-5.

7.4 Repudio de una firma digital

Todo participante en un evento puede decidir ulteriormente repudiar cualquier cosa que haya sido firmada digitalmente por otro participante. Por ejemplo, se puede negar la participación en un establecimiento de clave o en el origen de un mensaje de correo electrónico firmado, de la misma manera que es posible negar haber firmado un documento para evitar la responsabilidad que implique su contenido. Puede ocurrir que el repudio no funcione. En el marco de no repudio de la Rec. UIT-T X.813 | ISO/CEI 10181-4, se describe el siguiente proceso de resolución de disputas:

1) generación de evidencia;
2) transferencia, almacenamiento y recuperación de evidencia;
3) verificación de evidencia; y
4) resolución de controversias.

La evidencia generada puede incluir, entre otras cosas:

– registro de auditoría relacionados con el evento y aserción de intento;
– declaraciones realizadas ante notarios terceras partes;
– declaraciones de política;
– información firmada digitalmente, que incluya registro de auditoría y declaraciones notariales;
– indicaciones de tiempo de la información firmada digitalmente;
– certificados que soportan la firma digital;
– la información de revocación adecuada publicada y disponible al momento de la controversia; y
– cualquiera otra revocación de certificado que haya ocurrido después del evento que pudiese indicar que el compromiso de clave se realizó antes de ese instante.

Es posible preservar de diferentes maneras la integridad de los datos almacenados de modo que pueda probarse como evidencia, por ejemplo, el control de acceso, el almacenamiento mediante funciones generadoras por parte de terceros de confianza de datos generados, firmas digitales. Puede también ocurrir que se deba aumentar periódicamente la protección de dichos datos almacenados a fin de contrarrestar las mejoras en el procesamiento por computador y/o el análisis criptográfico.

NOTA – Si bien esta Especificación de directorio no especifica ni el tipo y cantidad de evidencia generada ni el nivel de integridad, cabe esperar que será acorde con el riesgo involucrado.

Es posible que para la verificación de evidencia sea necesario revalidar las firmas digitales de datos, por ejemplo mensajes, documentos, certificados, CRL e indicaciones de tiempo utilizadas en el proceso de validación inicial. El hecho de que un certificado haya expirado no impide que se utilice para revalidar las firmas creadas durante su periodo de validez. Se puede utilizar un certificado revocado siempre y cuando se establezca que era válido en el momento de ocurrir el evento que origina la disputa.

Aun cuando toda esta evidencia descrita se considere técnicamente válida, puede ocurrir que el firmante repudie con éxito el mensaje basándose en otras condiciones, por ejemplo, el intento, la comprensión o las competencias del signatario.

8 Extensiones de certificados de clave pública y de CRL

Las extensiones de certificados definidas en esta cláusula se utilizarán con certificados de clave pública, mientras no se indique lo contrario. Las extensiones a utilizar con certificados de atributo se definen en la cláusula 15. Las extensiones de CRL definidas en esta cláusula se pueden utilizar en las CRL, CARL y también para las ACRL y AARL definidas en la cláusula 17.

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 soportan 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) *Constricciones 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 soportan 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 implementa ese tipo de extensión puede procesar el resto del certificado pasando por alto la extensión. Si una extensión se indica como no crítica, un sistema que utiliza el certificado y reconoce la extensión procesará 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 interfuncionar 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 ejemplar de cada tipo de extensión en cualquier certificado, CRL, o asiento de CRL, respectivamente.

8.1 Tratamiento de políticas

8.1.1 Política de certificados

Este marco de autenticación contiene tres tipos de entidades: el usuario de certificado, la autoridad de certificación y el sujeto de certificado (o entidad final). Cada entidad funciona sometida a obligaciones con las otras dos entidades y, en contrapartida, disfruta de garantías limitadas ofertadas por ellas. Estas obligaciones y garantías se definen en una política de certificados. Una política de certificados es un documento (normalmente en lenguaje claro). Se puede referenciar mediante un único identificador, que puede estar definido en la extensión de políticas de certificados del certificado expedido por la autoridad de certificación, a la entidad final y a la que está ligado el usuario de certificados. Se puede expedir un certificado de conformidad con una o más de una política. Una autoridad de políticas realiza una definición de la política y asigna el identificador. El conjunto de políticas administradas por una autoridad de políticas se denomina un dominio de políticas. Todos los certificados se expiden de acuerdo con una política, incluso cuando la

política no está registrada en ningún sitio, ni referenciada en el certificado. Esta Especificación no prescribe el estilo ni el contenido de la política de certificados.

El usuario de certificado puede estar ligado a sus obligaciones en la política de certificados mediante el acto de importar una clave pública de autoridad y utilizando ésta como un ancla de confianza, o vinculándose a un certificado que incluya el identificador de política asociado. La autoridad de certificación puede estar ligada a sus obligaciones mediante la política por el hecho de expedir un certificado que incluya el identificador de política asociado. Además, la entidad final puede estar ligada a sus obligaciones en la política mediante el acto de solicitar y aceptar un certificado que incluya el identificador de política asociado y utilizando la clave privada correspondiente. Implementaciones que no utilicen la extensión de políticas de certificados lograrán la vinculación necesaria mediante otros medios.

El que una entidad declare sencillamente su conformidad a una política no satisface normalmente los requisitos de garantía de las demás entidades en el marco. Necesitan alguna razón para creer que las otras partes realizan una implementación fiable de la política. Sin embargo, si está establecido explícitamente en la política. los usuarios de certificado pueden aceptar las garantías de la autoridad de certificación de que sus entidades finales están de acuerdo en estar ligadas a sus obligaciones mediante la política, sin tener que confirmarlo directamente con ellas. Este aspecto de la política de certificados está fuera del ámbito de esta Especificación.

Una autoridad de certificación puede establecer limitaciones en la utilización de sus certificados para controlar el riesgo que asume al expedir certificados. Por ejemplo, puede restringir la comunidad de usuarios de certificado, los propósitos para los cuales pueden utilizar sus certificados y/o el tipo y cantidad de daños que puede asumir en el caso de un fallo por su parte, o aquellos de sus entidades finales. Estos asuntos deben definirse en la política de certificados.

Se puede incluir información adicional en la extensión de política de certificados en forma de calificadores de política para ayudar a las entidades afectadas a entender las disposiciones de la política.

8.1.2 Certificación cruzada

Una autoridad de certificación puede ser el sujeto de un certificado expedido por otra autoridad de certificación. En este caso, el certificado se denomina un certificado cruzado, la autoridad de certificación que es el sujeto del certificado se denomina la autoridad de certificación sujeto y la autoridad de certificación que expide el certificado cruzado la autoridad de certificación intermedia (véase la figura 2). Tanto el certificado cruzado como el certificado de la entidad final pueden contener una extensión de políticas de certificados.

Las garantías y obligaciones compartidas por una autoridad de certificación sujeto, por la autoridad de certificación intermedia y por el usuario de certificado se definen mediante la política de certificados identificada en el certificado cruzado, en cumplimento de lo cual la autoridad de certificación sujeto puede actuar como, o en nombre de, una entidad final. Además, las garantías y obligaciones compartidas por el sujeto de certificado, la autoridad de certificación sujeto y la autoridad de certificación intermedia se definen mediante la política de certificados identificada en un certificado de entidad final, según el cual una autoridad de certificación intermedia puede actuar como, o en nombre de, un usuario de certificado.

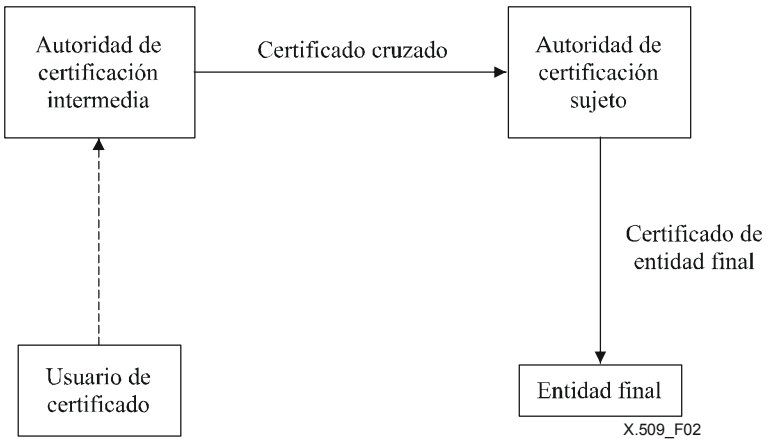


Figura 2 – Certificación cruzada

Un trayecto de certificación se considera válido mediante el conjunto de políticas que son comunes a todos los certificados en el trayecto.

Una autoridad de certificación intermedia puede, a su vez, ser el sujeto de un certificado expedido por otra autoridad de certificación, creando así trayectos de certificación de longitudes superiores a dos certificados. Además, puesto que la

confianza se degrada al crecer los trayectos de certificados en longitud, se precisan controles para asegurar que los certificados de entidad final con un nivel de confianza asociado inaceptablemente bajo serán rechazados por el usuario de certificado. Esto forma parte de la función del procedimiento de procesamiento de trayecto de certificación.

Además de la situación descrita anteriormente, deben considerarse dos casos especiales:

a) la autoridad de certificación no utiliza la extensión de certificados para transmitir sus requisitos de política a los usuarios de certificado; y

b) el usuario de certificado o la autoridad de certificación intermedia delega la tarea de controlar la política a la siguiente autoridad en el trayecto.

En el primer caso, el certificado no contendrá ninguna extensión de política de certificados. Por ello, el conjunto de políticas que validan el trayecto será nulo. Aunque, a pesar de todo, el trayecto puede ser válido. Los usuarios de certificado tendrán que seguir asegurándose de que están utilizado el certificado de conformidad con las políticas de las autoridades en el trayecto.

En el segundo caso, el usuario de certificado o la autoridad de certificación intermedia incluirán el valor especial *cualquier política* en el *conjunto de políticas inicial* o un certificado cruzado. Cuando un certificado incluya el valor especial *cualquier política*, no podrá incluir ningún otro identificador de política de certificados. El identificador *cualquier política* no tendrá ningún calificador de política asociado.

El usuario de certificado puede asegurar que todas sus obligaciones se envían de conformidad con la norma fijando el indicador *política explícita inicial*. De esta forma, sólo se aceptan en el trayecto las autoridades que utilizan la extensión de política de certificados normalizada como forma de lograr la vinculación y los usuarios de certificado no tienen obligaciones adicionales. Puesto que las autoridades también asumen obligaciones cuando actúan como, o en nombre de, un usuario de certificado, pueden asegurar que todas sus obligaciones se envían de conformidad con la norma incluyendo **requireExplicitPolicy** en el certificado cruzado.

8.1.3 Correspondencia de políticas

Algunos trayectos de certificación pueden cruzar fronteras entre dominios de políticas. Las garantías y obligaciones según la cuales se expide el certificado cruzado pueden ser materialmente equivalentes a alguna o a todas las garantías y obligaciones según las cuales la autoridad de certificación sujeto expide certificados a entidades finales, incluso cuando las autoridades de política bajo las que operan las dos autoridades de certificación puedan haber seleccionado identificadores únicos diferentes para dichas políticas materialmente equivalentes. En este caso, la autoridad de certificación intermedia puede incluir una extensión de correspondencias de políticas en el certificado cruzado. En la extensión de correspondencias de políticas, la autoridad de certificación intermedia asegura al usuario de certificado que seguirá disfrutando de las garantías habituales y que debe seguir cumpliendo sus obligaciones habituales, incluso cuando entidades subsiguientes en el trayecto de certificación operan en un dominio de políticas diferente. La autoridad de certificación intermedia incluirá una o más correspondencias para cada una de las políticas de un subconjunto bajo las que expidió el certificado cruzado, y no deberá incluir correspondencias para ninguna otra política. Si una o más de las políticas de certificados según las cuales opera la autoridad de certificación sujeto es idéntica a aquellas según las cuales opera la autoridad de certificación intermedia (es decir, tiene el mismo identificador único), entonces estos identificadores serán excluidos de la extensión de correspondencia de políticas, pero se incluirán en la extensión de políticas de certificados.

La correspondencia de políticas convierte todos los identificadores de políticas en certificados a lo largo del trayecto de certificación hasta el identificador de la política equivalente, como lo reconoce el usuario de certificado.

Las políticas no se corresponderán con el valor especial *cualquier política*.

Los usuarios de certificado pueden determinar que no se pueden vincular a certificados expedidos en un dominio de política distinto de su propio dominio, incluso cuando una autoridad de certificación intermedia fiable pueda determinar su política como materialmente equivalente a su propia política. Puede hacer esto fijando la *entrada de inhibición de correspondencia de políticas inicial* en el procedimiento de validación del trayecto. Además, una autoridad de certificación intermedia puede tomar una determinación similar en nombre de sus usuarios de certificado. Para asegurar que los usuarios de certificado cumplen correctamente este requisito, puede poner **inhibitPolicyMapping** en una extensión de constricciones de política.

8.1.4 Procesamiento de trayecto de certificación

El usuario de certificado debe elegir entre dos estrategias:

a) puede requerir que el trayecto de certificación sea válido mediante por lo menos una de un conjunto de políticas predeterminado por el usuario; o

b) puede solicitar al módulo de validación de trayecto que indique el conjunto de políticas para el cual el trayecto de certificación es válido.

La primera estrategia puede ser la más adecuada cuando el usuario de certificado conoce, *a priori*, el conjunto de políticas aceptables para el uso que se pretende.

La segunda estrategia puede ser la más adecuada cuando el usuario de certificado desconoce, *a priori*, el conjunto de políticas aceptables para el uso que se pretende.

En primer lugar, el procedimiento de validación del trayecto de certificación indicará que el trayecto sólo es válido si es válido bajo una o más de las políticas especificadas en el *conjunto de políticas inicial*, y devolverá el subconjunto del *conjunto de políticas inicial* para el cual el trayecto es válido. En segundo lugar, el procedimiento de validación del trayecto de certificación puede indicar que el trayecto no es válido mediante el *conjunto de políticas inicial*, pero si es válido mediante un conjunto diferente: el *conjunto de políticas constreñidas por las autoridades*. Entonces, el usuario de certificado tendrá que determinar si la utilización pretendida del certificado está de acuerdo con una o más de las políticas de certificados para las que el trayecto es válido. Al incluir el *conjunto de políticas inicial* en *cualquier política*, el usuario de certificado puede lograr que el procedimiento devuelva un resultado válido si el trayecto es válido mediante cualquier política (sin especificar).

8.1.5 Certificados autoexpedidos

Existen tres circunstancias bajo las cuales una autoridad de certificación puede expedir un certificado a sí mismo:

a) como forma adecuada de codificar la clave pública asociada con la clave privada empleada para firmar el certificado, de manera que pueda ser comunicada y almacenada como anclas de confianza por sus sistemas que utilizan certificado;

b) para certificar claves públicas adicionales de la CA utilizadas para fines distintos de los previstos por la categoría a tales como OCSP y posiblemente firma de CRL; y

c) para reemplazar su propio certificado expirado.

Estos tipos de certificado se denominan certificados autoexpedidos y se pueden reconocer mediante el hecho de que los nombres de expedidor y de sujeto que constan son idénticos. Para fines de validación del trayecto, los certificados autoexpedidos de categoría a) son certificados autofirmados y por lo tanto se verifican con la clave pública contenida en ellos, y si se encuentran en el trayecto deberán ignorarse.

Los certificados autoexpedidos del tipo b) sólo pueden aparecer como certificados finales en el trayecto y deben ser procesados como certificados finales.

Los certificados autoexpedidos del tipo c) (también conocidos como certificados intermedios autoexpedidos) pueden aparecer como certificados intermedios en el trayecto. En la práctica, cuando se sustituye una clave que se encuentra a punto de caducidad, una autoridad de certificación solicitará la expedición de cualquier certificado cruzado ligado que precise para su clave pública de sustitución antes de utilizar la clave. No obstante, si se encuentran certificados autoexpedidos de esta categoría en el trayecto, se procesarán como certificados intermedios, con la siguiente excepción: no contribuyen a la longitud del trayecto para fines de procesamiento del componente **pathLenConstraint** de la extensión **basicContraints** y de los valores *salto de certificados* asociados con los indicadores *inhibición de correspondencia de políticas pendiente* y *política explícita pendiente*.

Si una autoridad aprovecha la misma clave para firmar certificados y CRL, deberá utilizarse un solo certificado autoexpedido con categoría a). Si por el contrario utiliza una clave para firmar las CRL que es diferente a la que utiliza para firmar los certificados, esa autoridad puede autoexpedir dos certificados con categoría a), uno para cada una de las claves. En esta situación, los usuarios de los certificados necesitarán acceder a ambos certificados autoexpedidos para establecer anclas de confianza separadas para los certificados y las CRL firmados por esa autoridad. Alternativamente, una autoridad podrá expedir un certificado autoexpedido con categoría a) para la firma del certificado y otro con categoría b) para la firma de la CRL. En esta situación, los usuarios de los certificados utilizarán la clave certificada en el certificado de categoría a) como su ancla de confianza única para ambos certificados y las CRL firmadas por esa autoridad. En este caso, si el certificado autoexpedido de categoría b) fuera a ser utilizado para verificar las firmas en las CRL, no habría medios definidos en esta norma para comprobar la validez de ese certificado.

Si en un trayecto se encuentran certificados autoexpedidos de categoría b), deberán ignorarse.

NOTA – Otros mecanismos para la distribución de claves públicas de CA quedan fuera del alcance de esta Especificación.

8.2 Extensiones de información de claves y de política

8.2.1 Requisitos

Los requisitos siguientes están relacionados con información de claves y de políticas:

a) La actualización de pares de claves de CA puede ocurrir a intervalos regulares o en circunstancias especiales. Un campo de certificado necesita enviar un identificador de la clave pública para verificar la

firma del certificado. Un sistema que utiliza certificados puede usar estos identificadores para encontrar el certificado de CA correcto y validar la clave pública del expedidor del certificado.

b) En general, un sujeto de certificado tiene claves públicas diferentes y, en consecuencia, certificados diferentes para fines diferentes, por ejemplo firma digital y acuerdo de clave de cifrado. Se necesita un campo de certificado para ayudar a un usuario de certificado en la selección del certificado correcto para un determinado sujeto y un objetivo particular o para permitir a una CA que estipule que una clave de certificado sólo se puede utilizar para un determinado fin.

c) La actualización de un par de claves de sujeto puede ocurrir a intervalos regulares o en circunstancias especiales. Un campo de certificado necesita enviar un identificador para distinguir entre diferentes claves públicas para el mismo sujeto utilizado en distintos instantes en el tiempo. Un sistema que utiliza certificados puede usar estos identificadores para encontrar el certificado correcto.

d) Normalmente se utiliza la clave privada correspondiente a una clave pública certificada durante un periodo diferente a partir de la validez de la clave pública. En claves de firma digital, el periodo de utilización para una clave privada que firma es normalmente más corto que el de la clave pública que verifica. El periodo de validez del certificado indica un periodo durante el que se puede utilizar la clave pública, que no es necesariamente el mismo que el periodo de utilización de la clave privada. Si surge un compromiso de clave privada, el periodo de exposición se puede limitar si el verificador de firmas conoce el periodo de utilización legitimado. Existe por lo tanto un requisito para indicar el periodo de utilización de la clave privada en un certificado.

e) Puesto que los certificados se pueden utilizar en entornos en los que se aplican múltiples políticas de certificados, es necesario establecer disposiciones para incluir información de política de certificados en los certificados.

f) En el caso de certificados cruzados desde una organización a otra, se puede acordar algunas veces que alguna de las dos políticas de las organizaciones se considere equivalentes. Un certificado de CA necesita permitir al expedidor de certificado que indique que una de sus propias políticas de certificado es equivalente a otra política de certificado dominio de la CA sujeto. Esto se conoce como correspondencia de políticas.

g) Un usuario de un sistema de cifrado o de una firma digital que utiliza certificados definidos en esta Especificación de directorio necesita ser capaz de determinar por adelantado los algoritmos soportados por otros usuarios.

8.2.2 Campos de extensión de certificado de clave pública 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) *Utilización de clave extendida.*

e) *Periodo de utilización de clave privada.*

f) *Políticas de certificados.*

g) *Correspondencias de políticas.*

Estos campos de extensión se utilizarán únicamente como extensiones de certificado, salvo para el identificador de clave de autoridad que también se puede utilizar como una extensión CRL. A menos que se indique lo contrario, estas extensiones se pueden utilizar tanto en certificados de CA como en certificados de entidad final.

8.2.2.1 Extensión de identificador de clave de autoridad

Este campo, que se puede utilizar como una extensión de certificado o una extensión de CRL, identifica la clave pública que debe utilizarse para verificar la firma en este certificado o CRL. Permite distinguir distintas claves utilizadas por la misma CA (por ejemplo, como ocurre en la 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 se puede identificar mediante un identificador de clave explícito en el componente **keyIdentifier**, mediante la identificación de un certificado para la clave (indicando el expedidor de certificado en el componente **authorityCertIssuer** y el número de serie de certificado en el componente **authorityCertSerialNumber**), o mediante el identificador de clave explícito y la identificación de un certificado para la clave. Si se utilizan ambas formas de identificación, el expedidor de certificado o de CRL tendrá que asegurarse de que son coherentes. Un identificador de clave será único en relación con todos los identificadores de clave para la autoridad expedidora, para el certificado o CRL que contenga la extensión. No se requiere una implementación que soporte esta extensión para ser capaz de procesar todas las formas de nombre en el componente **authorityCertIssuer**. (Véase 8.3.2.1 para más detalles sobre el tipo **GeneralNames**).

Las autoridades de certificación asignarán números de serie de certificado de forma que cada par (expedidor, número de serie de certificado) identifique de forma unívoca un único certificado. La forma **keyIdentifier** puede utilizarse para seleccionar certificados de CA durante la construcción del trayecto. El par **authorityCertIssuer**, **authoritySerialNumber** sólo puede utilizarse para proporcionar preferencia a un certificado en relación con otros durante la construcción del trayecto.

Esta extensión siempre es no crítica.

8.2.2.2 Extensión de identificador de clave de sujeto

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

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

SubjectKeyIdentifier ::= KeyIdentifier
```

Un identificador de clave será único en relación con todos los identificadores de claves para el sujeto con el que se utiliza. Esta extensión siempre es no crítica.

8.2.2.3 Extensión de utilización de claves

Este campo identifica la utilización para la cual se emitió el certificado. Esta utilización inicial puede verse restringida aún más por la política. Es posible declarar esta política en una definición de política de certificado, un contrato u otra especificación. No obstante, una política no puede pasar por alto la restricción indicada por un bit **KeyUsage**, es decir una política de certificado no puede permitir que el certificado sea utilizado para una firma digital si **KeyUsage** indica que sólo puede hacerse a través de un acuerdo de claves.

El hecho de fijar un valor específico de **KeyUsage** en un certificado no indica, en un ejemplar de comunicación, que las partes actúan de conformidad con dicho valor, por ejemplo al firmar un documento. Los métodos de definición mediante los cuales las partes pueden señalar su disposición a establecer un ejemplar particular de comunicación (por ejemplo, el compromiso relativo al contenido de dicho ejemplar particular) queda fuera del alcance de esta Especificación de directorio, pero se puede prever que habrá varios métodos. Si bien no se recomienda, es posible utilizar el contenido del certificado, por ejemplo la política de certificado, para señalar la intención de firmar. No obstante, puesto que la señal se hizo cuando la CA expidió certificado, es probable que esta utilización no cumpla con el requisito de que la declaración de intención se haga en el momento de la firma del signatario.

Se puede fijar más de un bit en un ejemplar de la extensión **keyUsage**. Al fijar múltiples bit no cambiará el significado de cada uno de ellos sino que se indicará que es posible utilizar el certificado a todos los efectos indicados por los bits en cuestión. Es posible que esta operación presente varios riesgos. En el anexo I se presenta un resumen de ellos.

Este campo se define del modo siguiente:

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

KeyUsage ::= BIT STRING {
    digitalSignature        (0),
    contentCommitment       (1),
    keyEncipherment         (2),
    dataEncipherment        (3),
    keyAgreement            (4),
```

```
keyCertSign          (5),
cRLSign              (6),
encipherOnly         (7),
decipherOnly         (8) }
```

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

a) **digitalSignature**: para verificar firmas digitales que se utilizan con un servicio de autenticación de entidad, de autenticación de origen de datos y/o de integridad;

b) **contentCommitment**: para verificar las firmas digitales cuyo propósito es señalar que el signatario compromete con contenido firmado. La CA puede restringir aún más la utilización del certificado a fin de soportar el tipo de compromiso, por ejemplo a través de una política de certificado. El tipo exacto de compromiso del signatario, por ejemplo "revisado y aprobado" o "con la intención de restringirlo" puede indicarse en el contenido firmado, por ejemplo el propio documento firmado o alguna otra información firmada.

   Puesto que la firma de un compromiso de contenido se considera como una transacción firmada digitalmente, no es necesario fijar en el certificado bit **digitalSignature**. No obstante, de hacerlo no se afectará el nivel de compromiso que el firmante adquiere al firmar el documento.

   Cabe observar que si bien no es incorrecto referirse a este bit **keyUsage** utilizando el identificador **nonRepudiation**, sin embargo se desaconseja. Independientemente de qué identificador se utilice, la semántica de este bit es la especificada en la presente Especificación de directorio.

c) **keyEncipherment**: para claves de cifrado u otra información de seguridad, por ejemplo, para transporte de claves.

d) **dataEncipherment**: para cifrar datos de usuario, pero no claves ni otra información de seguridad como en el apartado c) anterior;

e) **keyAgreement**: para su utilización como una clave de acuerdo de clave pública.

f) **keyCertSign**: para verificar una firma de CA en certificados.

   Puesto que la CA considera la firma de un certificado como un compromiso con el contenido del certificado, no es necesario fijar en el certificado ni el bit **digitalSignature** ni el bit **contentCommitment**. Si se fija uno de ellos, o ambos, no se afecta el nivel de compromiso que el signatario ha adquirido al firmar el certificado.

g) **cRLSign**: para verificar una firma de autoridad en las CRL.

   Al considerarse que la firma de una CRL implica que el emisor de la CRL se compromete con su contenido, no es necesario fijar en el certificado ni el bit **digitalSignature** ni el bit **contentCommitment**. Si uno de éstos, o ambos, se fija esto no afecta el nivel de compromiso que el signatario adquiere al firmar la CRL.

h) **encipherOnly**: clave de acuerdo de clave pública para su utilización sólo en datos de cifrado utilizados con el bit **keyAgreement** también fijado (el significado con otro bit de utilización de clave fijado no está definido).

i) **decipherOnly**: clave de acuerdo de clave pública para su utilización únicamente en datos de descifrado utilizados con el bit **keyAgreement** también fijado (el significado con otro bit de utilización de clave fijado no está definido).

Se debería indicar en las especificaciones de aplicación cuál de los bits **digitalSignature** o **contentCommitment** son adecuados para la utilización. Cuando una aplicación que firma conozca la intención del firmante respecto a su compromiso con el contenido, dicha aplicación firmará y soportará dicha firma mediante un certificado que tenga el bit **digitalSignature** fijado en su extensión **keyUsage**.

Aun cuando se haya verificado una firma digital a través de un certificado que tenga solamente el bit **digitalSignature** fijado, es posible que haya otros factores externos a la verificación de la firma digital que cumplan una función a la hora de determinar la intención de quien firma. Por otra parte, aunque se haya verificado una firma digital utilizando un certificado que tenga solamente el bit **contentCommitment** fijado, quien firma puede argüir factores externos para negar su compromiso con el contenido firmado.

El bit **keyCertSign** se utiliza solamente en los certificados de la CA. Si **KeyUsage** está fijado a **keyCertSign**, el valor de la componente cA de la extensión **basicConstraints** se fijará a **TRUE**. Las CA también pueden usar otro de los bits de utilización de clave definidos en **KeyUsage**, por ejemplo **digitalSignature** para proporcionar autenticación e integridad en transacciones de administración en línea.

Esta extensión puede, ser o no crítica, a opción del expedidor del certificado.

Si la extensión se indica como crítica o no crítica mediante banderas, pero el sistema que utiliza certificado la reconoce, se utilizará el certificado solamente cuando se haya fijado a uno el bit de utilización de clave. Si la extensión se indica como no crítica mediante una bandera y el sistema que utiliza certificado no lo reconoce, se hará caso omiso de esta extensión. Cuando se fija a cero un bit quiere decir que la clave no está destinada a dicho propósito. Si la extensión está presente y todos sus bits fijados a cero, la clave está destinada a algún propósito diferente de los ya mencionados.

8.2.2.4 Extensión de utilización de clave extendida

Este campo indica uno o más fines para los cuales se puede utilizar la clave pública, además o en lugar de los fines básicos indicados 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
```

Una CA puede afirmar cualquier extensión de utilización de clave aplicando el identificador **anyExtendedKeyUsage**. Esto permite a la CA expedir un certificado que contenga los OID para utilizaciones de claves extendidas que puedan ser requeridas por aplicaciones que emplean certificado, sin restringir el certificado a esas utilizaciones de clave solamente. Si la extensión de utilización de clave restringiera la utilización de claves, la inclusión de este OID eliminaría esa restricción.

```
anyExtendedKeyUsage  OBJECT IDENTIFIER  ::= { 2 5 29 37 0 }[S9]
```

Cualquier organización que lo necesite puede definir los fines de las claves. Los identificadores de objeto utilizados para identificar los fines de las claves se asignarán de conformidad con la Rec. UIT-T X.660 | ISO/CEI 9834-1.

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 únicamente para uno de los fines indicados.

Si la extensión se indica como no crítica mediante banderas, indica el fin o fines pretendidos para la clave, y se puede utilizar para encontrar la clave o el certificado correcto de una entidad que tiene múltiples claves o certificados. Si esta extensión está presente y el sistema que utiliza el certificado reconoce y procesa el tipo de extensión **extendedKeyUsage**, este sistema se asegurará de que el certificado sólo se utilice para uno de los fines especificados. (Aplicaciones de uso puede necesitar sin embargo que se indique un fin determinado para que los certificados sean aceptables para dicha aplicación.)

Si el certificado contiene tanto un campo de utilización de clave crítico como un campo de utilización de clave extendida crítico, ambos campos se procesarán de forma independiente y el certificado sólo se utilizará para un fin coherente con ambos campos. Si no existe un fin coherente para ambos campos, entonces el certificado no se utilizará para ningún fin.

Esta Especificación define la siguiente finalidad de clave que podrá incluirse en la extensión de utilización de clave. Otros fines que también podrían incluirse se definen en diversas especificaciones, tales como la norma RFC 3280 del IETF.

```
keyPurposes             OBJECT IDENTIFIER  ::=    {ds 38 1}
```

8.2.2.5 Extensión de periodo de utilización de clave privada

Este campo indica el periodo para la utilización de la clave privada correspondiente a la clave pública certificada. Aplicable únicamente 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 la hora a partir de las cuales se podría utilizar la clave privada para firmas. Si el componente **notBefore** no está presente, no se proporciona ninguna información de cuando empieza el periodo de utilización válido de la clave privada. El componente **notAfter** indica la fecha y hora hasta la que se podría utilizar la clave privada para firmas. Si el componente **notAfter** no está presente, no se proporciona ninguna información de cuando concluye en periodo de utilización válido de la clave privada.

Esta extensión siempre es no crítica.

NOTA 1 – El periodo de utilización válido de la clave privada puede ser diferente de la validez certificada de la clave pública, como se indica en el periodo de validez del certificado. Con claves de firma digital, el periodo de utilización para la clave privada que firma es normalmente más corto que el de la clave pública que verifica.

NOTA 2 – Si el verificador de una firma digital quiere comprobar que la clave no ha sido revocada, por ejemplo, debido al compromiso de clave hasta el instante de la verificación, entonces, puede existir todavía un certificado para la clave pública en el instante de verificación. Después de que el certificado o certificados para una clave pública hayan caducado, un verificador de firmas no puede confiar en compromisos que se han notificado a través de las CRL.

8.2.2.6 Extensión de políticas de certificados

Este campo enumera las políticas de certificados, reconocidas por una CA exterior, que aplican al certificado, junto con información de calificador facultativa relativa a estas políticas de certificado. La lista de políticas de certificados se utiliza para determinar la validez de un trayecto de certificación, según se describe en la cláusula 10. Los calificadores facultativos no se utilizan en el procedimiento de procesamiento de trayecto de certificación, pero se proporcionan calificadores importantes como una salida a dicho proceso para el certificado que utiliza la aplicación para ayudar en la determinación de que un trayecto es adecuado para una determinada transacción. Normalmente, políticas de certificado diferentes estarán relacionadas con aplicaciones diferentes que pueden utilizar la clave certificada. La presencia de esta extensión en un certificado de entidad final indica las políticas de certificado para las que este certificado es válido. La presencia de esta extensión en un certificado expedido por una CA a otra CA indica las políticas de certificado para las que los trayectos de certificación que contienen este certificado pueden ser válidos. 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 transmite información de calificador para una política de certificado. El componente **policyIdentifier** contiene un identificador de una política de certificados y el componente **policyQualifiers** contiene valores del calificador de política para dicho elemento.

Esta extensión puede, a opción del expedidor del certificado, ser crítica o no crítica.

Si la extensión se indica como crítica mediante banderas, indica que el certificado sólo se utilizará para el fin de conformidad con las reglas indicadas por una de las políticas de certificado indicadas. Las reglas de una determinada política pueden requerir que el sistema que utiliza certificados procese el valor del calificador de una forma particular.

Si la extensión se indica como no crítica mediante banderas, la utilización de esta extensión no limita necesariamente la utilización del certificado a las políticas enumeradas. Sin embargo, un usuario de certificado puede necesitar que esté presente una determinada política con objeto de utilizar el certificado (véase la cláusula 10). Los calificadores de política pueden, a opinión del usuario del certificado, ser procesados o ignorados.

Cualquier organización que lo necesite puede definir las políticas de certificados y los tipos de calificador de política de certificado. Los identificadores de objeto utilizados para identificar las políticas de certificado y los tipos de calificador de política de certificado se asignarán de conformidad con la Rec. UIT-T X.660 | ISO/CEI 9834-1. Una CA puede afirmar cualquier política utilizando el identificador **anyPolicy** para confiar en un certificado para todas las políticas posibles. Debido a la necesidad de identificación de este valor especial para que aplique, independientemente de la aplicación o del entorno, se asigna ese identificador de objeto en esta Especificación. No se asigna ningún identificador de objeto en esta Especificación para políticas de certificado específicas. Dicha asignación es la responsabilidad de la entidad que define la política de certificado.

```
anyPolicy       OBJECT IDENTIFIER  ::=          { 2 5 29 32 0 }
```

El identificador **anyPolicy** no debe tener ningún calificador de política asociado.

La clase de objeto ASN.1 siguiente se utiliza para la definición de 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] }
```

Una definición de un tipo de calificador de políticas incluirá:

- una declaración de las semántica y de los valores posibles; y
- una indicación de si el identificador de calificador puede aparecer en una extensión de políticas de certificado sin un valor que le acompañe, y, en este caso, la semántica implicada.

NOTA – Se puede especificar un calificador como cualquier tipo ASN.1. Cuando se sabe que el calificador se utilizará fundamentalmente con aplicaciones que no tienen funciones de codificación ASN.1, se recomienda que se especifique el tipo **OCTET STRING**. El valor ASN.1 de **OCTET STRING** puede entonces transmitir un valor de calificador codificado de conformidad con cualquier convenio especificado por el elemento de política que define la organización.

8.2.2.7 Extensión 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 certificado que contiene este certificado, pueda considerarse una de las políticas de certificado del expedidor equivalente a una política de certificado diferente utilizada en el dominio de CA sujeto. 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 se reconoce en el dominio de la CA expedidora y que puede considerarse equivalente a la política de certificado indicada en el componente **subjectDomainPolicy** que está reconocida en el dominio de la CA sujeto.

Las políticas no se harán corresponder con el valor especial **anyPolicy**.

Esta extensión puede, a opinión del expedidor de certificado, ser crítica o no crítica. Se recomienda que sea crítica, en otro caso un usuario de certificado puede interpretar de forma incorrecta la estipulación de la CA expedidora.

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 U.S. 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 conexa. 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 sólo en entornos limitados en los cuales las declaraciones de política son muy simples.

8.3 Extensiones de información de sujeto y de expedidor

8.3.1 Requisitos

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

a) Los certificados tendrán 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.

8.3.2 Campos de extensión de certificado y de CRL

Se definen los siguientes campos de extensión:

a) *Nombre alternativo de sujeto*.

b) *Nombre alternativo de expedidor*.

c) *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.

8.3.2.1 Extensión 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. UIT-T 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 soporta esta extensión no tendrá 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 soportada. 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.

8.3.2.2 Extensión 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 soporta esta extensión tendrá que ser capaz de procesar todas las formas de nombres. Si la extensión se indica como crítica mediante banderas, 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 soportada. 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.

8.3.2.3 Extensión 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 puede ser crítica o no crítica, a opción del expedidor del certificado. No se necesita que un sistema que utilice certificados procese esta extensión para comprender todos los tipos de atributos incluidos en ella. Si la extensión se señala como crítica, el certificado que habrá de aceptarse debe entender al menos uno de los tipos de atributos contenidos en la misma. Si la extensión se señala como crítica y no se entiende ninguno de los tipos de atributos contenidos, se rechaza el certificado.

Si esta extensión está presente en un certificado de clave pública, podrán estar presentes también algunas de las extensiones definidas en la cláusula 15.

8.4 Constricciones de trayecto de certificación

8.4.1 Requisitos

Para el procesamiento del trayecto de certificación:

a) Los certificados de la entidad final tendrán 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.

b) Una CA tendrá 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 tendrá que ser automáticamente verificable por el usuario del certificado.

c) El procesamiento del trayecto de certificación tendrá 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 implementar el procesamiento del trayecto de certificación sin depender de interacciones en tiempo real con el usuario local.

e) Debe ser posible implementar 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 tendrán que funcionar en entornos en los cuales se reconocen múltiples políticas de certificado. Una CA tendrá que ser capaz de estipular en cuáles CA de otro dominio confía y para qué fines. Hay que soportar 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 tendrán que ser compatibles hacia atrás con el sistema de trayecto de certificación sin constricciones especificado en ediciones anteriores de la Rec. UIT-T X.509 | ISO/CEI 9594-8.

j) Una CA tendrá 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 tendrán que conformarse con declaraciones de seguridad específicas. Por ejemplo, un requisito de seguridad pudiera indicar que el procesamiento del trayecto de certificación deberá 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:

– 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

– 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.

k) Una CA tendrá que ser capaz de evitar que el valor especial cualquier política sea considerado una política válida en certificados subsiguientes en un trayecto de certificación.

8.4.2 Campos de extensión de certificado

Se definen los siguientes campos de extensión:

a) *Constricciones básicas.*

b) *Constricciones de nombre.*

c) *Constricciones de política.*

d) *Inhibición de cualquier 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 G figuran ejemplos de la utilización de estas extensiones.

8.4.2.1 Extensión 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 constricción 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. La constricción comienza a surtir efecto con el siguiente certificado en el trayecto. La constricción restringe la longitud del segmento del trayecto de certificación entre el certificado que contiene esta extensión y el certificado de la entidad final. No influye en el número de certificados de CA en el trayecto de certificación entre el ancla de confianza y el certificado que contiene esta extensión. En consecuencia, la longitud de un trayecto de certificación completo puede exceder la longitud máxima del segmento constreñido por esta extensión. La constricción controla el número de certificados de CA no autoexpedidos entre el certificado CA que contiene la constricción y el certificado de la entidad final. Por lo tanto, la longitud total de este segmento del trayecto, excluyendo los certificados autoexpedidos, puede ser superior al valor de la constricción hasta en dos certificados. (Esto incluye los certificados en los dos puntos extremos del segmento más los certificados de CA entre los dos puntos extremos que están constreñidos por el valor de esta extensión.)

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, o se indica como no crítica pero es reconocida por el sistema que utiliza el certificado, 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 certificado debe ser considerado como certificado de entidad final y no puede utilizarse 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.

8.4.2.2 Extensión de constricciones de nombre

Este campo, que se utilizará solamente en un certificado de CA, indica un espacio de nombre dentro del cual deberán 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-nameConstraint }

NameConstraintsSyntax ::= SEQUENCE {
    permittedSubtrees       [0]   GeneralSubtrees OPTIONAL,
    excludedSubtrees        [1]   GeneralSubtrees OPTIONAL,
    requiredNameForms       [2]   NameForms 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)

NameForms  ::= SEQUENCE {
    basicNameForms    [0]   BasicNameForms OPTIONAL,
    otherNameForms    [1]   SEQUENCE SIZE (1..MAX) OF OBJECT IDENTIFIER OPTIONAL }
(ALL EXCEPT ({ -- none; i.e., at least one component shall be present -- }))

BasicNameForms  ::= BIT STRING {
    rfc822Name                (0),
    dNSName                   (1),
    x400Address               (2),
    directoryName             (3),
    ediPartyName              (4),
    uniformResourceIdentifier (5),
    iPAddress                 (6),
    registeredID              (7) }  (SIZE (1..MAX))
```

Si están presentes, cada uno de los componentes **permittedSubtrees** y **excludedSubtrees** especifica uno o más subárboles de denominación, cada uno definido por el nombre de la raíz del subárbol y, facultativamente, dentro de ese subárbol, una zona que está limitada por un nivel superior, por un nivel inferior, o por ambos niveles. Si **permittedSubtrees** está presente, los nombres de sujeto dentro de estos subárboles son aceptables. Si **excludedSubtrees** está presente, todo certificado expedido por la CA sujeto o por CA subsiguientes en el trayecto de certificación que tiene un nombre de sujeto dentro de estos subárboles es inaceptable. Si ambos subárboles, **permittedSubtrees** y **excludedSubtrees**, están presentes y los espacios de nombres se superponen, tiene preferencia la instrucción de exclusión para los nombres en la región en que existe tal superposición. Si no están especificados subárboles permitidos ni subárboles excluidos para una forma de nombre, es aceptable cualquier nombre dentro de esa forma de nombre. Si está presente **requiredNameForms**, todos los certificados subsiguientes en el trayecto de certificación deben incluir un nombre de al menos una de las formas de nombre requeridas.

Si está presente **permittedSubtrees**, se aplica lo siguiente a todos los certificados subsiguientes en el trayecto. Si cualquier certificado contiene un nombre de sujeto (en el campo **subject** o en la extensión **subjectAltNames**) de una forma de nombre para la cual se especifican subárboles permitidos, el nombre debe estar dentro de al menos uno de los subárboles especificados. Si cualquier certificado contiene solamente nombres de sujeto de formas de nombre distintas de aquellas para las cuales están especificados subárboles permitidos, no se requiere que los nombres de sujeto caigan dentro de alguno de los subárboles especificados. Por ejemplo, supóngase que se especifican dos subárboles permitidos, uno para la forma de nombre DN y el otro para la forma de nombre rfc822, no se especifican subárboles excluidos, pero se especifica **requiredNameForms** con el bit **directoryName** y el bit **rfc822Name** presentes. Un certificado que contuviera solamente nombres distintos de un nombre de directorio o de un nombre rfc822 sería inaceptable. Sin embargo, si no se especificara **requiredNameForms**, tal certificado sería aceptable. Por ejemplo, supóngase que se especifican dos subárboles permitidos, uno para la forma de nombre DN y el otro para la forma de nombre rfc822, no se especifican subárboles excluidos, y no está presente **requiredNameForms**. Un certificado que contuviera solamente un DN y en el que el DN estuviera dentro del subárbol permitido especificado, sería aceptable. Un certificado que contuviera tanto un nombre DN como un nombre rfc822 y en el que solamente uno de ellos está dentro de su subárbol permitido especificado sería inaceptable. Un certificado que contuviera solamente nombres distintos de un nombre DN o de un nombre rfc822 sería también aceptable.

NOTA – Este ejemplo sólo tiene fines ilustrativos. En esta Recomendación | Norma Internacional no se define cómo tratar los nombres que tienen las formas de nombre del tipo **GeneralName**, excepto la forma de nombre **directoryName**, en su estructura jerárquica.

Si está presente **excludedSubtrees**, todo certificado expedido por la CA sujeto o por CA subsiguientes en el trayecto de certificación, que tenga un nombre de sujeto (en el campo **subject** o en la extensión **subjectAltNames**) dentro de estos subárboles es inaceptable. Por ejemplo, supóngase que se especifican dos subárboles excluidos, uno para la forma de nombre DN y el otro para la forma de nombre rfc822. Un certificado que contuviera solamente un DN y en el que el DN estuviera dentro del subárbol excluido especificado sería inaceptable. Un certificado que contuviera tanto un nombre DN como un nombre rfc822 y en el que al menos uno de ellos está dentro de su subárbol excluido especificado sería inaceptable.

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 sujeto del certificado, si no es nulo), se probará la coherencia de todos esos nombres con una constricción de nombre de esa forma de nombre.

Si está presente **requiredNameForms**, todos los certificados subsiguientes en el trayecto de certificación deben incluir un nombre de sujeto de al menos una de las formas de nombre requeridas.

De las formas de nombre disponibles mediante el tipo **GeneralName**, solamente las formas de nombre que tienen una estructura jerárquica bien definida se pueden utilizar en los campos **permittedSubtrees** y **excludedSubtrees**. La forma de nombre **directoryName** satisface este requisito; cuando se utiliza esta forma de nombre, un subárbol de denominación corresponde a un subárbol del DIT.

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

El campo **maximum** especifica el nivel inferior de la zona dentro del subárbol. Los nombres cuyo último componente está por debajo del nivel especificado no están contenidos en la zona. Un valor de **maximum** de cero corresponde a la base, es decir, a 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.

Para la forma de nombre **directoryName**, un **certificate** se considera subordinado a la **base** (y por consiguiente candidato a pertenecer al subárbol) si la **SEQUENCE** de los **RDN**, que forma el **DN** completo en **base**, es idéntica a la **SEQUENCE** inicial del mismo número de **RDN** que constituyen la primera parte del **DN** en el campo **subject** del **certificate**. El **DN** en el campo **subject** del **certificate** puede tener **RDN** finales adicionales en su secuencia que no aparecen en el **DN** en **base**. La regla de concordancia **distinguishedNameMatch** es útil para comparar el valor de **base** con la secuencia inicial de los **RDN** en el **DN** en el campo **subject** del certificado.

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

Las implementaciones conformes no están obligadas a reconocer todas las formas de nombre posibles.

Si la extensión está presente y está señalada como crítica, una implementación que emplea certificados debe reconocer y procesar todas las formas de nombre para las cuales hay una especificación de subárbol (permitido o excluido) en la extensión y un valor correspondiente en el campo **subject** o en la extensión **subjectAltNames** de cualquier certificado subsiguiente en el trayecto de certificación. Si aparece una forma de nombre no reconocida en una especificación de subárbol y en un certificado subsiguiente, ese certificado se debe tratar como si se hubiera encontrado una extensión crítica no reconocida. Si cualquier nombre de sujeto en el certificado está dentro de un subárbol excluido, el certificado es inaceptable. Si un subárbol está especificado para una forma de nombre que no está contenida en ningún certificado subsiguiente, ese subárbol puede ser ignorado. Si el componente **requiredNameForms** especifica solamente formas de nombre no reconocidas, ese certificado se debe tratar como si se hubiera encontrado una extensión crítica no reconocida. De lo contrario, al menos una de las formas de nombre reconocidas debe aparecer en todos los certificados subsiguientes en el trayecto.

Si la extensión está presente y está señalada como no crítica, y una implementación que emplea certificado no reconoce una forma de nombre utilizada en cualquier componente **base**, esa especificación de subárbol puede ser ignorada. Si la extensión está señalada como no crítica y la implementación que emplea certificados no reconoce alguna de las formas de nombre especificadas en el componente **requiredNameForms**, el certificado se tratará como si estuviese ausente el componente **requiredNameForms**.

Obsérvese que en algunos casos podrá ser necesario que una CA expida a otra CA más de un certificado a fin de alcanzar los resultados deseados si algunos de los requisitos de constricciones de nombre entran en conflicto. Por ejemplo, supóngase que la empresa Acme tiene 20 sucursales en los Estados Unidos.

La empresa Widget desea certificar de manera cruzada la CA principal de la empresa Acme, pero quiere que la comunidad Widget sólo aplique los certificados de Acme únicamente para los sujetos que cumplen los siguientes criterios:

- Para las sucursales 1 a 19 de la empresa Acme, todas las secciones son aceptables como sujetos;
- Para la sucursal 20 de la empresa Acme, ninguna de las secciones es aceptable como sujeto excepto para el sujeto de la sección de adquisiciones.

Esto podría lograrse expidiendo dos certificados de la siguiente manera; el primero tendría un **permittedSubtrees** de {base: C=US, O=Acme} y un **excludedSubtrees** de {base: C=US, O=Acme, OU=branch20}. El segundo tendría un **permittedSubtrees** de {base: C=US, O=Acme, OU=branch20, OU=Purchasing}.

El anexo G contiene ejemplos de aplicación de la extensión de constricciones de nombre.

8.4.2.3 Extensión de constricciones de políticas

Este campo especifica las constricciones 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, y el trayecto de certificación incluye un certificado expedido por una CA denominada, es necesario que todos los certificados contengan, 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 una de estas políticas mediante la correspondencia de políticas o *cualquier política*. La CA denominada es la CA que expide el certificado que contiene esta extensión (si el valor de **requireExplicitPolicy** es 0) o una CA que es el expedidor de un certificado subsiguiente en el trayecto de certificación (indicado por un valor distinto de 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 0) o una CA que es el sujeto de un certificado subsiguiente en el trayecto de certificación (indicado por un valor distinto de 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 constricción 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.

8.4.2.4 Extensión inhibición de cualquier política

Este campo especifica una constricción que indica que *cualquier política* no se considera una concordancia explícita para otras políticas de certificado para todos los certificados no autoexpedidos en el trayecto de certificación que se inicia con una CA denominada. La CA denominada es la CA que es el sujeto del certificado que contiene esta extensión (si el valor de **inhibitAnyPolicy** es 0) o una CA que es el sujeto de un certificado subsiguiente en el trayecto de certificación (indicado por un valor distinto de cero).

```
inhibitAnyPolicy        EXTENSION   ::= {
    SYNTAX              SkipCerts
IDENTIFIED BY           id-ce-inhibitAnyPolicy }
```

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.

8.5 Extensiones de la CRL básica

8.5.1 Requisitos

Los siguientes requisitos se relacionan con las CRL:

a) Los usuarios de certificados tendrán que ser capaces de seguir todas las CRL expedidas por un expedidor de CRL o punto de distribución CRL (véase 8.6) 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 autoridad 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 tendrán 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.

e) Los usuarios de certificados tendrá que ser capaces de determinar, a partir de la propia CRL, información adicional que incluya el ámbito de los certificados incluidos en la lista, el orden de las notificaciones de revocación y que tren de CRL es en el que el número de CRL es único.

f) Los expedidores tendrán que tener la capacidad de cambiar de forma dinámica la partición de las CRL e indicar a los usuarios de certificados las nuevas ubicaciones de las CRL importantes, si la partición se modifica.

g) Las CRL delta que actualicen una determinada CRL básica también podrían estar disponibles. Los usuarios de certificados tendrá que ser capaces de determinar, a partir de una determinada CRL, si están disponibles las CRL delta, donde están ubicadas y cuando se expedirá la siguiente CRL delta.

h) Además de que las CRL publiquen la notificación de que han sido revocados certificados, es necesario publicar la notificación de que los certificados serán revocados a partir de una fecha y hora específicas en el futuro.

i) Es necesario ofrecer medios más eficaces de indicar en una CRL que un conjunto de certificados ha sido revocado.

8.5.2 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;*

e) *ámbito de CRL;*

f) *referencia de estado;*

g) *identificador de tren de CRL;*

h) *lista ordenada;*

i) *información delta.*

El número de CRL, el ámbito de CRL, la referencia de estado, el identificador de tren de CRL, la lista ordenada y la información delta se utilizarán únicamente como campos de extensión de CRL y los otros campos se utilizarán únicamente como campos de extensión de asientos de CRL.

8.5.2.1 Extensión 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 autoridad 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.

8.5.2.2 Extensión 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),
    privilegeWithdrawn      (9),
    aaCompromise            (10) }
```

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

- **unspecified** se utiliza para revocar certificados por motivos distintos a los códigos específicos;
- **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;
- **privilegeWithdrawn** indica que un certificado (certificado de clave pública o de atributo) ha sido revocado debido a que se ha suprimido un privilegio incluido en dicho certificado;
- **aACompromise** indica que se sabe o se sospecha que se han comprometido aspectos de la AA validada en el certificado de atributo.

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 8.5.2.3). 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 8.6) 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 o alguna subsiguiente (delta o completa para el ámbito) contiene un asiento para el mismo certificado con el código de motivo **certificateHold**.

Esta extensión siempre es no crítica.

8.5.2.3 Extensión de código de instrucción de extensión

Este campo de extensión de asientos CRL proporciona 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, comunique con la CA" o "recupere el testigo del usuario".

8.5.2.4 Extensión 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 autoridad 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 autoridad 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.

8.5.2.5 Extensión de ámbito de CRL

NOTA – Se desaconseja la utilización de la extensión de alcance de la CRL.

El ámbito de una CRL se indica en la propia CRL utilizando la siguiente extensión de CRL. Para evitar que una sustitución de CRL esté en contradicción con una aplicación que no soporta la extensión de ámbito, dicha extensión, si está presente, tendrá que indicarse como crítica.

Esta extensión se puede utilizar para suministrar declaraciones de ámbito a diversos tipos de CRL incluidas:

- las CRL simples que proporcionan información de revocación sobre certificados expedidos por una única autoridad;
- las CRL indirectas que proporcionan información de revocación sobre certificados expedidos por múltiples autoridades;
- las CRL delta que actualizan la información de revocación previamente expedida;
- las CRL delta indirectas que proporcionan información de revocación que actualiza múltiples CRL básica expedidas por una única autoridad o por múltiples autoridades.

```
crlScope EXTENSION ::= {
    SYNTAX              CRLScopeSyntax
    IDENTIFIED BY       id-ce-cRLScope }

CRLScopeSyntax ::=      SEQUENCE SIZE (1..MAX) OF PerAuthorityScope

PerAuthorityScope ::= SEQUENCE {
    authorityName               [0]     GeneralName OPTIONAL,
    distributionPoint           [1]     DistributionPointName OPTIONAL,
    onlyContains                [2]     OnlyCertificateTypes OPTIONAL,
    onlySomeReasons             [4]     ReasonFlags OPTIONAL,
    serialNumberRange           [5]     NumberRange OPTIONAL,
    subjectKeyIdRange           [6]     NumberRange OPTIONAL,
    nameSubtrees                [7]     GeneralNames OPTIONAL,
    baseRevocationInfo          [9]     BaseRevocationInfo OPTIONAL
    }
```

```
OnlyCertificateTypes ::= BIT STRING {
    user            (0),
    authority       (1),
    attribute       (2) }

NumberRange ::= SEQUENCE {
    startingNumber      [0]     INTEGER OPTIONAL,
    endingNumber        [1]     INTEGER OPTIONAL,
    modulus                     INTEGER OPTIONAL }

BaseRevocationInfo ::= SEQUENCE {
    cRLStreamIdentifier     [0]     CRLStreamIdentifier     OPTIONAL,
    cRLNumber               [1]     CRLNumber,
    baseThisUpdate          [2]     GeneralizedTime }
```

Si la CRL es una CRL indirecta que proporciona información sobre el estado de revocación a múltiples autoridades, la extensión incluirá múltiples constructivos **PerAuthorityScope**, uno o más para cada una de las autoridades para las que se incluye información de revocación. Cada ejemplar de **PerAuthorityScope**, relativo a una autoridad distinta de la que expide esta CRL, incluirá el componente **authorityName**. Si la CRL es una dCRL que proporciona información de estado de revocación delta para múltiples CRL básicas expedidas por una única autoridad, la extensión incluirá múltiples constructivos **PerAuthorityScope**, uno para cada una de las CRL básica para las que esta dCRL suministra actualizaciones. Aunque deberían existir múltiples ejemplares del constructivo **PerAuthorityScope**, el valor del componente **authorityName**, si esta presente, será el mismo para todos los ejemplares.

Si la CRL es una dCRL indirecta que proporciona información de estado de revocación delta para múltiples CRL básica expedidas por múltiples autoridades, la extensión incluirá múltiples constructivos **PerAuthorityScope**, uno para cada una de las CRL básicas para las que esta dCRL suministra actualizaciones. Cada ejemplar de **PerAuthorityScope** relativo a una autoridad distinta de la que expide esta dCRL indirecta incluirá el componente **authorityName**.

Para cada ejemplar **PerAuthorityScope** presente en la extensión, los campos se utilizan como sigue. Cabe destacar que en el caso de CRL indirectas y dCRL indirectas cada ejemplar de **PerAuthorityScope** puede contener diferentes combinaciones de estos campos y diferentes valores.

El campo **authorityName**, si está presente, identifica a la autoridad que expidió los certificados para los que se proporciona la información de revocación. Si se omite **authorityName**, figurará por defecto el nombre del expedidor de la CRL.

El campo **distributionPoint**, si está presente, se usa como se describe en la extensión **issuingDistributionPoint**.

El campo **onlyContains**, si está presente, indica que tipo o tipos de certificados contiene la CRL relativos a información de estado de revocación. Si este campo está ausente, la CRL contiene información sobre todos los tipos de certificado.

El campo **onlySomeReasons**, si está presente se utiliza como se describe en la extensión **issuingDistributionPoint**.

El elemento **serialNumberRange**, si está presente, se utiliza como sigue. Cuando un valor de módulo está presente, el número de serie se reduce al valor de módulo dado antes de comprobar su presencia en la gama. Entonces, se considera que un certificado con un número de serie (reducido) está dentro del ámbito de la CRL si es:

- igual o superior a **startingNumber**, e inferior a **endingNumber**, cuando están presentes ambos; o
- igual o superior a **startingNumber**, cuando **endingNumber** no está presente; o
- inferior a **endingNumber** cuando **startingNumber** no está presente.

Si está presente el elemento **subjectKeyIdRange**, se interpreta de la misma forma que **serialNumberRange**, salvo que el número utilizado es el valor de la extensión de certificado **subjectKeyIdentifier**. La codificación de DER (reglas de codificación distinguida) del **BIT STRING** (omitiendo la etiqueta, la longitud y el octeto de bits no utilizados) debe considerarse como el valor de la codificación DER de un **INTEGER**. Si el bit0 del **BIT STRING** se pone en ese valor, debería preadjuntarse un octeto 0 adicional para garantizar que la codificación resultante represente un **INTEGER** positivo por ejemplo:

03 02 01 f7 (representa el conjunto de bits 0-6)

corresponde a

02 02 00 f7 (esto es, decimal 247)

El campo **nameSubtrees**, si está presente, utiliza los mismos convenios que para las formas de nombre especificadas en la extensión **nameConstraints**.

El campo **baseRevocationInfo**, si está presente, indica que la CRL es una dCRL en lo que respecta a los certificados cubiertos por dicho constructivo **PerAuthorityScope**. La utilización de la extensión **crlScope** para identificar que una CRL es una dCRL difiere de la utilización de la extensión **deltaCRLIdentifier** de la manera siguiente. En el caso de **crlScope**, la información en el componente **baseRevocationInfo** indica el punto en el tiempo a partir del cual la CRL que incluye esta extensión proporciona actualizaciones. Aunque esto se realiza haciendo referencia a una CRL, la CRL referenciada puede ser o no una que está completa para el ámbito considerado, donde la referencia de extensión **deltaCRLIdentifier** expidió una CRL que está completa para el ámbito considerado. Mientras que la información actualizada proporcionada en una dCRL que contiene la extensión **crlScope** es de actualizaciones de la información de revocación que está completa para un ámbito considerado, independientemente de si la CRL referenciada en **baseRevocationInfo** se expidió o no realmente como una que estuviera completa para este mismo ámbito. Este mecanismo proporciona más flexibilidad que la extensión **deltaCRLIndicator** puesto que los usuarios pueden construir las CRL completas localmente y hacerlo basándose en el tiempo en lugar de en la expedición de CRL básicas completas para el ámbito considerado. En ambos casos, una dCRL siempre proporciona actualizaciones de estados de revocación para certificados en un ámbito determinado a partir de un determinado instante en el tiempo. Sin embargo, en el caso de **deltaCRLIndicator**, dicho instante en el tiempo tendrá que ser uno para el cual una CRL que está completa para dicho ámbito esté expedida y referenciada. En el caso **crlScope**, dicho instante en el tiempo puede ser uno para el cual la CRL referenciada que se expidió puede o no estar completa para dicho ámbito.

En función de la política de la autoridad responsable, se pueden publicar varias dCRL antes de que se publique una CRL básica nueva. Las dCRL que contienen la extensión **crlScope** para referenciar su punto de construcción no tienen necesariamente que hacer referencia a **cRLNumber** de la última CRL básica expedida en el campo **BaseRevocationInfo**. Sin embargo, el **cRLNumber** referenciado en el campo **BaseRevocationInfo** de una dCRL será inferior o igual al **cRLNumber** de la última CRL expedida que esté completa para el ámbito considerado.

Cabe destacar que la extensión **issuingDistributionPoint** y la extensión **crlScope** pueden chocar y no se pretende que se utilicen juntas. Sin embargo, si la CRL contiene tanto una extensión **issuingDistributionPoint** como una extensión **crlScope**, entonces un certificado de clave pública se considera dentro del ámbito de la CRL si y sólo si cumple los criterios de ambas extensiones. Si la CRL contiene una extensión **AAissuingDistributionPoint**, pero no contiene ninguna de las extensiones **issuingDistributionPoint** o **crlScope**, en ese caso el ámbito no incluye certificados de clave pública. Si la CRL no contiene una extensión **issuingDistributionPoint**, **AAissuingDistributionPoint** o **crlScope**, el ámbito representa el ámbito completo de la autoridad, y podrá utilizarse la CRL para cualquier certificado de esa autoridad. De modo similar, las extensiones **iAAssuingDistributionPoint** y **crlScope** pueden entrar en conflicto entre sí y no estar destinadas a utilizarse juntas. No obstante, si la CRL contiene ambas extensiones **AAissuingDistributionPoint** y **crlScope**, en ese caso un certificado de atributo cae dentro del alcance de la CRL únicamente si cumple con los criterios de ambas extensiones. Si la CRL contiene una extensión **issuingDistributionPoint**, pero no contiene ninguna de las extensiones **AAissuingDistributionPoint** o **crlScope**, el ámbito no incluye certificados de atributo. Si la CRL no contiene una extensión **issuingDistributionPoint**, **AAissuingDistributionPoint**, o **crlScope** entonces el ámbito es el ámbito completo de autoridad y la CRL puede ser utilizada por cualquier certificado proveniente de esa autoridad.

Cuando un sistema que utiliza certificados usa una CRL que contiene una extensión **crlScope** para comprobar el estado de un certificado, comprobará que el certificado y los códigos de motivo de interés se encuentran dentro del ámbito de la CRL, según se define en la extensión **crlScope** de la forma siguiente:

a) El sistema que utiliza certificados tendrá que comprobar que el certificado se encuentra dentro del ámbito indicado mediante la intersección de los ámbitos del **serialNumberRange**, **subjectKeyIdRange** y **nameSubtrees** y que es coherente con **distributionPoint** y **onlyContains**, si están presentes, para el constructivo **PerAuthorityScope** pertinente.

b) Si la CRL contiene un componente **onlySomeReasons** en la extensión **crlScope**, el sistema que utiliza certificados tendrá que comprobar que los códigos de motivo incluidos en esta CRL son adecuados para los fines de esta aplicación. En caso contrario, pueden ser necesarias CRL adicionales. Hay que destacar que si la CRL contiene tanto una extensión **crlScope** como una extensión **issuingDistributionPoint** y ambas contienen un componente **onlySomeReasons**, entonces sólo aquellos códigos de motivo incluidos en los componentes **onlySomeReasons** de ambas extensiones están cubiertos por esta CRL.

8.5.2.6 Extensión de referencia de estado

Esta extensión de CRL es para su utilización en la estructura de CRL como un medio de encaminar información sobre notificaciones de revocación a usuarios de certificado. De esta forma, estaría presente en una estructura de CRL que no contiene por sí misma ninguna notificación de revocación de certificado. Una estructura de CRL que incluya esta extensión no será utilizada por el usuario de certificado o por partes confiantes para obtener notificaciones de revocación, sino más bien como una herramienta para asegurar que se utiliza la información de revocación adecuada. Cualquier CRL que contenga esta extensión no debe ser utilizada como la fuente para que una parte confiante compruebe el estado de revocación de cualquier certificado. Más bien, si una CRL contiene esta extensión, podrá ser

utilizada por una parte confiante como un mecanismo adicional para localizar las CRL pertinentes para comprobar el estado de revocación.

Esta extensión realiza dos funciones básicas:

- Esta extensión proporciona mecanismos para publicar una "lista de CRL" fiduciaria que incluye toda la información pertinente para ayudar a partes confiantes a determinar si tienen o no información de revocación suficiente para sus necesidades. Por ejemplo, una autoridad puede expedir periódicamente una lista de CRL nueva y autenticada, normalmente con una frecuencia de reedición relativamente alta (en comparación con frecuencias de reedición de otras CRL). La lista podría incluir una hora/fecha de la última actualización para cada CRL referenciada. Un usuario de certificado, al obtener esta lista, puede determinar rápidamente si sus copias de las CRL siguen actualizadas. Esto puede eliminar muchas recuperaciones de CRL innecesarias. Además, utilizando este mecanismo, los usuarios de certificado conocerán las CRL expedidas por una autoridad dentro de su ciclo de actualización normal, mejorando así la rapidez del sistema de CRL.
- Esta extensión también proporciona un mecanismo para reencaminar a una parte confiante desde una ubicación preliminar (por ejemplo indicada en una extensión de punto de distribución de CRL, o en un asiento de directorio de una autoridad expedidora) a una ubicación diferente para información de revocación. Esto permite a las autoridades modificar el esquema de partición de las CRL que utilizan sin influir en certificados o usuarios de certificado existentes. Para lograr esto, la autoridad incluiría cada nueva ubicación y el ámbito de la CRL que se encontrará en dicha ubicación. La parte confiante compararía el certificado pertinente con las declaraciones de ámbito y seguirá al puntero hacia la nueva ubicación para la información de revocación pertinente al certificado que están validando.

La extensión es extensible en sí misma y, en el futuro, también se podrá hacer referencia a otros esquemas de revocación no basados en CRL, utilizando esta extensión.

```
statusReferrals EXTENSION ::= {
    SYNTAX              StatusReferrals
    IDENTIFIED BY       id-ce-statusReferrals }

StatusReferrals ::= SEQUENCE SIZE (1..MAX) OF StatusReferral

StatusReferral ::= CHOICE {
    cRLReferral         [0]     CRLReferral,
    otherReferral       [1]     INSTANCE OF OTHER-REFERRAL}


CRLReferral ::= SEQUENCE {
    issuer              [0]     GeneralName OPTIONAL,
    location            [1]     GeneralName OPTIONAL,
    deltaRefInfo        [2]     DeltaRefInfo OPTIONAL,
    cRLScope                    CRLScopeSyntax,
    lastUpdate          [3]     GeneralizedTime OPTIONAL,
    lastChangedCRL      [4]     GeneralizedTime OPTIONAL}

DeltaRefInfo ::= SEQUENCE {
    deltaLocation       GeneralName,
    lastDelta           GeneralizedTime OPTIONAL }

OTHER-REFERRAL ::= TYPE-IDENTIFIER
```

El campo **issuer** identifica la identidad que firma la CRL; por defecto, indicará el nombre del expedidor de la CRL acompañante.

El campo **location** proporciona las ubicaciones a las que se debe dirigir la referencia y, por defecto, indica el mismo valor que el nombre de **issuer**.

El campo **deltaRefInfo** proporciona una ubicación alternativa facultativa a partir de la cual se puede obtener una dCRL y una fecha facultativa de la CRL delta anterior.

El campo **cRLScope** proporciona el ámbito de la CRL que se encontrará en la ubicación referenciada.

El campo **lastUpdate** es el valor del campo **thisUpdate** en la CRL referenciada, expedida más reciente.

lastChangedCRL es el valor del campo **thisUpdate** en la CRL expedida más reciente que ha modificado su contenido.

OTHER-REFERRAL proporciona extensibilidad para permitir que se acomoden en el futuro otros esquemas de revocación no basados en CRL.

Esta extensión siempre se indica como crítica mediante banderas para asegurar que la CRL que contiene esta extensión no vincula inadvertidamente a sistemas que utilizan los certificados como fuente de información del estado de revocación sobre certificados.

Si está presente esta extensión y el sistema que utiliza certificados la reconoce, dicho sistema no utilizará la CRL como fuente de información del estado de revocación. El sistema utilizará la información incluida en esta extensión y otros medios fuera del ámbito de esta Especificación para localizar la información adecuada del estado de revocación.

Si está presente esta extensión pero el sistema que utiliza certificados no la reconoce, dicho sistema no utilizará la CRL como fuente de información del estado de revocación. El sistema utilizará otros medios fuera del ámbito de esta Especificación para localizar la información de revocación adecuada.

8.5.2.7 Extensión de identificador de flujo de CRL

El campo identificador de flujo de CRL se utiliza para identificar el contexto en el que el número de CRL es único.

```
cRLStreamIdentifier EXTENSION      ::= {
    SYNTAX              CRLStreamIdentifier
    IDENTIFIED BY       id-ce-cRLStreamIdentifier }

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

Esta extensión siempre es no crítica.

Cada valor de esta extensión, tendrá que ser único para cada autoridad. El identificador de flujo de CRL combinado con un número de CRL sirve como identificador único para cada CRL expedida por una determinada autoridad, independientemente del tipo de CRL.

8.5.2.8 Extensión de lista ordenada

La extensión de lista ordenada indica que la secuencia de certificados revocados en el campo **revokedCertificates** de una CRL está en orden ascendente, ya sea mediante el número de serie de certificado o mediante la fecha de revocación. Este campo se define como sigue:

```
orderedList EXTENSION      ::= {
    SYNTAX              OrderedListSyntax
    IDENTIFIED BY       id-ce-orderedList }

OrderedListSyntax ::= ENUMERATED {
ascSerialNum        (0),
ascRevDate          (1) }
```

Esta extensión siempre es no crítica.

- **ascSerialNum** indica que la secuencia de certificados revocados en una CRL es en orden ascendente del número de serie de certificado, basado en el valor del componente **serialNumber** de cada asiento en la lista.
- **ascRevDate** indica que la secuencia de certificados revocados en una CRL es en orden ascendente de la fecha de revocación, basada en el valor del componente **revocationDate** de cada asiento en la lista.

Si no está presente **orderedList**, no se proporciona información sobre el orden, si existe, de la lista de certificados revocados en la CRL.

8.5.2.9 Extensión de información delta

Esta extensión de CRL se utiliza en las CRL que no son dCRL y se utiliza para indicar a las partes confiantes que también se dispone de dCRL para la CRL que contiene esta extensión. La extensión proporciona la ubicación en la que se puede encontrar la dCRL relacionada y facultativamente el instante en el que se expedirá la próxima dCRL.

```
deltaInfo EXTENSION        ::=   {
    SYNTAX              DeltaInformation
    IDENTIFIED BY       id-ce-deltaInfo }

DeltaInformation        ::=         SEQUENCE {
    deltaLocation       GeneralName,
    nextDelta           GeneralizedTime OPTIONAL }
```

Esta extensión siempre es no crítica.

8.5.2.10 Extensión por revocar (To be revoked)

Esta extensión de CRL permite notificar que algunos certificados serán revocados a partir de una fecha y hora específicas en el futuro. La extensión **toBeRevoked** es útil para especificar el motivo de la revocación de un certificado,

la fecha y hora en que será revocado y el grupo de certificados que habrán de ser revocados. Cada lista puede contener un solo número de serie de certificado, una gama de números de serie de certificados o un **subtree** denominado. Los certificados pueden ser de clave pública o de atributo.

```
toBeRevoked  EXTENSION  ::= {
  SYNTAX              ToBeRevokedSyntax
  IDENTIFIED BY       id-ce-toBeRevoked }

ToBeRevokedSyntax::=      SEQUENCE SIZE(1..MAX) OF ToBeRevokedGroup

ToBeRevokedGroup ::=      SEQUENCE {
  certificateIssuer   [0] GeneralName OPTIONAL,
  reasonInfo          [1] ReasonInfo     OPTIONAL,
  revocationTime          GeneralizedTime,
  certificateGroup        CertificateGroup }

ReasonInfo                ::=     SEQUENCE {
  reasonCode              CRLReason,
  holdInstructionCode     HoldInstruction OPTIONAL }

CertificateGroup     ::=  CHOICE {
  serialNumbers           [0]     CertificateSerialNumbers,
  serialNumberRange       [1]     CertificateGroupNumberRange,
  nameSubtree             [2]     GeneralName }

CertificateGroupNumberRange ::= SEQUENCE {
  startingNumber    [0]     INTEGER,
  endingNumber      [1]     INTEGER }

CertificateSerialNumbers ::= SEQUENCE SIZE(1..MAX) OF CertificateSerialNumber
```

El campo **certificateIssuer**, si está presente, identifica la autoridad (CA o AA) que expidió todos los certificados enumerados en este **ToBeRevokedGroup**. Si se omite el campo **certificateIssuer** se utilizará por defecto el nombre del expedidor de la CRL.

El campo **reasonInfo**, si está presente, identifica el motivo de las revocaciones de los certificados. Si está presente, este campo indica que todos los certificados identificados en **ToBeRevokedGroup** serán revocados por el motivo indicado en este campo. Si **reasonCode** contiene el valor **certificateHold**, es probable que **holdInstructionCode** también esté presente. Si está presente, **holdInstructionCode** indica la medida que habrá de adoptarse al encontrar cualquiera de los certificados identificados en **RevokedGroup**. Esta medida sólo será adoptada, tras haber transcurrido la hora de revocación indicada en el campo **revocationTime**.

El campo **revocationTime** indica la fecha y hora en que será revocado este grupo de certificados y a partir de que momento se considerará no válido. La fecha debe ser posterior a la hora **thisUpdate** de la CRL que contiene esta extensión. Si **revocationTime** es anterior a la hora **nextUpdate** de la CRL que contiene esta extensión, los certificados se considerarán revocados entre la hora **revocationTime** y la hora **nextUpdate** por una parte confiante que utilice una CRL que contenga esta extensión. De lo contrario, se trata de una notificación de que estos certificados serán revocados a una hora específica en el futuro. Una vez transcurrido el tiempo de revocación la CA habrá revocado o no el certificado. Si ha revocado el certificado, las CRL futuras lo incluirán en la lista de certificados revocados, al menos hasta que expire el certificado. Si la CA no ha revocado el certificado, pero pretende revocarlo en el futuro, podrá incluir el certificado en esta extensión en CRL subsiguientes con un **revocationTime** revisado. Si la CA ya no pretende revocar el certificado, podrá excluirlo de todas las CRL subsiguientes y el certificado no se considerará revocado.

El campo **certificateGroup** enumera el conjunto de certificados que han de ser revocados. Este campo identifica los certificados expedidos por la autoridad identificada en **certificateIssuer** que van a ser revocados en la fecha/hora identificadas en **revocationTime**. Este conjunto de certificados no sigue siendo refinado por ningún control exterior (por ejemplo, **issuingDistributionPoint**).

Si está presente **serialNumbers**, el certificado o los certificados con números de serie indicados en este campo, y expedidos por el expedidor de certificados identificado, serán revocados en el momento especificado.

Si está presente **serialNumberRange**, todos los certificados de la gama que comienzan con el número de serie inicial y terminan con el número de serie final, y expedidos por el expedidor de certificados identificado, serán revocados en el momento especificado.

Si está presente **nameSubtree**, todos los certificados con un nombre de sujeto/titular subordinado al nombre especificado y expedidos por el expedidor de certificados identificado, serán revocados a la hora especificada. Si **nameSubtree** contiene un DN, deberán considerarse todos los DN asociados con el sujeto de un certificado de clave

pública (es decir, el campo **subject** y la extensión **subjectAltNames**) o el campo **holder** de un certificado de atributo. Para otras formas de nombre deberá considerarse la extensión **subjectAltNames** de los certificados de clave pública y el campo **holder** de los certificados de atributo. Si al menos uno de los nombres asociados con el sujeto/titular, contenido en el certificado, se encuentra dentro del subárbol especificado en **nameSubtree**, ese certificado será revocado en el momento especificado. Como en el caso de la extensión **nameConstraints**, no todas las formas de nombre son adecuadas para la especificación de **subtree**. En esta extensión deberían utilizarse sólo aquellas que tienen reglas de subordinación reconocidas.

A opción del expedidor de CRL, esta extensión puede señalarse como crítica o no crítica mediante banderas. Dado que la información proporcionada en esta extensión se aplica a revocaciones que se realizarán en el futuro, se recomienda que se señale como no crítica mediante una bandera, reduciendo el riesgo de problemas de interoperabilidad y de compatibilidad hacia atrás.

8.5.2.11 Extensión de grupo de certificados revocados

Un conjunto de certificados que ha sido revocado puede publicarse utilizando la extensión CRL siguiente. Cada lista de certificados a revocar se asocia con un expedidor de certificados y una hora de revocación específicos, y puede contener una gama de números de serie de certificados o un subárbol denominado. Estos certificados pueden ser de clave pública o de atributo.

```
revokedGroups  EXTENSION  ::= {
  SYNTAX              RevokedGroupsSyntax
  IDENTIFIED BY       id-ce-RevokedGroups }

RevokedGroupsSyntax ::=    SEQUENCE SIZE (1..MAX) OF RevokedGroup

RevokedGroup ::=    SEQUENCE {
  certificateIssuer             [0]     GeneralName OPTIONAL,
  reasonInfo                    [1]     ReasonInfo OPTIONAL,
  invalidityDate                [2]     GeneralizedTime OPTIONAL,
  revokedcertificateGroup       [3]     RevokedCertificateGroup }

RevokedCertificateGroup ::= CHOICE {
  serialNumberRange         NumberRange,
  nameSubtree               GeneralName }
```

El campo **certificateIssuer**, si está presente, identifica la autoridad (CA o AA) que expidió todos los certificados enumerados en este **RevokedGroup**. Si se omite **certificateIssuer**, se aplica por defecto el nombre del expedidor de la CRL.

El campo **reasonInfo**, si está presente, identifica el motivo de las revocaciones de los certificados. Si está presente, este campo indica que todos los certificados identificados en **RevokedGroup** fueron revocados por el motivo indicado en este campo. Si **reasonCode** contiene el valor **certificateHold**, es probable que **holdInstructionCode** también esté presente. Si está presente, **holdInstructionCode** indica la medida que habrá de adoptarse al encontrar cualquiera de los certificados identificados en **RevokedGroup**.

El campo **invalidityDate**, si está presente, indica la hora a partir de la cual se considerarán no válidos todos los certificados identificados en **RevokedGroup**. Esta fecha debe ser anterior a la fecha incluida en el campo **thisUpdate** de la CRL. Si se omite, todos los certificados identificados en **RevokedGroup** deben considerarse no válidos al menos a partir de la hora indicada en el campo **thisUpdate** de la CRL. Si el estado del certificado anterior a la hora **thisUpdate** es crítico para un sistema que utiliza certificado (por ejemplo, para determinar si una firma digital creada antes de producirse la expedición de esta CRL tuvo lugar mientras el certificado aún era válido o después de haber sido revocado), se necesitarán técnicas adicionales de comprobación del estado de revocación para determinar la fecha y hora reales a partir de las cuales debe considerarse no válido un certificado determinado.

El campo **revokedCertificateGroup** enumera el conjunto de certificados que han sido revocados. Este campo identifica los certificados expedidos por la autoridad identificada en el **certificateIssuer** revocados en las condiciones especificadas. Este conjunto de certificados no sigue siendo refinado por ningún control exterior (por ejemplo, **issuingDistributionPoint**).

Si está presente **serialNumberRange**, son aplicables todos los certificados que contienen números de serie de certificados dentro de la gama especificada, expedidos por el expedidor de certificados identificado.

Si está presente **nameSubtree**, todos los certificados con un nombre de sujeto/titular subordinado al nombre especificado y expedidos por el expedidor de certificados identificado serán revocados a la hora especificada. Si el **nameSubtree** contiene un DN, deberán considerarse todos los DN asociados con el sujeto de un certificado de clave pública (es decir, el campo **subject** y la extensión **subjectAltNames**) o el campo **holder** de un certificado de atributo. Para otras formas de nombre deberán considerarse la extensión **subjectAltNames** de los certificados de clave pública y

el campo **holder** de los certificados de atributo. Si al menos uno de los nombres asociados con el sujeto/titular, contenido en el certificado, se encuentra dentro del subárbol especificado en **nameSubtree**, ese certificado ha sido revocado. Como en el caso de la extensión **nameConstraints**, no todas las formas de nombre son adecuadas para la especificación de subtree. En esta extensión deberían utilizarse sólo aquellas que tienen reglas de subordinación reconocidas.

Esta extensión se señala siempre como crítica, mediante banderas. De lo contrario, un sistema que utilice certificado podrá suponer incorrectamente que los certificados identificados como revocados en esta extensión, no están revocados. Cuando esta extensión esté presente, puede ser la única indicación de los certificados revocados en una CRL (es decir, **revokedCertificates** puede estar vacío) o puede enumerar los certificados revocados adicionales a los indicados en el campo **revokedCertificates**. Un certificado revocado no debe enumerarse tanto en el campo **revokedCertificates** como en esta extensión.

8.5.2.12 Extensión de certificados expirados en la CRL

Este campo de extensión CRL indica que la CRL incluye los avisos de revocación de los certificados expirados.

```
expiredCertsOnCRL EXTENSION  ::= {
  SYNTAX      ExpiredCertsOnCRL
  IDENTIFIED BY      id-ce-expiredCertsOnCRL }

ExpiredCertsOnCRL  ::= GeneralizedTime
```

Esta extensión es siempre no crítica.

El ámbito de una CRL que contiene esta extensión se amplía para poder incluir el estado de revocación de los certificados que expiraron en el momento exacto especificado en la extensión o después del mismo. Si se especifican limitaciones en el ámbito de la CRL (mediante códigos de motivo o puntos de distribución), se aplica lo mismo también a los certificados expirados. El estado de revocación de un certificado no debe actualizarse una vez que el certificado ha expirado.

8.6 Extensiones de puntos de distribución de CRL y CRL delta

8.6.1 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 autoridades. Como ya tiene una lista de certificados previamente revocados, sólo tendrá 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 dCRL:

a) Para controlar los tamaños de CRL es necesario poder asignar subconjuntos del conjunto de todos los certificados emitidos por una autoridad 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 CRL 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 tendrán 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 autoridades.

c) Se necesitan distintas CRL que contengan certificados de autoridades revocados y certificados de entidad final revocados. Esto facilita el procesamiento del trayecto de certificación porque cabe esperar que las CRL para certificados de autoridades 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 dCRL) que sólo contienen asientos para certificados que han sido revocados desde la expedición de una CRL básica.

f) Para las CRL delta, hay que prever que se indique la fecha/hora a partir de la cual esta lista incluye actualizaciones.

g) Hay un requisito para indicar en un certificado donde se puede encontrar la CRL más reciente (por ejemplo la delta más reciente).

8.6.2 Campos de extensión de punto de distribución de CRL y de CRL delta

Se definen los siguientes campos de extensión:

a) *puntos de distribución de CRL;*

b) *punto de distribución de expedidor;*

c) *AAissuingDistributionPoint;*

d) *expedidor de certificado;*

e) *indicador de CRL delta;*

f) *actualización básica;*

g) *CRL más reciente.*

Los puntos de distribución de CRL y CRL más reciente se utilizarán únicamente como una extensión de certificado. El punto de distribución expedidor, el punto de distribución expedidor de AA, el indicador de CRL delta y la utilización básica se utilizarán únicamente como extensiones de CRL. El expedidor de certificados se utilizará únicamente como una extensión de asiento de CRL.

Aunque la extensión de punto de distribución expedidor y la extensión de punto de distribución expedidor de AA se emplean para fines similares, se aplican a diferentes certificados. La extensión de punto de distribución expedidor sólo se aplica a los certificados de clave pública expedidos a usuarios y/o CA. La extensión de punto de distribución expedidor de AA sólo se aplica a los certificados de atributo expedidos a los usuarios y a los AA así como a los certificados de clave pública expedidos a las SOA. Si una sola CRL cubre tipos de certificados que abarcan a todos ellos, la CRL necesitaría incluir ambas extensiones.

8.6.2.1 Extensión de puntos de distribución de CRL

La extensión de puntos de distribución de CRL se utilizará únicamente como una extensión de certificado y puede ser utilizada en certificados de autoridad, certificados de clave pública de entidad final y en certificados de atributo. Este campo identifica el punto o puntos de distribución de CRL a los cuales el usuario de 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 autoridad.

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),
    privilegeWithdrawn      (7),
    aACompromise            (8) }
```

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 tendrá 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 autoridad.

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 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 los códigos de motivos de interés. Cuando los puntos de distribución se utilizan para distribuir información de CRL para todos los códigos de motivo de revocación y todos los certificados expedidos por la CA incluyen **cRLDistributionPoints** como una extensión crítica, no se requiere a la CA que también publique una CRL completa en el asiento 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 autoridad (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 1 – 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 autoridad expedidora es un aspecto de la política de la autoridad.

NOTA 2 – El significado de cada código de motivo es el que se define en el campo de código de motivos de 8.5.2.2 de esta Especificación.

8.6.2.2 Extensión de punto de distribución expedidor

Este campo de extensión de CRL identifica el punto de distribución de CRL para los certificados de clave pública de esta CRL particular, e indica si ésta es indirecta, o está limitada únicamente a un subconjunto de la información de revocación. Si se utilizan sólo CRL separadas, el conjunto completo de éstas cubrirá todo el conjunto de certificados cuyo estado de revocación será comunicado a través del mecanismo CRL. Por consiguiente, el conjunto completo de CRL separadas será equivalente a una CRL completa para el mismo conjunto de certificados, si el expedidor de CRL no estuviera usando CRL separadas. La limitación puede basarse en un subconjunto del total de certificados o en un subconjunto de motivos de revocación. La CRL está firmada por la clave privada del expedidor de la CRL – los puntos de distribución de CRL no tienen su propio par de claves. No obstante, para una CRL distribuida por el directorio,

la CRL está almacenada en el asiento del punto de distribución de CRL, que no puede ser el asiento de directorio del expedidor de ésta. Si el campo punto de distribución expedidor, el campo punto de distribución expedidor AA, y el campo ámbito de la CRL están todos ausentes, la CRL contendrá asientos a todos los certificados de clave pública válidos no revocados expedidos por el emisor de la CRL. Cuando no haya ni campo de punto de distribución de emisor ni de ámbito de CRL, pero sí el de punto de distribución de emisor AA, el ámbito de la CRL no cubrirá certificados de clave pública.

Cuando aparece un certificado en una CRL, es posible borrarlo de una CRL posterior tras su expiración. Este campo se define así:

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

IssuingDistPointSyntax ::= SEQUENCE {
    -- If onlyContainsUserPublicKeyCerts and onlyContainsCACerts are both FALSE,
    -- the CRL covers both certificate types
    distributionPoint                [0] DistributionPointName OPTIONAL,
    onlyContainsUserPublicKeyCerts   [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 varias formas de nombres. Si **onlyContainsUserPublicKeyCerts** es verdadero, la CRL contiene revocaciones para los certificados de clave pública de entidad extremo. Si **onlyContainsCACerts** es verdadero, la CRL contiene revocaciones para certificados de CA. Si tanto **onlyContainsUserPublicKeyCerts** como **onlyContainsCACerts** son falsos, la CRL contiene revocaciones para certificados de clave pública de entidad extrema y certificados CA. Si **onlySomeReasons** está presente, la CRL únicamente contiene revocaciones de certificado de clave pública para el motivo o motivos identificados, de lo contrario tendrá revocaciones para todos los motivos. Si **indirectCRL** es verdadero, es posible que la CRL contenga notificaciones de revocación para certificados de clave pública de otras autoridades diferentes del expedidor de la CRL. La autoridad que se encarga de cada asiento viene indicada por la extensión del asiento CRL del expedidor de certificado en dicha entrada o es conforme a las reglas por defecto que se describen en 8.6.2.3. En una tal CRL, es responsabilidad de su expedidor garantizar que está completa en el sentido de que contiene todos los asientos de revocación, coherente con los indicadores **onlyContainsUserPublicKeyCerts**, **onlyContainsCACerts**, y **onlySomeReasons**, para todas las autoridades que identifican este expedidor de CRL en su certificado de clave pública.

Si las CRL están separadas por código de motivo, y éste cambia para un certificado revocado (provocando que el certificado pase de un tren CRL a otro), será necesario seguir incluyendo el certificado en el tren CRL del motivo de revocación antiguo hasta que se hayan alcanzado los instantes **nextUpdate** de todas las CRL, que no enumera el certificado, en el tren CRL del nuevo código de motivo.

Si la CRL contiene una extensión **issuingDistributionPoint** con el campo **distributionPoint** presente, al menos un nombre del punto de distribución en el certificado (por ejemplo, **cRLDistributionPoints**, **freshestCRL**, **issuer**) debe concordar con un nombre del punto de distribución en la CRL. Asimismo, puede darse el caso de que solo esté presente el campo **nameRelativeToCRLIssuer**. En ese supuesto, se realizaría una comparación de nombres en todo el DN, que se crearía agregando el valor del **nameRelativeToCRLIssuer** al DN encontrado en el campo **issuer** de la CRL. Si los nombres que se están comparando son DN (en oposición a los nombres de otras formas dentro de la construcción **GeneralNames**), se utiliza la regla de concordancia **distinguishedNameMatch** para determinar la igualdad de los dos DN.

En el caso de las CRL distribuidas a través del directorio se aplican las reglas siguientes. Si la CRL es una dCRL se la distribuirá a través del atributo **deltaRevocationList** del punto de distribución asociado o, si no se ha identificado un punto de distribución, a través del atributo **deltaRevocationList** de la entrada de expedidor CRL, sin importar la configuración de los tipos de certificado cubiertos por la CRL. A menos que la CRL sea una dCRL:

- Se distribuirá una CRL que tenga fijada **onlyContainsCACerts** y no contenga una extensión **AAissuingDistributionPoint**, mediante el atributo **authorityRevocationList** del punto de distribución asociado o, si no se define punto de distribución, mediante el atributo **authorityRevocationList** de la entrada de expedidor CRL.
- Se distribuirá una CRL que tenga fijado **onlyContainsCACerts** y que contenga una extensión **AAissuingDistributionPoint** con el **containsUserAttributeCerts** fijado a falso, a través del atributo **authorityRevocationList** del punto de distribución asociado o, de no haber un punto de distribución identificado, mediante el atributo **authorityRevocationList** de la entrada de expedidor de CRL.

- Se distribuirá una CRL que tenga solamente **onlyContainsCACerts** puesto a falso a través del atributo **certificateRevocationList** del punto de distribución asociado o, si no se ha identificado el punto de distribución, mediante el atributo **certificateRevocationList** de la entrada de expedidor CRL.
- Se distribuirá una CRL que contenga tanto una extensión **issuingDistributionPoint** como una **AAissuingDistributionPoint** con **containsUserAttributeCerts** fijado, mediante el atributo **certificateRevocationList** del punto de distribución asociado o, si no se ha identificado un punto de distribución, a través del atributo **certificateRevocationList** de la entrada de expedidor de CRL.

Esta extensión siempre es crítica. Un usuario de certificado que no entienda dicha extensión no puede suponer que la CRL contiene una lista completa de certificados revocados o la autoridad identificada. Las CRL que no contengan extensiones críticas no podrán contener todas las inserciones actuales de CRL correspondientes a la autoridad que expide, incluidas aquéllas para los certificados de usuarios revocados y certificados de autoridad.

NOTA 1 – Los medios que utilizan las autoridades para comunicar la información de revocación a los expedidores de CRL están fuera del alcance de esta Especificación de directorio.

NOTA 2 – Cuando una autoridad publique una CRL con **onlyContainsUserPublicKeyCerts** u **onlyContainsCACerts** puestos a verdadero, garantizará que todos los certificados de CA cubiertos por dicha CRL contengan la extensión **basicConstraints**.

8.6.2.3 Extensión 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.

8.6.2.4 Extensión de indicador de CRL delta

El campo indicador de CRL delta identifica que una CRL es una CRL delta (dCRL) que proporciona actualizaciones a una CRL básica referenciada. La CRL básica referenciada es una CRL que se expidió explícitamente como una CRL completa para un determinado ámbito. La CRL que contiene la extensión de indicador de CRL delta contiene actualizaciones del estado de revocación de certificado para este mismo ámbito. Este ámbito no incluye necesariamente todos los motivos de revocación o todos los certificados expedidos por una CA, en particular en el caso en el que la CRL es un punto de distribución de CRL. Sin embargo, la combinación de una CRL que contiene la extensión de indicador de CRL delta además de la CRL referenciada en el componente **BaseCRLNumber** de esta extensión es equivalente a una CRL completa para el ámbito correspondiente, en el instante de la publicación de la dCRL.

Este campo se define como sigue:

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

BaseCRLNumber ::= CRLNumber
```

El valor de tipo **BaseCRLNumber** identifica el número de CRL de la CRL básica que se utilizó en el punto de partida de la generación de esta dCRL. La CRL referenciada será una CRL que esté completa para el ámbito correspondiente.

Esta extensión siempre es crítica. Un usuario de certificado que no comprende la utilización de las dCRL no debe emplear una CRL que contenga esta extensión, puesto que la CRL puede no estar tan completa como el usuario espera.

8.6.2.5 Extensión de actualización básica

Este campo de actualización básica se utiliza en las dCRL y se emplea para identificar la fecha/hora a partir de la cual esta dCRL proporciona actualizaciones al estado de revocación. Esta extensión sólo se utilizará en las dCRL que contengan la extensión **deltaCRLIndicator**. Una dCRL que contenga en su lugar la extensión **crlScope** no precisa esta extensión, puesto que el campo **baseThisUpdate** de la extensión **crlScope** se puede utilizar para los mismos fines.

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

Esta extensión siempre es no crítica.

8.6.2.6 Extensión de CRL más reciente

La extensión de CRL más reciente puede utilizarse como un certificado o como una extensión de CRL. En los certificados, esta extensión puede utilizarse en certificados expedidos a autoridades así como en certificados expedidos a usuarios. Este campo identifica la CRL a la que debe hacer referencia un usuario de certificado para obtener la información de revocación más reciente (por ejemplo, la última dCRL). Este campo se define como sigue:

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

El valor del tipo **CRLDistPointsSyntax** es el que se define en la extensión de puntos de distribución de CRL en 8.6.2.1.

A opción del expedidor del certificado, esta extensión puede ser crítica o no crítica. Si la extensión CRL mas reciente se indica como crítica, el sistema que utiliza certificados no debe emplear el certificado sin extraer primero y comprobar la CRL más reciente. Si la extensión se indica como no crítica mediante banderas, el sistema que utiliza certificados puede utilizar medios locales para determinar si es preciso comprobar o no la CRL más reciente.

8.6.2.7 Extensión del punto de distribución expedidor AA

Este campo de extensión CRL identifica el punto de distribución CRL para los certificados de atributo de esta CRL particular, e indica si la CRL es indirecta, o está limitada a abarcar solamente un subconjunto de la información de revocación. La limitación se puede basar en un subconjunto del contenido del certificado o en un subconjunto de motivos de revocación. La CRL está firmada por la clave privada del expedidor de CRL – los puntos de distribución de CRL no tienen sus propios pares 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 del directorio del expedidor de la CRL. Si la CLR no contiene la extensión de punto de distribución expedidor, ni la extensión de punto de distribución de expedidor AA, ni tampoco el campo ámbito de CRL, la CRL contendrá asientos para todos los certificados de atributo que no hayan expirados y que son revocados emitidos por el expedidor de CRL. Si la CLR no contiene el campo de punto de distribución expedidor AA ni el campo de ámbito de CRL están ausentes, pero, sin embargo, contiene el punto de distribución expedidor, el ámbito de la CRL no incluirá los certificados de atributo.

Cuando aparece un certificado en una CRL, es posible borrarlo de las CRL posteriores después de su expiración.

Este campo se define del modo siguiente:

```
AAissuingDistributionPoint EXTENSION  ::= {
    SYNTAX AAIssuingDistPointSyntax
    IDENTIFIED BY id-ce-AAissuingDistributionPoint }

AAIssuingDistPointSyntax  ::= SEQUENCE {
    distributionPoint              [ 0 ] DistributionPointName OPTIONAL,
    onlySomeReasons                [ 1 ] ReasonFlags OPTIONAL,
    indirectCRL                    [ 2 ] BOOLEAN DEFAULT FALSE,
    containsUserAttributeCerts     [ 3 ] BOOLEAN DEFAULT TRUE,
    containsAACerts                [ 4 ] BOOLEAN DEFAULT TRUE,
    containsSOAPublicKeyCerts      [ 5 ] BOOLEAN DEFAULT TRUE }
```

La componente **distributionPoint** contiene el nombre del punto de distribución en una o varias formas de nombre. Si el campo contiene **onlySomeReasons**, la CRL contendrá únicamente las revocaciones para los certificados de atributo correspondientes al motivo o motivos identificados, de lo contrario la CRL contendrá revocaciones para todos los motivos.

Si **indirectCRL** es verdadero, es probable que la CRL contenga notificaciones de revocación para certificados de atributo de autoridades diferentes de aquella que expide la CRL. La autoridad responsable de cada asiento viene indicada por la extensión de asientos CRL del expedidor de certificado en dicho asiento o de conformidad con las reglas por defecto descritas en 8.6.2.3. En dicha CRL, es responsabilidad del expedidor CRL garantizar que ésta sea completa en el sentido de que contenga todos los asientos de revocación, coherente con los indicadores **containsUserAttributeCerts**, **containsAACerts, containsSOAPublicKeyCerts** y **onlySomeReasons**, de todas las autoridades que identifican este expedidor CRL en su certificado de atributo.

Si **containsUserAttributeCerts** es verdadero, la CRL contendrá revocaciones para certificados de atributo expedidos a entidades extremas que no son AA. Si **containsAACerts** es verdadero, la CRL contendrá revocaciones para certificados de atributo expedidos a entidades que sí son AA.

Si **containsSOAPublicKeyCerts** es verdadero, la CRL contendrá revocaciones para certificados de clave pública expedidos a una entidad que es una SOA a efectos de gestión de privilegios (es decir certificados que contienen la extensión **SOAIdentifier**). En el caso de CRL distribuidas a través del directorio se aplican las siguientes reglas: si la

CRL es una dCRL, se distribuirá mediante el atributo **deltaRevocationList** del punto de distribución asociado o, si no se ha identificado punto de distribución, a través del atributo **deltaRevocationList** del asiento de expedidor de CRL, independientemente de la configuración de los tipos de certificados abarcados por la CRL. Al menos que la CRL sea una CRL:

- Se distribuirá una CRL que no contenga una extensión **issuingDistributionPoint** que tenga fijado únicamente **containsAACerts** y/o **containsSOAPublicKeyCerts**, mediante el atributo **attributeAuthorityRevocationList** del punto de distribución asociado o, si no se ha identificado punto de distribución, a través del atributo **attributeAuthorityRevocationList** del asiento de emisor de CRL.
- Se distribuirá una CRL que no contenga una extensión **issuingDistributionPoint** que tenga fijado **containsUserAttributeCerts** (con o sin **containsAACerts** y/o **containsSOAPublicKeyCerts** también fijado), mediante el atributo **attributeCertificateRevocationList** del punto de distribución asociado o, si no hay un punto de distribución identificado, a través del atributo **attributeCertificateRevocationList** del asiento de expedidor de CRL.
- Se distribuirá, de conformidad con 8.6.2.2, una CRL que contenga una extensión **issuingDistributionPoint**.

Esta extensión es siempre es crítica. Un usuario de certificado que no entienda esta extensión no puede suponer que la CRL contiene una lista completa de certificados revocados de la autoridad identificada. La CRL que no contengan extensiones críticas contendrán todos los asientos actuales de CRL para todos los certificados de usuario y de autoridad revocados.

NOTA 1 – Los mecanismos mediante los cuales las autoridades comunican la información de revocación a la CRL están fuera del alcance de esta Especificación de directorio.

NOTA 2 – Cuando una autoridad publique una CRL con **containsAACerts** fijado a **verdadero** y **containsUserAttributeCerts** no puesto a **verdadero**, la autoridad garantizará que los certificados AA cubiertos por esta CRL contengan la extensión **basicAttConstraints**.

NOTA 3 – Cuando una autoridad publica una CRL con **containsSOAPublicKeyCerts** puesto a **verdadero**, garantizará que todos los certificados SOA abarcados por esta CRL contengan la extensión **SOAIdentifier**.

9 Relación entre la CRL delta y la básica

Una dCRL incluye una extensión **deltaCRLIndicator** o **crlScope** para indicar la información de revocación básica que se está actualizando con esta dCRL.

Si está presente **deltaCRLIndicator** en una dCRL, la información de revocación básica que se está actualizando es la CRL básica referenciada en dicha extensión. La CRL básica referenciada por una extensión **deltaCRLIndicator** será una CRL que se ha expedido como completa para este ámbito (es decir, no es una dCRL en sí misma).

Si la extensión **crlScope** está presente e incluye el componente **baseRevocationInfo** para referenciar la información de revocación básica que está siendo actualizada, se trata de una referencia a un determinado punto en el tiempo a partir del cual esta dCRL proporciona actualizaciones. La componente **baseRevocationInfo** hace referencia a una CRL que puede o no haber sido expedida como completa para dicho ámbito (es decir, la CRL referenciada puede haberse expedido únicamente como una dCRL). Sin embargo, la dCRL que contiene el componente **baseRevocationInfo** actualiza la información de revocación que está completa para el ámbito de la CRL referenciada en el instante en que la CRL referenciada se expidió. El usuario del certificado puede aplicar la dCRL a una CRL que está completa para el ámbito correspondiente y que se expidió en el mismo instante o después de que se expidiera la CRL referenciada en la dCRL que incluye al componente **baseRevocationInfo**.

Debido a la posibilidad de información contradictoria, una CRL no debe contener una extensión **deltaCRLIndicator** y una extensión **crlScope** con el componente **baseRevocationInfo**. Una CRL puede contener la extensión **deltaCRLIndicator** y la extensión **crlScope** únicamente si el componente **baseRevocationInfo** no está presente en la extensión **crlScope**.

Una dCRL puede también ser una CRL indirecta, puesto que puede contener información de revocación actualizada relacionada con las CRL básicas expedidas por una o más autoridades. La extensión **crlScope** se utilizará como medio para identificar una CRL como una dCRL indirecta. La extensión **crlScope** contendrá un ejemplar del componente **PerAuthorityScope** por cada CRL básica para la cual la dCRL indirecta proporciona información actualizada.

La aplicación de una dCRL a la información de revocación básica referenciada tendrá que reflejar de forma precisa el estado actual de revocación.

– Puede aparecer una notificación de revocación de certificado, con el motivo de revocación **certificateHold,** ya sea en una dCRL o en una CRL que esté completa para un determinado ámbito. Este código de motivo pretende indicar una revocación temporal del certificado, pendiente de una decisión ulterior de revocar permanentemente el certificado o reinstalarlo como uno que no esté revocado.

- Si un certificado se enumeró como revocado con el motivo de revocación **certificateHold** en una CRL (ya sea dCRL o una CRL que está completa para un determinado ámbito), cuyo **cRLNumber** es *n*, y la retención se libera en consecuencia, se tendrá que incluir el certificado en todas las dCRL expedidas después de liberar la retención cuando el **cRLNumber** de la CRL básica referenciada es inferior o igual a *n*. Dependiendo de la extensión utilizada para indicar que esta CRL es una dCRL, el número de CRL de una CRL básica referenciada es el valor del componente **BaseCRLNumber** de la extensión **deltaCRLIndicator** o el elemento **cRLNumber** del componente **BaseRevocationInfo** de la extensión **cRLScope.** El certificado tendrá que poner en una lista con el motivo de revocación **removeFromCRL** a menos que el certificado se vuelva a revocar en consecuencia por uno de los motivos de revocación incluidos en la dCRL, en cuyo caso, el certificado se tendrá que poner en una lista con el motivo de revocación adecuado para la revocación subsiguiente.
- Si el certificado no se suprimió de la retención, pero se revocó permanentemente, se tendrá que indicar en todas las dCRL subsiguientes en las que **cRLNumber** de la CRL básica referenciada sea inferior al **cRLNumber** de la CRL (ya sea una dCRL o una CRL que está completa para un determinado ámbito) en la que apareció por primera vez la notificación de revocación permanente. Dependiendo de la extensión utilizada para indicar que esta CRL es una dCRL, el número de CRL de una CRL básica referenciada será el valor del componente básico **BaseCRLNumber** de la extensión **deltaCRLIndicator** o el elemento **cRLNumber** del componente **BaseRevocationInfo** de la extensión **cRLScope**.

– Una notificación de revocación de certificado puede aparecer en primer lugar en una dCRL y es posible que caduque el periodo de validez del certificado antes de expedir la siguiente CRL que está completa para el ámbito considerado. En este caso, la notificación de revocación tendrá que incluirse en todas las dCRL subsiguientes hasta que esa notificación de revocación esté incluida en por lo menos una CRL expedida que esté completa para el ámbito de dicho certificado.

Una CRL que está completa para un determinado ámbito, en el instante vigente, se puede construir localmente de cualesquiera de las formas siguientes:

– recuperando la dCRL actual para dicho ámbito y combinándola con una CRL expedida que está completa para dicho ámbito y que tiene un **cRLNumber** superior o igual al **cRLNumber** de la CRL básica referencia en la dCRL; o

– recuperando la CRL actual para dicho ámbito y combinándola con una CRL construida localmente que está completa para dicho ámbito y que se construyó con una dCRL que tiene un **cRLNumber** superior o igual al **cRLNumber** de la CRL básica referenciada en la dCRL actual.

10 Procedimiento de procesamiento del trayecto de certificación

El procesamiento del trayecto de certificación se realiza en un sistema que tendrá 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.

10.1 Entradas al procesamiento del trayecto

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

a) un conjunto de certificados que comprenden un trayecto de certificación;

NOTA – Cada certificado de un trayecto de certificación es único. Un trayecto de certificación que contiene el mismo certificado dos veces o más no es un trayecto de certificación válido.

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*, pero no puede ser nulo;

d) un valor de indicador *política explícita inicial*, que indica si un identificador de política aceptable tendrá que aparecer explícitamente en el campo de extensión de políticas de certificado de todos los certificados en el trayecto;

e) un valor de indicador *inhibición de correspondencia de políticas inicial*, que indica si la correspondencia de políticas está prohibida en el trayecto de certificación;

f) un valor de indicador *inhibición de política inicial*, cualquier política, que indica si el valor especial **anyPolicy**, caso de figurar en una extensión de políticas de certificación, se considera una correspondencia para cualquier valor específico de políticas de certificación en un conjunto constreñido;

g) la fecha/hora actual (si no está disponible internamente en el módulo de procesamiento de trayecto de certificación);

h) un *conjunto de subárboles permitidos inicial* que contiene un conjunto inicial de especificaciones de subárbol que define subárboles en los cuales se permiten nombres de sujeto (con la forma de nombre utilizada para especificar los subárboles). En los certificados del trayecto de certificación todos los nombres de sujeto de una forma de nombre determinada, para los que se definen subárboles permitidos iniciales, caerán dentro del conjunto de subárboles permitidos para esa forma de nombre dada. Esta entrada también puede contener el valor especial ilimitado para indicar que todos los nombres de sujeto son aceptables inicialmente. En la cláusula 10, los nombres de sujeto son aquellos valores de nombre que aparecen en el campo sujeto de la extensión subjectAltName;

i) un *conjunto de subárboles excluidos inicial* que contiene un conjunto inicial de especificaciones de subárbol que define subárboles en los cuales no pueden hallarse los nombres de sujeto de los certificados del trayecto de certificación. Esta entrada también puede ser un conjunto vacío para indicar que inicialmente no hay en vigor exclusiones de subárboles;

j) unas *formas de nombre requeridas iniciales* que contienen un conjunto inicial de formas de nombre que indican que todos los certificados del trayecto deben incluir un nombre de sujeto de al menos una de las formas de nombre especificadas. Esta entrada también puede ser un conjunto vacío para indicar que los nombres de objeto de los certificados no requieren formas de nombre específicas.

Los valores de c), d), e) y f) 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.

Cabe destacar que puesto que hay entradas individuales al proceso de validación del trayecto, un usuario de certificado puede limitar la confianza que tiene en cualquier clave pública de confianza dada por un conjunto determinado de políticas de certificados. Esto se puede lograr asegurando que una clave pública determinada es la entrada al proceso sólo cuando la entrada conjunto de política inicial incluye políticas para las que el usuario de certificado confía en dicha clave pública. Puesto que otra entrada al proceso es el propio trayecto de certificación, se podría realizar este control de transacción por transacción.

10.2 Salidas del procesamiento de trayecto

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) el conjunto de políticas de autoridades constreñidas y sus calificadores asociados según los cuales es válido el trayecto de certificación, o el valor especial *cualquier política*;

d) el conjunto de políticas de usuario constreñido, formado a partir de la intersección del *conjunto de políticas constreñidas por las autoridades* y del *conjunto de política inicial*;

e) el *indicador política explícita*, que indica si el usuario de certificado o una autoridad en el trayecto necesita que se identifique una política aceptable para cualquier certificado en el trayecto; y

f) 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.

10.3 Variables del procesamiento del trayecto

El procedimiento utiliza el siguiente conjunto de variables de estado:

a) *conjunto de políticas constreñidas por las autoridades*: Un cuadro de identificadores y calificadores de política de los certificados del trayecto de certificación (las filas representan políticas, sus calificadores y el historial de correspondencias, y las columnas representan certificados en el trayecto de certificación);

b) *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 necesitan tomar el valor especial *ilimitado*;

c) *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;

d) *formas de nombre requeridas*: Un conjunto (que puede estar vacío) de conjuntos de formas de nombre. Para cada conjunto de formas de nombre, todo certificado subsiguiente debe contener un nombre de una de las formas de nombre del conjunto.

e) *indicador política explícita*: Indica si una política aceptable necesita ser explícitamente identificada en cada certificado del trayecto;

f) *profundidad del trayecto*: Número entero siguiente al número de certificados en el trayecto de certificación para los que se ha completado el procesamiento;

g) *indicador inhibición de correspondencia de políticas*: Indica si se inhibe la correspondencia de políticas;

h) *indicador inhibición de cualquier política*: Indica el valor especial **anyPolicy** que se considera una correspondencia para cualquier política específica de certificado;

i) *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 número entero denominado s*alto de certificados* que da el número de certificados que hay que saltar antes de que se aplique la constricción.

10.4 Paso de inicialización

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

a) escribir *cualquier política* en las columnas cero y primera de la fila cero del cuadro *conjunto de políticas constreñidas por la autoridad*;

b) inicializar la variable *subárboles permitidos* al valor *conjunto de subárboles permitidos inicial*;

c) inicializar la variable *subárboles excluidos* al valor *conjunto de subárboles excluidos inicial*;

d) inicializar la variable *formas de nombre requeridas* al valor *formas de nombre requeridas iniciales*;

e) inicializar el *indicador política explícita* al valor *política explícita inicia*l

f) inicializar *profundidad de trayecto* a uno;

g) inicializar el *indicador de inhibición de correspondencia de políticas* al valor *inhibición de correspondencia de políticas inicial*;

h) inicializar el *indicador inhibición de cualquier política* al valor *inhibición de cualquier política inicial*;

i) inicializar los tres indicadores *constricciones pendientes* a no fijados.

10.5 Procesamiento de certificado

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.

10.5.1 Comprobaciones básicas de certificado

Se aplican las siguientes comprobaciones a un certificado. Si se encuentran certificados autofirmados en el trayecto, se ignoran.

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 de versión 3, comprobar que **basicConstraints** está presente y que el componente **cA** en la extensión **basicConstraints** es **TRUE** (verdadero). Si el componente **pathLenConstraint** está presente, comprobar que el trayecto de certificación actual no viola esa constricción (ignorando certificados autoexpedidos intermedios).

c) Si no está presente la extensión de políticas de certificados, fijar a cero el *conjunto de políticas constreñidas por las autoridades* suprimiendo todas las filas del cuadro *conjunto de políticas constreñidas por las autoridades*.

d) Si está presente la extensión de políticas de certificado, para cada política, *P*, en la extensión que no sea **anyPolicy**, habrá que adjuntar los calificadores de política asociados con *P* a cada fila del cuadro *conjunto de políticas constreñidas de las autoridades*, cuya casilla correspondiente a la columna [*profundidad de trayecto*] contenga el valor *P*. De no haber ninguna fila en el cuadro de conjunto de *políticas constreñidas por las autoridades* que contenga *P* en su casilla correspondiente a la columna [*profundidad de trayecto*], pero si el valor que figura en el *conjunto de políticas constreñidas por las autoridades*[*0, profundidad de trayecto*] es *cualquier política*, habrá que añadir una nueva fila en el cuadro, duplicando la fila cero e incluyendo el identificador de política *P* junto con sus cualificadores en la casilla correspondiente a la columna [*profundidad de trayecto*] de la nueva fila.

e) Si está presente la extensión de políticas de certificado y no incluye el valor **anyPolicy** o se ha fijado el indicador *inhibición de cualquier política* y el certificado no es un certificado intermedio autoexpedido, suprimir cualquier fila en cuya casilla correspondiente a la columna [*profundidad de trayecto*] figure el valor *cualquier política*, así como cualquier columna cuya casilla correspondiente a la columna [*profundidad de trayecto*] no contenga ninguno de los valores de extensión de políticas de certificado.

f) Si está presente la extensión de políticas de certificado y ésta incluye el valor **anyPolicy** y el *indicador inhibición de cualquier política* no se ha fijado, habrá que adjuntar los calificadores de política asociados con **anyPolicy** a cada fila del cuadro *conjunto de políticas constreñidas de las autoridades* cuyas casillas correspondientes a la columna [*profundidad de trayecto*] contengan el valor *cualquier política* o un valor que no aparezca en la extensión de políticas de certificado.

g) Si el certificado no es un certificado autoexpedido intermedio, comprobar que el nombre de sujeto está dentro del espacio de nombre dado por el valor de *subárboles permitidos* y no se encuentra dentro del espacio de nombre dado por el valor *subárboles excluidos*.

h) Si el certificado no es un certificado autoexpedido intermedio, y si *formas de nombre requeridas* no es un conjunto vacío, para cada conjunto de formas de nombre de *formas de nombre requeridas* comprobar que hay un nombre de sujeto en el certificado de una de las formas de nombre del conjunto.

10.5.2 Procesamiento de certificados intermedios

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. Si se encuentran certificados autofirmados en el trayecto, se ignoran.

a) Si la extensión **nameConstraints** con un componente **permittedSubtrees** está presente en el certificado, fijar la variables 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 le 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 la extensión **nameConstraints** con un componente **requiredNameForms** está presente en el certificado, la variable formas de nombre requeridas se fija a la unión (lógica) de su valor anterior y el conjunto constituido por el conjunto de formas de nombre especificadas en la extensión de certificado. Si el componente **requiredNameForms** contiene más de una forma de nombre, la variable formas de nombre requeridas señalará que un nombre de al menos una de las formas de nombre indicadas en esta extensión estará presente en todos los certificados subsiguientes. La unión (lógica) de un valor de la variable formas de nombre requeridas y el valor de la extensión del certificado actual es un conjunto de conjuntos que señalan requisitos para todos los certificados subsiguientes. Por ejemplo, si el formas de nombre requeridas actual se fija de manera que requiera que ya sea un nombre DN o un nombre rfc822 esté presente en certificados y la extensión actual en el certificado que se está procesando indica que se requieren ya sean nombres rfc822 o nombres DNS, la unión (lógica) resultante que es el nuevo formas de nombre requeridas indica que cada uno de los certificados subsiguientes debe tener ya sea un nombre rfc822 o ambos nombres, es decir, un nombre DN y un nombre DNS.

d) Si está fijado el *indicador inhibición de correspondencia de políticas*:

   – procesar cualquier extensión de correspondencias de políticas situando, para cada correspondencia identificada en la extensión, todas las filas del cuadro *conjunto de políticas constreñidas por las autoridades* cuyos valores de la columna *[profundidad de trayecto*] es igual al valor de política de dominio de expedidor en la extensión y borrar la fila.

e) Si el *indicador inhibición de correspondencia de políticas* no está fijado:

- procesar cualquier extensión de correspondencias de políticas situando, para cada correspondencia identificada en la extensión, todas las filas del cuadro *conjunto de políticas constreñidas de las autoridades* cuyo valor de la columna [*profundidad de trayecto*] es igual al valor de política de dominio de expedidor en la extensión, y escribir el valor de política de dominio de sujeto proveniente de la extensión en la columna [*profundidad de trayecto*+1] de la misma fila. Si la extensión corresponde a una política de dominio de expedidor con más de una política de dominio de sujeto, entonces la fila afectada deberá copiarse y se deberá añadir una nueva entrada a cada fila. Si el valor del *conjunto de políticas constreñidas por las autoridades*[0, *profundidad de trayecto*] es *cualquier política*, escribir entonces cada identificador de política de dominio de expedidor proveniente de la extensión correspondencias de políticas en la columna [*profundidad de trayecto*], duplicando filas cuando sea necesario y manteniendo calificadores si existen, y escribir el valor de la política de dominio de sujeto proveniente de la extensión en la columna [*profundidad de trayecto*+1] de la misma fila;
- si el indicador *inhibición de correspondencia de políticas pendiente* está fijado y el certificado no es autoexpedido, disminuir el correspondiente valor *salto de certificados* y, si este valor llega a cero, fijar el indicador *inhibición de correspondencia de políticas*;
- si la constricción **inhibitPolicyMapping** está presente en el certificado hacer los siguiente: para un valor de **SkipCerts** de 0, fijar el *indicador inhibición de correspondencia de políticas*. Para cualquier otro valor **SkipCerts**, fijar el indicador *inhibición de correspondencia de políticas pendiente*, y fijar el correspondiente valor *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).

f) Para cualquier fila no modificada en ninguno de los pasos c) o d), anteriores (y para cualquier fila en el caso de que no esté presente una extensión de correspondencia de certificado), escribir el identificador de política de la columna [*profundidad de trayecto*] en la columna [*profundidad de trayecto*+1] de la fila.

g) Si el *indicador inhibición de cualquier política* no está fijado:

- si está fijado el indicador *inhibición de cualquier política pendiente* y el certificado no es un certificado autoexpedido, disminuir el valor correspondiente *salto de certificados* y, si este valor llega a cero, fijar el *indicador inhibición de cualquier política*;
- si la constricción **inhibitAnyPolicy** está presente en el certificado, realizar la siguiente operación. Para un valor 0 **SkipCerts**, fijar el *indicador inhibición de cualquier política*. Para cualquier otro valor **SkipCerts**, fijar el indicador *inhibición de cualquier política pendiente*, y fijar el valor correspondiente *salto de certificados* al valor **SkipCert** o al valor anterior *salto de certificados*, el que sea menor (si el indicador *inhibición de cualquier política pendiente* estaba ya fijado).

h) Incrementar [*profundidad de trayecto*].

10.5.3 Procesamiento de indicador política explícita

Para todos los certificados, se realizan entonces las actuaciones siguientes:

a) si el *indicador política explícita* no está fijado:

- si está fijado el indicador *política explícita pendiente* y el certificado no es un certificado intermedio autoexpedido, disminuir el valor correspondiente *salto de certificado* y, si este valor llega a cero, fijar el *indicador política explícita*;
- si la constricción **requireExplicitPolicy** está presente en el certificado, realizar la siguiente operación. Para un valor 0 **SkipCerts**, fijar *el indicador política explícita*. Para cualquier otro valor **SkipCerts**, fijar el *indicador política explícita pendiente* y fijar el valor correspondiente *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);
- si el componente **requireExplicitPolicy** y el trayecto de certificación incluye un certificado expedido por una CA designada, es necesario que todos los certificados en el trayecto contengan en la extensión de políticas de certificados un identificador aceptable de políticas. Un identificador aceptable de políticas es un identificador de la política de certificados requerida por el usuario del trayecto de certificación, el identificador de una política que haya sido declarada equivalente, mediante la correspondencia de políticas, o cualquier política. La CA designada es la CA expedidora del certificado que contiene esta extensión (si el valor de **requireExplicitPolicy** es 0) o una CA sujeto de un certificado ulterior en el trayecto de certificación (según se indique mediante un valor no nulo).

10.5.4 Procesamiento final

Una vez se hayan tramitado todos los certificados en el trayecto, se realizan las siguientes acciones:

a) Determinar el *conjunto de políticas constreñidas por las autoridades* a partir del cuadro correspondiente. Si éste está vacío, dicho conjunto es vacío o nulo. Si el *conjunto de políticas constreñidas por las autoridades*[*0, profundidad de trayecto*] es *cualquier política* el *conjunto de políticas constreñidas por las autoridades* es *cualquier política*. De lo contrario, dicho conjunto es, para cada fila en el cuadro, el valor en la célula más a la izquierda que no contenga el identificador *cualquier política*.

b) Calcular el conjunto de políticas constreñidas por las autoridades mediante la intersección de los conjuntos *de políticas constreñidas por las autoridades* y *de política inicial*.

c) Si está puesto el *indicador política explícita*, verificar que ni el *conjunto de políticas constreñidas por las autoridades* ni el *conjunto de políticas constreñidas por el usuario* sean vacíos.

Si falla cualquiera de las pruebas anteriores, se ha de terminar el procedimiento, retornar una indicación de fallo, un código de motivo adecuado *el indicador política explícita*, el *conjunto de política constreñida por las autoridades* y el *conjunto de políticas constreñidas por el usuario*. Si el fallo se debe a que el *conjunto de políticas constreñidas por el usuario* está vacío, el trayecto es válido conforme a la(s) política(s) constreñida(s) por las autoridades, pero inaceptable para el usuario.

Si ninguna de estas pruebas falla en el certificado final, se termina el procedimiento, se retorna una indicación de éxito junto con el *indicador política explícita*, el *conjunto de políticas constreñidas por las autoridades* y el *conjunto de políticas constreñidas por el usuario*.

11 Esquema del directorio PKI

Esta cláusula define los elementos del esquema de directorio utilizados para representar información PKI en el directorio. Incluye la especificación de clases de objeto importantes, atributos y reglas de concordancia de valores de atributo.

11.1 Clases y formas de nombre de objeto de directorio PKI

Esta subcláusula incluye la definición de clases de objeto utilizadas para representar objetos PKI en el directorio.

11.1.1 Clase de objeto de usuario PKI

La clase de objeto de usuario PKI se utiliza para definir asientos para objetos que pueden ser sujetos de certificados de clave pública.

```
pkiUser    OBJECT-CLASS   ::= {
    SUBCLASS OF       {top}
    KIND              auxiliary
    MAY CONTAIN       {userCertificate}
    ID                id-oc-pkiUser }
```

11.1.2 Clase de objeto de usuario PKI de CA

La clase de objeto PKI de CA se utiliza para definir asientos para objetos que actúan como autoridades de certificación.

```
pkiCA      OBJECT-CLASS   ::= {
    SUBCLASS OF       {top}
    KIND              auxiliary
    MAY CONTAIN       {cACertificate |
                       certificateRevocationList |
                       authorityRevocationList |
                       crossCertificatePair }
ID                    id-oc-pkiCA }
```

11.1.3 Clases y formas de nombre de objeto punto de distribución de CRL

La clase de objeto punto de distribución de CRL se utiliza para definir asientos para objetos que actúan como puntos de distribución de CRL.

```
cRLDistributionPoint      OBJECT-CLASS        ::= {
    SUBCLASS OF           { top }
    KIND                  structural
    MUST CONTAIN          { commonName }
    MAY CONTAIN           { certificateRevocationList |
```

```
                        authorityRevocationList |
                        deltaRevocationList }
    ID                  id-oc-cRLDistributionPoint }
```

La forma de nombre punto de distribución de CRL especifica como se pueden denominar los asientos de clases de objeto **cRLDistributionPoint**.

```
cRLDistPtNameForm       NAME-FORM ::= {
    NAMES               cRLDistributionPoint
    WITH ATTRIBUTES     { commonName}
    ID                  id-nf-cRLDistPtNameForm }
```

11.1.4 Clase de objeto de CRL delta

La clase de objeto de CRL delta se utiliza para definir asientos para objetos que albergan listas de revocación delta (por ejemplo, CA, AA, etc.)

```
deltaCRL                OBJECT-CLASS   ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         {deltaRevocationList}
    ID                  id-oc-deltaCRL }
```

11.1.5 Clases de objeto de política de certificados y CPS

La clase de objeto CPS se utiliza para definir asientos para objetos que contienen información de política de certificados y/o de prácticas de certificación.

```
cpCps                   OBJECT-CLASS   ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         {certificatePolicy |
                        certificationPracticeStmt}
    ID                  id-oc-cpCps }
```

11.1.6 Clase de objeto de trayecto de certificados PKI

La clase de objeto de trayecto de certificados PKI se utiliza para definir asientos para objetos que contienen trayectos PKI. Se utilizarán normalmente junto a asientos de estructuras **pkiCA** o **pkiUser**.

```
pkiCertPath             OBJECT-CLASS   ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         { pkiPath }
    ID                  id-oc-pkiCertPath }
```

11.2 Atributos de directorio PKI

Esta subcláusula incluye la definición de atributos de directorio para almacenar elementos de información PKI en el directorio.

11.2.1 Atributo de certificado de usuario

Un usuario puede obtener uno o más certificados de clave pública de una o más autoridades de certificación. El tipo de atributo **userCertificate** contiene los certificados de clave pública que un usuario ha obtenido de una o más CA.

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

11.2.2 Atributo de certificado de CA

El atributo **cACertificate** de un asiento de directorio de CA se utilizará para almacenar certificados autoexpedidos (si existen) y certificados expedidos a esta CA por otras CA en el mismo entorno que esta CA. En el caso de certificados v3, estos certificados incluirán una extensión **basicConstraints** con un valor **cA** fijado a **VERDADERO**. La definición del entorno es únicamente una cuestión de política local.

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

11.2.3 Atributo de pares de certificados cruzados

Los elementos **issuedToThisCA** del atributo **crossCertificatePair** de un asiento de directorio de CA se utilizarán para almacenar todos los certificados salvo los autoexpedidos a esta CA. Opcionalmente, los elementos **issuedByThisCA** del atributo **crossCertificatePair** de un asiento de directorio de CA pueden contener un subconjunto de certificados expedidos por esta CA a otras CA. Si una CA expide un certificado a otra CA, y esta CA no está jerárquicamente subordinada a la CA expedidora, la CA expedidora deberá colocar ese certificado en el elemento **issuedByThisCA** del atributo **crossCertificatePair** en el asiento de su propio directorio. Cuando están presentes tanto los elementos **issuedToThisCA** como **issuedByThisCA** en un único valor de atributo, el nombre de expedidor en un certificado coincidirá con el nombre correspondiente en el otro y viceversa, y la correspondiente clave pública en un certificado debe permitir verificar la firma digital del otro certificado y viceversa. En las ediciones anteriores se utilizó el término **forward** en lugar de **issuedToThisCA** y el término **reverse** en lugar de **issuedByThisCA**.

Cuando un elemento **issuedByThisCA** está presente, no es necesario almacenar en el mismo valor de atributo el valor de elemento **issuedToThisCA** ni el valor de elemento **issuedByThisCA**; dicho de otra forma, pueden almacenarse en un solo valor de atributo o en dos valores de atributo.

En el caso de certificados v3, estos incluirán una extensión **basicConstraints** con el valor **cA** fijado a **VERDADERO**.

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

CertificatePair         ::=     SEQUENCE {
issuedToThisCA          [0]     Certificate OPTIONAL,
issuedByThisCA          [1]     Certificate OPTIONAL
                                -- at least one of the pair shall be present -- }
(WITH COMPONENTS { ..., issuedToThisCA PRESENT} |
WITH COMPONENTS { ..., issuedByThisCA PRESENT})
```

11.2.4 Atributo de lista de revocación de certificados

El atributo siguiente incluye una lista de certificados revocados.

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

11.2.5 Atributo de lista de revocación de autoridades

El atributo siguiente contiene una lista de certificados de autoridad revocados.

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

11.2.6 Atributo de lista de revocación delta

Se define el siguiente tipo de atributo para retener una dCRL en un asiento de directorio.

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

11.2.7 Atributo de algoritmos soportados

Se define un atributo de directorio para soportar la selección de un algoritmo para su utilización cuando se comunica con una entidad final distante que utiliza los certificados que se definen en esta Especificación de directorio. La ASN.1 siguiente 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 del atributo de múltiples valores tendrá un valor distinto de **algorithmIdentifier**. El valor del componente **intendedUsage** proporciona una indicación de la utilización que se pretende del algoritmo (véase 8.2.2.3 para las utilizaciones reconocidas). El valor del componente **intendedCertificatePolicies** identifica las políticas de certificados y, como opción, los calificadores de política de certificado con los que se puede utilizar el algoritmo identificado.

11.2.8 Atributo de declaración de práctica de certificación

El atributo **certificationPracticeStmt** se utiliza para almacenar información sobre una declaración de práctica de certificación de autoridad.

```
certificationPracticeStmt     ATTRIBUTE  ::=  {
   WITH SYNTAX        InfoSyntax
   ID                 id-at-certificationPracticeStmt }

InfoSyntax          ::=       CHOICE {
   content          DirectoryString {ub-content},
   pointer          SEQUENCE {
      name                        GeneralNames,
      hash                        HASH { HashedPolicyInfo } OPTIONAL } }

   POLICY   ::=  TYPE-IDENTIFIER

HashedPolicyInfo      ::=   POLICY.&Type( {Policies} )

Policies POLICY ::= {...} -- Defined by implementors --
```

Si está presente **content**, está incluido el contenido completo de la declaración de práctica de certificación de autoridad.

Si está presente **pointer**, el componente **name** hace referencia a una o más ubicaciones en las que se puede encontrar una copia de la declaración de práctica de certificación. Si está presente el componente **hash**, contiene el TROCEO del contenido de la declaración de práctica de certificación que se encontrará en una ubicación referenciada. Este troceo se puede utilizar para realizar una comprobación completa del documento referenciado.

11.2.9 Atributo de política de certificados

El atributo **certificatePolicy** se utiliza para almacenar información sobre una política de certificado.

```
certificatePolicy          ATTRIBUTE  ::=  {
   WITH SYNTAX        PolicySyntax
   ID                 id-at-certificatePolicy }

PolicySyntax  ::=          SEQUENCE {
   policyIdentifier        PolicyID,
   policySyntax            InfoSyntax
}

   PolicyID   ::=  CertPolicyId
```

El componente **policyIdentifier** incluye el identificador de objeto registrado para la política de certificado determinada.

Si está presente **content**, se incluye el contenido completo de la política de certificado.

Si está presente **pointer**, el componente **name** hace referencia a una o más ubicaciones en las que se puede encontrar una copia de la política de certificado. Si está presente el componente **hash,** contiene el TROCEO del contenido de la política de certificados que se encontrará en una ubicación referenciada. Este troceo se puede utilizar para realizar una comprobación completa del documento referenciado.

NOTA – La opción de incluir un troceo en este atributo es únicamente para poder realizar una comprobación total de los datos localizados en una fuente distinta del directorio. El TROCEO almacenado en el directorio debe ser protegido. Para este fin, podrían utilizarse los servicios de seguridad de directorio, incluyendo la autenticación fuerte y el control de acceso y/o atributos firmados. Además, aun si el TROCEO concuerda con el documento original CP/CPS, existen requisitos de seguridad adicionales para poder garantizar que la propia especificación original es el documento correcto (por ejemplo, el documento está firmado por una autoridad pertinente).

11.2.10 Atributo de trayecto PKI

El atributo de trayecto PKI se utiliza para almacenar trayectos de certificados, constituidos por una secuencia de certificados.

```
pkiPath    ATTRIBUTE   ::= {
   WITH SYNTAX         PkiPath
   ID                  id-at-pkiPath }
```

Este atributo se puede almacenar en un asiento de directorio de la clase de objeto **pkiCA** o **pkiUser**.

Cuando se almacenan en asientos **pkiCA**, valores de este atributo contienen trayectos de certificación que excluyen certificados de entidad final. Como tal, el atributo se utiliza para almacenar trayectos de certificación que son utilizados frecuentemente por partes confiantes asociadas con esa CA. Un valor de este atributo se puede utilizar junto con cualquier certificado de entidad final expedido por el último sujeto de certificado en el valor de atributo.

Cuando se almacenan en asientos **pkiUser**, valores de este atributo contienen trayectos de certificación que incluyen el certificado de entidad final. En este caso, la entidad final es el usuario cuyo asiento contiene este atributo. Los valores del atributo representan trayectos de certificación completos para certificados expedidos a este usuario.

11.3 Reglas de concordancia de directorios PKI

Esta Especificación de directorio define reglas de concordancia para utilizarlas con atributos del tipo **Certificate**, **CertificatePair**, **CertificateList**, **CertificatePolicy** y **SupportedAlgorithm**, respectivamente. Esta cláusula define asimismo reglas de concordancia para facilitar la selección de certificados o CRL con características específicas, a partir de atributos con valores múltiples que tiene múltiples certificados o CRL. La regla de concordancia de certificados mejorada permite realizar una concordancia más sofisticada con los certificados conservados en asientos de directorio.

11.3.1 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.

11.3.2 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,
   subject                   [11]  Name                      OPTIONAL,
   nameConstraints           [12]  NameConstraintsSyntax     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 SIZE (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:

serialNumber concuerda si el valor de este componente en el valor de atributo equivale al del valor presentado;

issuer concuerda si el valor de este componente en el valor de atributo equivale al del valor presentado;

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;

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 la 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;

certificateValid concuerda si el valor presentado está dentro del periodo de validez del valor de atributo almacenado;

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;

subjectPublicKeyAlgID concuerda si es igual al componente **algorithm** del **algorithmIdentifier** del componente **subjectPublicKeyInformation** del valor de atributo almacenado;

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;

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;

policy concuerda si al menos un miembro del **CertPolicySet** presentado aparece en la extensión de políticas de certificados en el valor de atributo almacenado o si el certificado presentado o el almacenado contienen el valor especial **cualquier política** en el componente de **políticas**. No hay concordancia si no hay extensión de políticas de certificado en el valor del atributo almacenado;

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.

subject concuerda si el valor de este componente en el valor de atributo es igual que en el valor presentado**.**

nameConstrains concuerda si los nombres de sujeto en el valor del atributo almacenado se encuentran dentro del espacio de nombres dado por el valor del componente de subárboles permitidos del valor presentado y no se encuentran en el espacio de nombre dado por el valor del componente subárboles excluidos del valor presentado.

11.3.3 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 {
    issuedToThisCAAssertion     [0] CertificateExactAssertion OPTIONAL,
    issuedByThisCAAssertion     [1] CertificateExactAssertion OPTIONAL }
    ( WITH COMPONENTS               {..., issuedToThisCAAssertion PRESENT} |
      WITH COMPONENTS               {..., issuedByThisCAAssertion PRESENT} )
```

Esta regla de concordancia devuelve VERDADERO si los componentes que están presentes en los componentes **issuedToThisCAAssertion** e **issuedByThisCAAssertion** del valor presentado concuerdan con los componentes correspondientes de los componentes **issuedToThisCA** e **issuedByThisCA**, respectivamente, en el valor de atributo almacenado.

11.3.4 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 **issuedToThisCA** o **issuedByThisCA** del par.

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

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

Esta regla de concordancia devuelve VERDADERO si todos los componentes que están presentes en los componentes **issuedToThisCAAssertion** e **issuedByThisCAAssertion** del valor presentado concuerdan con los correspondientes componentes de los componentes **issuedToThisCA** e **issuedByThisCA**, respectivamente, en el valor de atributo almacenado.

11.3.5 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, deberá concordar por lo menos con una forma de nombre.

11.3.6 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,
    authorityKeyIdentifier  [3]     AuthorityKeyIdentifier  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; también hay concordancia si el valor de atributo almacenado no contiene **reasonFlags** en la extensión de punto de

distribución expedidor, o si el valor de atributo almacenado no contiene ninguna extensión de punto de distribución expedidor;

NOTA – Aunque una CRL concuerde en un valor particular de **reasonFlags**, la CRL puede no contener notificaciones de revocación con dicho código de motivo.

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**;

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;

authorityKeyIdentifier concuerda si el valor de este componente en el valor de atributo almacenado es igual al del valor presentado; no concuerda si el valor de atributo almacenado no contiene la 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**.**

11.3.7 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 **SupportedAlgorithms**.

```
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.

11.3.8 Concordancia de políticas

La regla de concordancia de políticas compara la equivalencia de un valor presentado con un valor de atributo del tipo **CertificatePolicy** o con un valor de atributo del tipo **privPolicy**.

```
policyMatch MATCHING-RULE     ::= {
    SYNTAX     PolicyID
    ID         id-mr-policyMatch }
```

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

11.3.9 Concordancia de trayectos PKI

La regla de concordancia **pkiPathMatch** compara la equivalencia de un valor presentado con un valor de atributo del tipo **pkiPath**. Un sistema que utiliza certificados puede usar esta regla de concordancia para seleccionar un trayecto que empiece con un certificado expedido por una CA en la que confía y que termine con un certificado expedido por el sujeto especificado.

```
pkiPathMatch MATCHING-RULE  ::= {
    SYNTAX     PkiPathMatchSyntax
    ID         id-mr-pkiPathMatch }

PkiPathMatchSyntax      ::=  SEQUENCE {
    firstIssuer        Name,
    lastSubject        Name }
```

Esta regla de concordancia devuelve VERDADERO si el valor presentado en el componente **firstIssuer** concuerda con los elementos correspondientes del campo **issuer** del primer certificado en la **SEQUENCE** en el valor almacenado y el valor presentado en el componente **lastSubject** concuerda con los elementos correspondientes del campo de sujeto del último certificado en la **SEQUENCE** en el valor almacenado. Esta regla de concordancia devuelve FALSO si fracasa cualquiera de las concordancias.

11.3.10 Concordancia de certificados mejorada

La regla de concordancia de certificados mejorada permite comparar un valor presentado con un valor de atributo de tipo **Certificate**. La regla selecciona uno o varios certificados basándose en diversas características.

```
enhancedCertificateMatch MATCHING-RULE  ::= {
    SYNTAX     EnhancedCertificateAssertion
    ID         id-mr-enhancedCertificateMatch }
```

```
EnhancedCertificateAssertion  ::= 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]  AltName                    OPTIONAL,
   policy                  [9]  CertPolicySet              OPTIONAL,
   pathToName              [10] GeneralNames               OPTIONAL,
   subject                 [11] Name                       OPTIONAL,
   nameConstraints         [12] NameConstraintsSyntax      OPTIONAL
}

(ALL EXCEPT ({ -- none; at least one component shall be present -- }))

AltName  ::= SEQUENCE {
  altnameType        AltNameType,
  altNameValue       GeneralName OPTIONAL }
```

La operación de búsqueda en el directorio permite combinar diversos valores de **EnhancedCertificateAssertion** en especificaciones de filtro, incluyendo la lógica y/o. Esta regla de concordancia devuelve VERDADERO si todos los componentes presentes en el valor presentado concuerdan con los componentes correspondientes del valor de atributo, de la siguiente manera:

La concordancia de los componentes **serialNumber**; **issuer**; **subjectKeyIdentifier**; **authorityKeyIdentifier**; **certificateValid**, **privateKeyValid**, **policy**, **subject** y **nameConstraints** es la definida para los mismos componentes en la regla de concordancia **certificateMatch**.

El componente **subjectAltName** contiene un tipo **altNameType** y campos facultativos **altNameValue**. Si **altNameValue** está presente, el valor tendrá la misma forma de nombre indicada en **altNameType**.

subjectAltName concuerda si al menos una de las siguientes condiciones es verdadera:

- El valor presentado contiene solo el componente **altNameType** y el valor del atributo almacenado contiene la extensión del nombre alternativo del sujeto con un componente **AltNames** del mismo tipo indicado en el valor presentado.
- El valor presentado contiene ambos componentes **altNameType** y **altNameValue** y el valor del atributo almacenado contiene la extensión del nombre alternativo del sujeto con un componente **AltNames** del mismo tipo y valor indicados en el valor presentado.

subjectAltName no concuerda si al menos una de las siguientes condiciones es verdadera:

- El valor del atributo almacenado no contiene la extensión del nombre alternativo del sujeto.
- El valor del atributo almacenado contiene la extensión del nombre alternativo del sujeto pero el componente **AltNames** no incluye el mismo tipo identificado en el valor presentado.
- El valor presentado contiene ambos componentes **altNameType** y **altNameValue** y el valor del atributo almacenado contiene la extensión del nombre alternativo del sujeto con un componente **AltNames** del mismo tipo indicado en el valor presentado, pero el valor almacenado no contiene el mismo valor de ese tipo como en el valor presentado.

No podrá definirse la concordancia **subjectAltName** si el valor presentado contiene los componentes **altNameType** y **altNameValue** y el valor del atributo almacenado contiene la extensión del nombre alternativo del sujeto con un componente **AltNames** del mismo tipo indicado en el valor presentado, pero el tipo es tal que el directorio no puede comparar valores para poder determinar una concordancia. Ello puede deberse a que la forma de nombre no es adecuada para la concordancia o a que el directorio es incapaz de realizar las comparaciones correspondientes.

pathToName concuerda a menos que el certificado tenga una extensión de constricciones de nombre que inhiba la construcción de un trayecto de certificación a cualquiera de los valores de nombre presentados. Por ejemplo, si se trata de recuperar certificados que formen un trayecto a un certificado de usuario que tiene un valor de sujeto "dc=com; dc=corporate; cn=john.smith", puede resultar útil incluir una aserción en la operación de búsqueda que contenga este DN en el componente **pathToName**. Un certificado almacenado que contenga una extensión de limitaciones de nombre que excluya todo el subárbol por debajo de la base "dc=com; dc=company A" fracasaría en la validación del trayecto de certificación a ese certificado de usuario y por consiguiente, no sería un valor concordante para esta aserción de muestra.

SECCIÓN 3 – MARCO DE CERTIFICADOS DE ATRIBUTO

El marco de certificados de atributo que se define en esta sección proporciona las bases sobre las cuales se pueden construir infraestructuras de gestión de privilegios (PMI). Estas infraestructuras pueden soportar aplicaciones tales como el control de acceso.

La vinculación de un privilegio a una entidad la proporciona una autoridad mediante una estructura de datos firmados digitalmente denominada un certificado de atributo o mediante un certificado de clave pública que contiene una extensión definida explícitamente para estos fines. Se define el formato de los certificados de atributo, y se incluyen un mecanismo de extensibilidad y un conjunto de extensiones de certificado específicas. Se puede o no necesitar la revocación de certificados de atributo. Por ejemplo, en algunos entornos, los periodos de validez de certificado de atributo pueden ser muy cortos (por ejemplo minutos), desechando la necesidad de un esquema de revocación. Si, por alguna razón, una autoridad revoca un certificado de atributo expedido con anterioridad, los usuarios deben ser capaces de saber que se ha producido una revocación de forma que no utilicen un certificado en el que no se puede confiar. Las listas de revocación constituyen un esquema que puede utilizarse para notificar revocaciones a los usuarios. El formato de las listas de revocación se define en la sección 2 de esta Especificación, incluidos un mecanismo de extensibilidad y un conjunto de extensiones de lista de revocación. También se definen extensiones adicionales. En el caso de certificados y de listas de revocación, otros organismos también pueden definir extensiones adicionales que sean útiles a sus entornos específicos.

Un sistema que utiliza certificados de atributo, precisa validar un certificado antes de utilizar dicho certificado para una aplicación. Por ello también se definen los procedimientos para realizar dicha validación y se incluyen la verificación de integridad del propio certificado, su estado de revocación y su validez en relación con la utilización pertinente.

Este marco incluye algunos elementos facultativos que resultan adecuados únicamente en algunos entornos. Aunque los modelos se definen como completos, este marco se puede utilizar en entornos en los que no se utilizan todos los componentes de los modelos definidos. Por ejemplo, hay entornos en los que no se requiere la revocación de certificados de atributo. La delegación de privilegios y la utilización de cometidos son también aspectos de este marco que no se pueden aplicar de forma universal. Sin embargo, se incluyen en esta Especificación de forma que se pueda también soportar aquellos entornos que tengan requisitos para ello.

El directorio utiliza certificados de atributo para proporcionar control de accesos basado en reglas a la información de directorio.

12 Certificados de atributo

Los certificados de clave pública tienen el propósito fundamental de proporcionar un servicio de identidad sobre el que se pueden construir otros servicios de seguridad, tales como integridad de datos, autenticación de entidades, confidencialidad y autorización. En esta Especificación se proporcionan dos mecanismos distintos para vincular un atributo de privilegios a un titular.

Los certificados de clave pública, utilizados en combinación con el servicio de autenticación de entidades, puede suministrar directamente un servicio de autorización, si los privilegios están asociados con el sujeto mediante las prácticas de la CA expedidora. Los certificados de clave pública pueden contener una extensión **subjectDirectoryAttributes** que incluya privilegios asociados con el sujeto del certificado de clave pública. Este mecanismo es adecuado en situaciones en las que la autoridad que expide el certificado de clave pública (CA) es también la autoridad que delega el privilegio (AA) y el periodo de validez del privilegio corresponde al periodo de validez del certificado de clave pública. Las entidades finales no pueden actuar como AA. Si se incluye cualquiera de las extensiones definidas en la cláusula 15 de esta Especificación en un certificado de clave pública, dichas extensiones aplican por igual a todos los privilegios asignados en la extensión **subjectDirectoryAttributes** de ese certificado de clave pública.

En el caso más general, los privilegios de entidad tendrán duraciones que no concuerdan con el periodo de validez de un certificado de clave pública. Los privilegios tendrán a menudo una duración mucho más corta. La autoridad que asigna el privilegio será con frecuencia distinta de la autoridad que envía a esa misma entidad un certificado de clave pública y diferentes privilegios pueden ser asignados por diferentes autoridades de atributo (AA). Los privilegios también pueden ser asignados basándose en un contexto temporal y el aspecto 'poner/quitar' de los privilegios puede ser asíncrono con la duración del certificado de clave pública y/o asíncrono con privilegios de entidad expedidos desde una AA diferente. La utilización de certificados de atributo, expedidos por una AA proporciona una infraestructura de gestión de privilegios (PMI) flexible que se puede establecer y gestionar de forma independiente desde una PKI. Al mismo tiempo, existe una relación entre ambas siempre que se utilice la PKI para autenticar identidades de expedidores y titulares en certificados de atributo.

12.1 Estructura de certificado de atributo

Un certificado de atributo es una estructura diferenciada de un certificado de clave pública de sujeto. Un sujeto puede tener múltiples certificados de atributo asociados con cada uno de sus certificados de clave pública. No se requiere que la misma autoridad cree el certificado de clave pública y el certificado o certificados de atributo para un usuario; de hecho, la separación de tareas aconsejará a menudo actuar de otra forma. En entornos en los que diferentes autoridades tienen la responsabilidad de expedir certificados de clave pública y de atributo, el certificado o certificados de clave pública expedidos por una autoridad de certificación (CA) y el certificado o certificados de atributo expedidos por una autoridad de atributo (AA) se firmarán utilizando claves de firma privadas diferentes. En entornos en los que una única entidad es tanto la CA, que expide certificados de clave pública, como la AA, que expide certificados de atributo, se recomienda encarecidamente que se utilice para firmar certificados de atributo una clave diferente de la clave utilizada para firmar certificados de clave pública. Los intercambios entre la autoridad expedidora y la entidad que recibe un certificado están fuera del ámbito de esta Especificación.

El certificado de atributo se define como sigue:

```
AttributeCertificate ::= SIGNED {AttributeCertificateInfo}

AttributeCertificateInfo ::= SEQUENCE
    {
    version                 AttCertVersion, -- version is v2
    holder                  Holder,
    issuer                  AttCertIssuer,
    signature               AlgorithmIdentifier,
    serialNumber            CertificateSerialNumber,
    attrCertValidityPeriod  AttCertValidityPeriod,
    attributes              SEQUENCE OF Attribute,
    issuerUniqueID          UniqueIdentifier OPTIONAL,
    extensions              Extensions    OPTIONAL
    }

AttCertVersion ::= INTEGER { v2(1) }

Holder      ::=   SEQUENCE
              {
              baseCertificateID     [0] IssuerSerial        OPTIONAL,
                   -- the issuer and serial number of the holder's Public Key Certificate
              entityName            [1] GeneralNames        OPTIONAL,
                   -- the name of the entity or role
              objectDigestInfo      [2] ObjectDigestInfo    OPTIONAL
                   -- used to directly authenticate the holder, e.g., an executable
                   -- at least one of baseCertificateID, entityName or objectDigestInfo shall be present --}

ObjectDigestInfo    ::= SEQUENCE {
    digestedObjectType  ENUMERATED {
       publicKey            (0),
       publicKeyCert        (1),
       otherObjectTypes     (2) },
    otherObjectTypeID       OBJECT IDENTIFIER  OPTIONAL,
    digestAlgorithm         AlgorithmIdentifier,
    objectDigest            BIT STRING }

AttCertIssuer ::=     [0]   SEQUENCE {
issuerName                  GeneralNames OPTIONAL,
baseCertificateID       [0] IssuerSerial OPTIONAL,
objectDigestInfo        [1] ObjectDigestInfo OPTIONAL }
-- At least one component shall be present
    ( WITH COMPONENTS { ..., issuerName PRESENT } |
    WITH COMPONENTS { ..., baseCertificateID PRESENT } |
    WITH COMPONENTS { ..., objectDigestInfo PRESENT } )

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

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

El número **version** establece la diferencia entre diferentes versiones del certificado de atributo. Para los certificados de atributo expedidos de acuerdo con la sintaxis en esta Especificación **version** deberá ser **v2**.

El campo **holder** encamina la identidad del titular de certificado de atributo.

El componente **baseCertificateID**, si está presente, identifica a un determinado certificado de clave pública que debe utilizarse para autenticar la identidad de este titular cuando se establecen privilegios con este certificado de atributo.

El componente **entityName**, si está presente, identifica a uno o más nombres del titular. Si **entityName** es el único componente presente en **holder**, se puede utilizar cualquier certificado de clave pública que tenga uno de estos nombres como su sujeto para autenticar la identidad de ese titular cuando se establezcan privilegios con este certificado de atributo. Si están presentes **baseCertificateID** y **entityName**, sólo se puede utilizar el certificado especificado por **baseCertificateID**. En este caso, se incluye **entityName** únicamente como útil para ayudar al verificador de privilegios a localizar el certificado de clave pública identificado.

NOTA 1 – Al utilizar únicamente **GeneralNames** para identificar al titular, existe el riesgo de que éste indique únicamente un nombre para el titular. Esto resulta normalmente insuficiente para permitir la autenticación de la entidad del titular con el fin de expedir privilegios a dicho titular. La utilización del nombre y del número de serie de expedidor de un certificado de clave pública específico, sin embargo, permite al usuario de certificados de atributo utilizar el proceso de autenticación realizado por la CA cuando expide ese certificado de clave pública en particular. Asimismo, algunas de las posibilidades de **GeneralNames** (por ejemplo, **IPAddress**) resultan inapropiadas para su utilización en la denominación de un titular de certificado de atributo, en particular cuando el titular es un cometido y no una entidad individual. Otro problema con la utilización de **GeneralNames** como identificador para un titular es que muchas formas de nombre en ese constructivo no tienen autoridades de registro estrictas o procesos para la asignación de nombres.

El componente **objectDigestInfo**, si está presente, se utiliza directamente para autenticar la identidad de un titular, incluido un titular ejecutable, (por ejemplo, un applet). El titular se autentica comparando un extracto de la información correspondiente, creada por el verificador de privilegios con el mismo algoritmo identificado en **objectDigestInfo**, con el contenido de **objectDigest**. Si ambos son idénticos, el titular se autentica para establecer privilegios con este certificado de atributos.

- **publicKey** se indicará cuando se incluya un troceo de una clave pública de entidad. El troceo de una clave pública puede no identificar de forma unívoca un certificado (es decir, puede aparecer el mismo valor de clave en múltiples certificados). Para unir un certificado de atributo a una clave pública se calcula el troceo en la representación de esa clave pública que estaría presente en un certificado de clave pública. De forma específica, la entrada del algoritmo de troceo será la codificación DER de una representación **SubjectPublicKeyInfo** de la clave. Cabe destacar que esto también incluye el algoritmo **AlgorithmIdentifier** como **BIT STRING**. Si el valor de clave pública utilizado como entrada para la función de troceo se ha extraído de un certificado de clave pública, entonces, es posible (por ejemplo, si se heredan parámetros para el algoritmo de firma digital) que no sea una entrada suficiente para el troceo. La entrada correcta para el TROCEO en este contexto incluirá el valor de los parámetros heredados y por lo tanto puede diferir del valor de **SubjectPublicKeyInfo** presente en el certificado de clave pública.
- **publicKeyCert** indicará que cuando se trocea un certificado de clave pública, el troceo se realiza sobre toda la codificación DER del certificado de clave pública, incluidos los bits de firma.
- **otherObjectTypes** se incluirá cuando se trocean objetos distintos de las claves públicas o de los certificados de clave pública (por ejemplo, objetos de soporte lógico). Se puede suministrar como opción la identidad del tipo de objeto. La parte del objeto que ha de trocearse se puede determinar mediante el identificador establecido explícitamente para el tipo o, si no se suministra el identificador, mediante el contexto en el cual se utiliza el objeto.

El campo **issuer** transporta la identidad de la AA que expidió el certificado.

- El componente **issuerName**, si está presente, identifica uno o más nombres para el expedidor.
- El componente **baseCertificateID**, si está presente, identifica al expedidor haciendo referencia a un certificado de clave pública específico para el cual este expedidor es el sujeto.
- El componente **objectDigestInfo**, si está presente, identifica al expedidor, proporcionando un troceo de la información de identificación del expedidor.

signature identifica al algoritmo criptográfico utilizado para firmar digitalmente el certificado de atributo.

serialNumber es el número de serie que identifica de forma unívoca al certificado de atributo en el ámbito de su expedidor.

El campo **attrCertValidityPeriod** transmite el periodo de tiempo durante el cual el certificado de atributo se considera válido, expresado en formato **GeneralizedTime**.

El campo **attributes** contiene los atributos asociados con el titular que se están certificando (por ejemplo, los privilegios).

NOTA 2 – En el caso de certificados de atributos de descriptor de atributo, esta secuencia de atributos puede estar vacía.

issuerUniqueID se puede utilizar para identificar al expedidor del certificado de atributo en ejemplares en los que el componente de expedidor no es suficiente.

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

Si aparecen elementos desconocidos en la extensión, y ésta no ha sido marcada como crítica, esos elementos desconocidos serán ignorados de acuerdo con las reglas de extensibilidad documentadas en 12.2.2 de la Rec. UIT-T X.519 | ISO/CEI 9594-5.

El marco para los certificados de atributo descritos en esta sección se centra fundamentalmente en el modelo en el que se sitúa el privilegio en certificados de atributo. Sin embargo, como se ha mencionado anteriormente, las extensiones de certificados definidas en esta sección pueden también situarse en un certificado de clave pública que utilice la extensión **subjectDirectoryAttributes**.

12.2 Trayectos de certificados de atributo

De la misma forma que con los certificados de clave pública, puede existir un requisito para encaminar un trayecto de certificados de atributo (por ejemplo, en un protocolo de aplicación para establecer privilegios). El tipo de datos ASN.1 siguiente se puede utilizar para representar un trayecto de certificados de atributo.

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

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

13 Relación entre autoridad de atributo, SOA y autoridad de certificación

La autoridad de atributo (AA) y la autoridad de certificación (CA) son lógicamente (y en muchos casos físicamente) totalmente independientes. La creación y mantenimiento de una "identidad" puede (y a menudo debe) ser separada de la PMI. Por tanto, puede existir y estar en explotación la PKI completa, incluidas las CA, antes del establecimiento de la PMI. La CA, aunque es la fuente de autoridad para identidades en su dominio, no es automáticamente la fuente de autoridad para privilegios. La CA, por lo tanto, no será forzosamente una AA y, lógicamente, no será necesariamente responsable de tomar la decisión sobre cuales otras entidades serán capaces de funcionar como AA (por ejemplo, incluyendo este tipo de designación en sus certificados de identidad).

La fuente de autoridad (SOA, *source of authority*) es una entidad en la que confía un verificador de privilegios, siendo la entidad con la responsabilidad última para asignar un conjunto de privilegios. Un recurso puede limitar la autoridad de la SOA confiando en ciertas SOA para funciones específicas (por ejemplo, una para leer privilegios y otra diferente para escribir privilegios). Una SOA es en sí misma una AA, puesto que expide certificados a otras entidades en los que se asignan privilegios a esas entidades. Una SOA es análoga a una 'CA raíz' o 'ancla de confianza' en la PKI, puesto que un verificador de privilegios confía en certificados firmados por la SOA. En algunos entornos existe la necesidad de que las CA tengan un control estrecho sobre las entidades que pueden actuar como SOA. Este marco proporciona un mecanismo para soportar dicho requisito. En otros entornos no se precisa dicho control y los mecanismos para determinar las entidades que pueden actuar como SOA puede estar fuera del ámbito de esta Especificación.

Este marco es flexible y puede satisfacer los requisitos de muchos tipos de entornos.

a) En muchos entornos se asignarán todos los privilegios directamente a entidades individuales mediante una única AA, que será la SOA.

b) Otros entornos pueden requerir soporte para caracterizar cometidos facultativos, en los que se expiden certificados a entes que les asignan diversos cometidos. Los privilegios asociados con el cometido se asignan implícitamente a dichos entes. Los privilegios de cometido pueden ser asignados en un certificado de atributo expedido al propio cometido o mediante otros medios (por ejemplo, configurados localmente).

c) Otra característica facultativa de este marco es el soporte de la delegación de privilegios. Si se realiza la delegación, la SOA asigna privilegios a una entidad que también tiene permiso para actuar como una AA y para delegar el privilegio. La delegación puede continuar a través de varias AA intermedias hasta que se asigna finalmente a una entidad final que ya no puede delegar dicho privilegio. Las AA intermedias pueden o no ser también capaces de actuar como asertores de privilegios para los privilegios que delegan.

d) En algunos entornos, la misma entidad física puede estar actuando como una AA y como una CA. Este cometido lógico dual para la misma entidad física siempre se produce cuando el privilegio se transmite en una extensión **subjectDirectoryAttributes** de un certificado de clave pública. En otros entornos entidades físicas diferenciadas actúan como CA y como AA. En este último caso, el privilegio se asigna utilizando certificados de atributo en lugar de certificados de clave pública.

Cuando los certificados de atributos se dirigen a certificados de clave pública para sus expedidores y sus titulares, la PKI se utiliza para autenticar titulares (asertores de privilegios) y verificar las firmas digitales de los expedidores.

En esta Especificación se describen dos modelos de delegación. En el primer modelo de delegación el delegador de privilegios es una AA que puede expedir certificados delegando ese privilegio a otros. El segundo modelo permite un servicio de delegación (DS, *delegation service*) independiente donde la entidad expide certificados en nombre de otra AA (que puede o no expedir ella misma los AC). Este DS no puede actuar como detentador de ese privilegio. El modelo DS es especialmente pertinente en entornos que desean mantener cierta gestión central del conjunto de privilegios delegados en su dominio. Por ejemplo, un conjunto de uno o varios servidores DS que realizan delegación, en lugar de ser titulares de privilegios individuales, permite determinar todo el conjunto de privilegios delegado en un entorno por determinar desde un recurso centralizado y permite que las decisiones de política y gestión puedan modificarse convenientemente. En los servidores DS son posibles dos modelos de despliegue distintos. En un modelo, un privilegio es asignado por una SOA a los titulares de privilegios y esos titulares están autorizados a delegar ese privilegio a otros. Sin embargo, más que expedir los certificados de atributos que delegan los privilegios en sí, el titular del privilegio solicita al DS que delegue ese privilegio en su nombre. El propio DS no posee ese privilegio y por tanto no puede actuar como detentador del mismo, pero sí está autorizado por la SOA a expedir certificados de atributos en nombre de otros titulares de privilegios. El segundo modelo de despliegue es similar al primero con la siguiente excepción. El DS es en realidad un titular al que se ha asignado el privilegio a delegar, pero no está autorizado a actuar como detentador del privilegio, sino sólo como delegador. En este caso, la extensión noAssertion debe fijarse en el AC expedido al DS por la SOA. El DS se denomina expedidor indirecto.

En ambos modelos de despliegue la SOA expide atributos/privilegios a las AA subordinadas. Las AA piden luego al DS que expida un subconjunto de estos atributos de privilegios a otros titulares. En el segundo modelo de despliegue el DS puede comprobar que una AA está delegando dentro del ámbito general establecido por la SOA, mientras que en el primer modelo el DS no puede hacer dicha comprobación y la parte confiante tendrá que comprobar que la delegación se realizó correctamente.

13.1 Privilegios en certificados de atributo

Las entidades puede adquirir privilegios de dos formas:

- Una AA puede asignar unilateralmente privilegios a una entidad mediante la creación de un certificado de atributo (posiblemente sólo por su propia iniciativa, o a petición de alguna tercera parte). Este certificado se puede almacenar en un depósito accesible a todos y puede ser procesado posteriormente por uno o más verificadores de privilegios para tomar una decisión de autorización. Todo esto no puede ocurrir sin el conocimiento o la actuación explícita de la entidad.
- Como alternativa, una entidad puede solicitar un privilegio a alguna AA. Una vez creado, este certificado se puede devolver (sólo) a la entidad solicitante, que lo proporciona explícitamente cuando solicita acceso a algún recurso protegido.

Cabe destacar que en ambos procedimientos la AA necesita mantener la diligencia debida para asegurar que se asigna realmente este privilegio a la entidad. Esto puede implicar algún mecanismo fuera de banda, análogo a la certificación por una CA de una vinculación de pares identidad/clave.

El certificado de atributo basado en la PMI es adecuado en entornos en los que es cierta cualquiera de las afirmaciones siguientes:

- una entidad diferente es responsable de asignar privilegios particulares a un titular antes que de expedir certificados de clave pública al mismo sujeto;
- existen algunos atributos de privilegio que se deben asignar a un titular, desde diversas autoridades;
- la duración de un privilegio difiere de la de la validez del certificado de clave pública del titular (generalmente la duración de privilegios es mucho mas corta); o
- el privilegio es válido únicamente durante ciertos intervalos de tiempo que son asíncronos con el de la validez de la clave pública de usuario o la validez de otros privilegios.

13.2 Privilegios en certificados de clave pública

En algunos entornos, los privilegios están asociados con el sujeto mediante las actuaciones de una CA. Este tipo de privilegio se puede incluir directamente en certificados de clave pública (reutilizando así gran parte de la infraestructura

ya establecida), en lugar de expedir certificados de atributo. En estos casos, el privilegio se incluye en la extensión **subjectDirectoryAttributes** del certificado de clave pública.

Este mecanismo es adecuado en entornos en los que es cierta una o más de las afirmaciones siguientes:

- la misma entidad física está actuando como una CA y como una AA;
- la duración del privilegio es conforme a la de la clave pública incluida en el certificado;
- no se permite delegar el privilegio; o
- se permite su delegación pero, para cada delegación, todos los privilegios en el certificado (en la extensión **subjectDirectoryAttributes**) tienen los mismos parámetros de delegación y todas las extensiones relativas a la delegación se aplican por igual a todos los privilegios en el certificado.

14 Modelos de PMI

14.1 Modelo general

El modelo general de gestión de privilegios está constituido por tres entidades. El objeto, el asertor de privilegios y el verificador de privilegios.

El objeto puede ser un recurso al que se está protegiendo, por ejemplo en una aplicación de control de acceso. El recurso que se está protegiendo se considera entonces como el objeto. Este tipo de objeto tiene métodos que se pueden invocar (por ejemplo, el objeto puede ser un cortafuegos que tiene un método de objeto "Permitir entrada", o el objeto puede ser un fichero en un sistema de ficheros que tenga los métodos de objeto leer, escribir, ejecutar). Otro tipo de objetos en este modelo puede ser un objeto que se firmó en una aplicación de no repudio.

El asertor de privilegios es una entidad que contiene un privilegio determinado y afirma su privilegio para un determinado contexto de utilización.

El verificador de privilegios es la entidad que toma la determinación de si son o no suficientes los privilegios afirmados para el contexto de utilización dado.

La determinación éxito/fracaso tomada por el verificador de privilegios depende de cuatro cosas:

- el privilegio del asertor de privilegios;
- la política de privilegios vigente;
- las variables de entorno actuales, si viene al caso; y
- la sensibilidad del método de objeto, si viene al caso.

El privilegio de un titular de privilegio refleja el grado de confianza otorgado a ese titular, por el expedidor del certificado, en que el titular del privilegio cumplirá aquellos aspectos de política que no están impuestos por medios técnicos. Este privilegio se encapsula en el certificado o certificados de atributo del titular de privilegio (o en la **subjectDirectoryAttributes** de su certificado de clave pública), que pueden presentarse al verificador de privilegios en la petición de invocación o puede distribuirse mediante algún otro medio, como a través del directorio. La codificación de privilegios se realiza mediante la utilización del constructor **Attribute**, que incluye un **AttributeType** y un **SET OF AttributeValue**. Algunos tipos de atributos utilizados para especificar privilegios pueden tener una sintaxis muy simple, como un único **INTEGER** o un **OCTET STRING**. Otros pueden tener sintaxis más complejas. En el anexo D se proporciona un ejemplo.

La política de privilegios especifica el grado de privilegio que se considera suficiente para una sensibilidad dada del método de objeto o del contexto de utilización. La política de privilegios necesita estar protegida para su integridad y autenticidad. Existen algunas posibilidades para encaminar políticas. En un extremo existe la idea de que la política en realidad no se transmite en absoluto, pero que simplemente se define y se mantiene localmente en el entorno del verificador de privilegios. En el otro extremo está la idea de que algunas políticas son "universales" y deberían enviarse a, y con el conocimiento de, todas las entidades del sistema. Entre estos extremos existen muchas posibles variaciones. En esta Especificación se definen componentes de esquemas para almacenar información de políticas de privilegios en el directorio.

La política de privilegios especifica el umbral de aceptación de un determinado conjunto de privilegios. Es decir, define con precisión cuando un verificador de privilegios debe concluir que un conjunto de privilegios presentado es "suficiente" de forma que puede facilitar el acceso (al objeto, al recurso, a la aplicación solicitados, etc.) al asertor de privilegios.

En esta Especificación no está normalizada la sintaxis para la definición de la política de privilegios. El anexo D incluye algunos ejemplos de sintaxis que podrían utilizarse para estos fines. Sin embargo, sólo son ejemplos. Para ello se puede

utilizar cualquier sintaxis incluido texto claro. Independientemente de la sintaxis utilizada para definir la política de privilegios, cada ejemplar de privilegios tendrá que estar identificado de forma unívoca. Para esto se utilizan identificadores de objeto.

PrivilegePolicy ::= OBJECT IDENTIFIER

Las variables de entorno, si procede, capturan aquellos aspectos de política necesarios para la determinación éxito/fracaso (por ejemplo, hora del día o balance de cuenta corriente) que están disponibles mediante algún medio local del verificador de privilegios. La representación de variables de entorno es sólo un asunto local.

La sensibilidad del método de objeto, si procede, puede reflejar atributos del documento o petición que ha de procesarse, tales como el valor monetario de la transferencia de fondos que se pretende autorizar o la confidencialidad del contenido del documento. La sensibilidad del método de objeto puede estar codificada explícitamente en una etiqueta de seguridad asociada o en un certificado de atributo propiedad del método de objeto, o puede ser encapsulada implícitamente en la estructura y en el contenido del objeto de datos asociado. Puede codificarse en una de diversas formas diferentes. Por ejemplo, puede codificarse fuera del ámbito de la PMI en la etiqueta X.411 asociada con un documento, en los campos de un intercambio EDIFACT, o codificada en la aplicación del verificador del privilegios. De otra forma, se puede realizar dentro de la PMI, en un certificado de atributo asociado con un método de objeto. Para algunos contextos de utilización, no se utiliza ninguna sensibilidad del método de objeto.

No existe necesariamente una relación de vinculación entre un verificador de privilegios y una AA determinada. De la misma forma que los titulares de privilegios pueden tener certificados de atributo expedidos por muchas AA diferentes, los verificadores de privilegios pueden aceptar certificados expedidos por numerosas AA, que no necesitan estar relacionadas jerárquicamente unas con otras para garantizar el acceso a un determinado recurso.

El marco de certificado de atributos se puede utilizar para gestionar privilegios de diversos tipos y con otros fines. Los términos utilizados en esta Especificación, tales como asertor de privilegios, verificador de privilegios, etc. son independientes de la aplicación o de la utilización.

14.1.1 PMI en el contexto de control de acceso

Existe un marco normalizado para el control de acceso (Rec. UIT-T X.812 | ISO/CEI 10181-3) que define un conjunto de términos que son específicos a la aplicación de control de acceso. Aquí se proporciona una correspondencia de los términos genéricos utilizados en esta Especificación con los del marco de control de acceso, para esclarecer la relación entre este modelo y dicha Especificación.

El asertor de privilegios en esta Especificación representaría el papel de un 'iniciador' en el marco de control de acceso.

El verificador de privilegios de esta Especificación representaría un papel de una 'función de decisión de control de acceso (ADF)' en el marco de control de acceso.

El método de objeto en el que se está afirmando el privilegio en esta Especificación se correspondería con el 'objetivo' definido en el marco de control de acceso.

La variables ambientales en esta Especificación corresponderían con la 'información contextual' en el marco de control de acceso.

La política de privilegios tratada en esta Especificación podría incluir la 'política de control de acceso' y las "reglas de política de control de acceso" del marco de control de acceso.

Este modelo permite incluir una PMI sin discontinuidades en una red existente de recursos que se han de proteger. En particular, el que el verificador de objetos actúe como una pasarela hacia un método de objeto sensible, que garantiza o deniega solicitudes para invocación de un método de objeto, permite proteger el objeto con muy poco o ningún esfuerzo del propio objeto. El verificador de privilegios verifica todas las peticiones y sólo pasan a los métodos de objeto apropiados aquellas que están adecuadamente autorizadas.

14.1.2 PMI en un contexto de no repudio

Existe un marco normalizado para no repudio (Rec. UIT-T X.813 | ISO/CEI 10181-4) que define un conjunto de términos que son específicos al no repudio. Aquí se proporciona una correspondencia de los términos genéricos utilizados en esta Especificación con los del marco de no repudio, para esclarecer la relación entre este modelo y esa Especificación.

El asertor de privilegios en esta Especificación representaría el papel de un 'sujeto de evidencia' o un 'originador' en el marco de no repudio.

El verificador de privilegios de esta Especificación representaría del papel de 'usuario de evidencia' o de 'receptor' en el marco de no repudio.

El método de objeto en el que se está afirmando el privilegio en esta Especificación correspondería al 'objetivo' definido en el marco de no repudio.

La variables ambientales en esta Especificación se corresponderían con 'fecha y hora en la que se generó o verificó la evidencia' en el marco de no repudio.

La política de privilegios tratada en esta Especificación podría incluir 'política de seguridad de no repudio' en el marco de no repudio.

14.2 Modelo de control

El modelo de control ilustra como se ejerce el control en el acceso al método de objeto sensible. Hay cinco componentes del modelo: el asertor de privilegios, el verificador de privilegios, el método de objeto, la política de privilegios y las variables ambientales (véase la figura 3). El asertor de privilegios tiene privilegio; el método de objeto tiene sensibilidad. Las técnicas descritas aquí permiten al verificador de privilegios controlar el acceso al método de objeto mediante el asertor de privilegios, de conformidad con la política de privilegios. Tanto el privilegio como la sensibilidad pueden ser parámetros con múltiples valores.

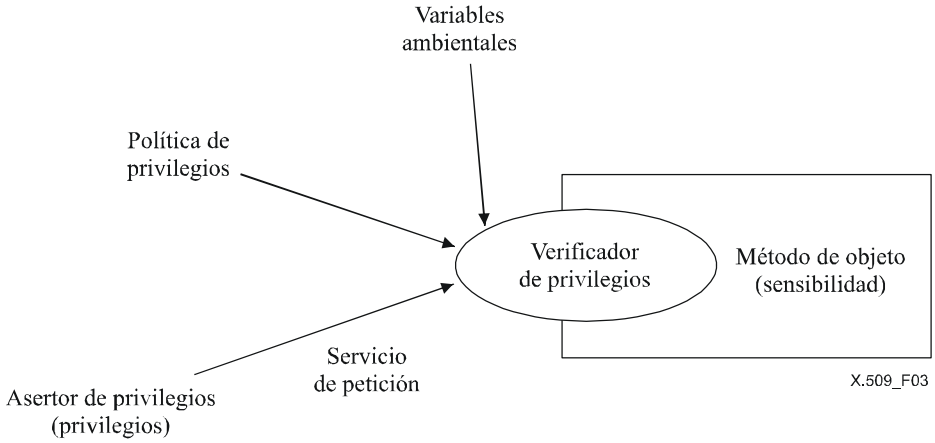


Figura 3 – Modelo de control

El asertor de privilegios puede ser una entidad identificada por un certificado de clave pública o un objeto ejecutable identificado por su imagen de disco, etc.

14.3 Modelo de delegación

En algunos entornos puede existir la necesidad de delegar el privilegio, aunque esto es un aspecto facultativo del marco y no se requiere en todos los entornos. Hay cuatro componentes del modelo de delegación: el verificador de privilegios, la SOA, otras AA y el asertor de privilegios (véase la figura 4).

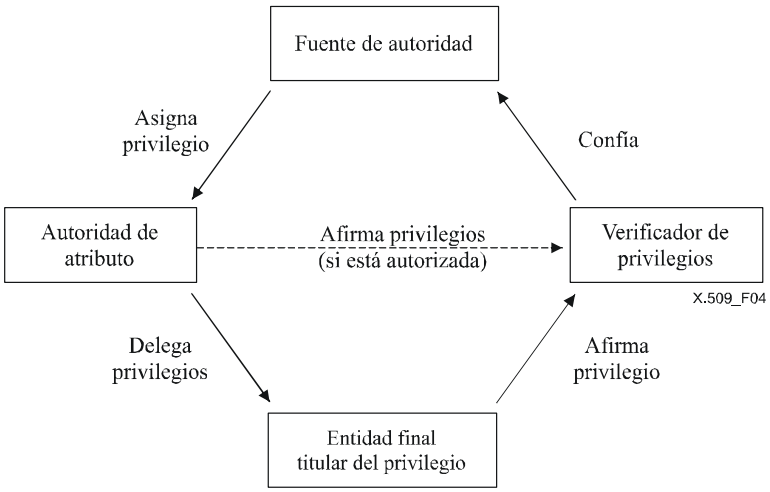


Figura 4 – Modelo de delegación

Como en entornos en los que no se utiliza delegación, la SOA es el expedidor inicial de certificados que asigna privilegios a los titulares de privilegios. Sin embargo, en este caso la SOA autoriza al titular del privilegio a actuar como una AA y a delegar posteriormente dicho privilegio a otras entidades mediante la expedición de certificados que contienen el mismo privilegio (o un subconjunto del mismo). La SOA puede imponer restricciones en la delegación que se puede realizar (por ejemplo, limitar la longitud el trayecto, limitar el espacio de nombre en el que se puede realizar la delegación). Cada una de estas AA intermediarias puede, en certificados que expiden a otros titulares de privilegios, autorizar que aquellos titulares que están actuando también como AA realicen delegaciones ulteriores. Una restricción universal en la delegación es que ninguna AA puede delegar más privilegios que los que tiene. Un delegador también puede restringir ulteriormente la capacidad de otras AA subsiguientes.

Cuando se utiliza delegación, el verificador de privilegios confía en la SOA para que delegue algunos o todos los privilegios a los titulares, algunos de los cuales pueden delegar ulteriormente algunos o todos de los privilegios a otros titulares.

El verificador de privilegios confía en la SOA como autoridad para un determinado conjunto de privilegios para el recurso. Si el certificado de asertor de privilegios no está expedido por la SOA, el verificador de privilegios tendrá que encontrar un trayecto de delegación de los certificados entre el del asertor de privilegios y alguno expedido por la SOA. La validación de dicho trayecto de delegación incluye la comprobación de que cada AA tiene privilegios suficientes y que estaba debidamente autorizada para delegar dichos privilegios.

En el caso en el que se envíen los privilegios mediante certificados de atributo, el trayecto de delegación está diferenciado del trayecto de validación de los certificados utilizados para validar los certificados de clave pública de las entidades implicadas en el proceso de delegación. Sin embargo, la calidad de autenticidad que tiene el proceso de validación de certificados de clave pública tendrá que ser proporcionado a la sensibilidad del método de objeto que se está protegiendo.

Un trayecto de delegación estará constituido totalmente por certificados de atributo o totalmente por certificados de clave pública. Un delegador que obtiene su privilegio en un certificado de atributo sólo puede delegar, si está autorizado, mediante la expedición de certificados de atributo subsiguientes. De forma similar, un delegador que obtiene su privilegio de un certificado de clave pública, si está autorizado, sólo puede delegar mediante la expedición de certificados de clave pública subsiguientes. Sólo las AA puede delegar privilegios. Las entidades finales no pueden.

14.4 Modelo de cometidos

Los cometidos proporcionan un medio para asignar privilegios indirectamente a entes individuales. A los entes individuales se expiden certificados de asignación de cometido para asignarles uno o más cometidos mediante el atributo de cometido contenido en el certificado. Se asignan privilegios específicos a un nombre de cometido mediante certificados de especificación de cometidos, en lugar de a titulares de privilegios individuales mediante certificados de atributo. Este nivel de indirección permite, por ejemplo, actualizar los privilegios asignados a un cometido, sin influir en los certificados que asignan cometidos a los entes individuales. Los certificados de asignación de cometidos pueden ser certificados de atributo o certificados de clave pública. Los certificados de especificación de cometidos pueden ser certificados de atributo, pero no certificados de clave pública. Si no se utilizan certificados de especificación de cometidos, la asignación de privilegios a un cometido se puede realizar por otros medios (por ejemplo, se puede configurar localmente en un verificador de privilegios).

Las siguientes afirmaciones son todas posibles:

- cualquier AA puede definir cualquier número de cometidos;
- AA diferentes pueden definir y administrar por separado el propio cometido y los miembros de un cometido;
- se puede delegar el hecho de ser miembro de un cometido, de la misma forma que cualquier otro privilegio; y
- a los cometidos y al hecho de ser miembros se les puede asignar cualquier duración adecuada.

Si el certificado de asignación de cometidos es un certificado de atributo, el atributo **role** está contenido en el componente **attributes** del certificado de atributo. Si el certificado de asignación de cometidos es un certificado de clave pública, el atributo **role** está contenido en la extensión **subjectDirectoryAttributes**. En este último caso, cualesquiera privilegios adicionales contenidos en el certificado de clave pública son privilegios que están asignados directamente al sujeto de certificado y no son privilegios asignados al cometido.

Así, un asertor de privilegios puede presentar un certificado de asignación de cometidos al verificador de privilegios demostrando únicamente que el asertor de privilegios tiene un cometido determinado (por ejemplo, "gestor" o "adjudicatario"). El verificador de privilegios puede conocer *a priori*, o puede tener que descubrir mediante algún otro medio, los privilegios asociados con el cometido que se está afirmando con el fin de tomar una decisión de autorización éxito/fracaso. El certificado de especificación de cometidos se puede utilizar para este fin.

Un verificador de privilegios necesita tener un conocimiento de los privilegios especificados para el cometido. La asignación de dichos privilegios a un cometido se puede realizar dentro de la PMI en un certificado de especificación de cometidos o fuera de la PMI (por ejemplo, configurado localmente). Si los privilegios de cometidos están afirmados en un certificado de especificación de cometidos, en esta Especificación se proporcionaran mecanismos para vincular dicho certificado con el certificado de asignación de cometidos pertinente para el asertor de privilegios. Ninguna otra entidad puede delegar un certificado de especificación de cometidos. El expedidor de un certificado de asignación de cometidos puede ser independiente del expedidor del certificado de especificación de cometidos y éstos se pueden administrar (caducado, revocado, etc.) de forma totalmente separada. El mismo certificado (certificado de atributo o certificado de clave pública) puede ser un certificado de asignación de cometidos así como puede contener la asignación de otros privilegios directamente al mismo ente individual. Sin embargo, un certificado de especificación de cometidos tendrá que ser un certificado diferenciado.

NOTA – La utilización de cometidos en un marco de autorización puede incrementar la complejidad del procesamiento del trayecto, puesto que este tipo de funcionalidad define fundamentalmente otro trayecto de delegación que necesita seguirse. El trayecto de delegación para el certificado de asignación de cometidos puede implicar diferentes AA y puede ser independiente de la AA que expidió el certificado de especificación de cometidos.

14.4.1 Atributo de cometido

La especificación de tipos de atributos de privilegio es generalmente un asunto específico de aplicación que se encuentra fuera del ámbito de esta Especificación. La única excepción la constituye un atributo definido aquí para la asignación de un titular a un cometido. La especificación de valores para el atributo de cometidos se encuentra fuera del ámbito de esta Especificación.

```
role   ATTRIBUTE  ::= {
       WITH SYNTAX          RoleSyntax
       ID                   id-at-role }

RoleSyntax ::= SEQUENCE {
roleAuthority       [0]     GeneralNames     OPTIONAL,
roleName            [1]     GeneralName }
```

Este atributo de privilegio se utilizaría para poblar el campo **attributes** de un certificado de asignación de cometidos. Si el certificado de asignación de cometidos es un certificado de clave pública, se utilizaría este atributo para poblar la extensión **subjectDirectoryAttributes** de dicho certificado de clave pública.

roleAuthority, si está presente, identifica la autoridad reconocida que es responsable de expedir el certificado de especificación de cometidos.

Si está presente **roleAuthority**, y un verificador de privilegios utiliza un certificado de especificación de cometidos para determinar los privilegios asignados a un cometido, por lo menos uno de los nombres en **roleAuthority** tendrá que estar presente en el campo **issuer** de ese certificado de especificación de cometidos. Si el verificador de privilegios utilizó medios distintos de un certificado de especificación de cometidos para determinar los privilegios asignados al cometido, los mecanismos para asegurar que dichos privilegios se asignaron mediante una autoridad denominada en este componente se encuentran fuera del ámbito de esta Especificación.

Si **roleAuthority** está ausente, la identidad de la autoridad responsable tendrá que determinarse mediante otros medios. La extensión **roleSpecCertIdentifier** en un certificado de asignación de cometidos es una forma de lograr esta vinculación, si se utilizara un certificado de especificación de cometidos para asignar privilegios al cometido.

El componente **roleName** identifica el cometido asignado al titular de un certificado de asignación de cometidos que contiene ese atributo. Si un verificador de privilegios utiliza un certificado de especificación de cometidos para determinar los privilegios asignados a ese cometido, este nombre de cometido tendrá que aparecer también en el campo **holder** del certificado de especificación de cometidos.

14.5 Atributo de información de privilegios XML

Por lo general, la especificación de privilegios es una cuestión específica de la aplicación que queda fuera del alcance de esta Especificación. Aunque este atributo no define información alguna de privilegios específicos, sí proporciona un atributo contenedor en el cual pueden transportarse privilegios codificados en XML en certificados de atributos.

```
xmlPrivilegeInfo    ATTRIBUTE  ::= {
  WITH SYNTAX       UTF8String -- contains XML-encoded privilege information
  ID                id-at-xMLPrivilegeInfo }
```

El esquema XML para el tipo de atributo de cometido puede definirse con ASN.1 o con XSD.

El XML contenido en **UTF8String** debe ser autoidentificador.

Lo siguiente representa un esquema ASN.1 en el que se define un tipo de atributo de cometido XML, seguido por una especificación XSD para el mismo tipo de atributo, y por un ejemplo de ejemplar XML. Éste representa un ejemplar para ambos ejemplares de esquema ASN.1 y XSD, y puede ser validado mediante cualquiera de las herramientas ASN.1 o XSD.

El esquema de ejemplo define un atributo de cometido con un ID, una autoridad de expedición y el nombre del cometido.

```
CERTIFICATE-ATTRIBUTE DEFINITIONS ::=
BEGIN
  Role ::= [UNCAPITALIZED] SEQUENCE {
     id            [ATTRIBUTE] XML-ID,
     authorities     SEQUENCE (1..MAX) OF
                    authority UTF8String,
     name           UTF8String }

  XML-ID  ::= UTF8String
END
```

El esquema XSD que sigue es una definición alternativa (exactamente equivalente):

```
<schema xmlns="http://www.w3.org/2000/08/XMLSchema">
 <element name="role">
  <attribute name="id" type="ID"/>
  <complexType>
   <sequence>
    <element name="authorities">
     <complexType>
      <sequence>
       <element name="authority" type="string" minOccurs="1" maxOccurs="*"/>
      </sequence>
     </complexType>
    </element>
    <element name="name" type="string"/>
   </sequence>
  </complexType>
 </element>
</schema>
```

Un ejemplo de ejemplar conforme con las definiciones de esquema anteriores, que sería un valor del tipo de atributo **xMLPrivilegeInfo,** podría ser:

```
<role id="123" xmlns="http://www.example.org/certificates/attribute">
<authorities>
<authority>Fictitious Organization</authority>
</authorities>
<name>manager</name>
</role>
```

15 Extensiones de certificados de gestión de privilegios

Se pueden incluir las extensión de certificados siguientes en certificados para la gestión de privilegios. Junto con la definición de las propias extensiones, se proporcionan también las reglas para los tipos de certificados en los que puede estar presente la extensión.

Con la excepción de la extensión de identificador SOA, cualquiera de las extensiones que se pueden incluir en un certificado de clave pública se incluirá únicamente si ese certificado de clave pública es uno de los que asigna privilegios a su sujeto (es decir, estará presente la extensión **subjectDirectoryAttributes**). Si cualquiera de dichas extensiones está presente en un certificado de clave pública, dicha extensión aplica a todos los privilegios presentes en la extensión **subjectDirectoryAttributes**.

Las listas de revocación utilizadas para publicar notificaciones de revocación para certificados de atributo (ACRL y AARL) pueden contener cualquier extensión CRL o de asiento de CRL, como se define en la sección 2 de esta Especificación para las CRL y las CARL.

Esta cláusula especifica las extensiones en los ámbitos siguientes:

a) *Gestión básica de privilegios*: Estas extensiones de certificado transmiten información importante para la afirmación de un privilegio.

b) *Revocación de privilegios*: Estas extensiones de certificado transmiten información relativa a la ubicación de la información de estado de revocación.

c) *Fuente de autoridad*: Estas extensiones de certificado están relacionadas con la fuente de asignación de privilegios en la que confía para un verificador para un recurso determinado.

d) *Cometidos*: Estas extensiones de certificado transmiten información relativa a la ubicación de certificados de especificación de cometidos relacionados.

e) *Delegación*: Estas extensiones de certificado permiten establecer constricciones en delegaciones subsiguientes de privilegios asignados.

15.1 Extensiones de gestión básica de privilegios

15.1.1 Requisitos

Los siguientes requisitos se relacionan con la gestión básica de privilegios:

a) Los expedidores tendrán que ser capaces de establecer constricciones en relación con el tiempo durante el cual un privilegio se puede afirmar.

b) Los expedidores tendrán que ser capaces de dirigir certificados de atributo a servidores/servicios específicos.

c) Puede ser necesario que los expedidores transmitan información para presentarla a los asertores de privilegios y/o verificadores de privilegios que utilizan el certificado.

d) Los expedidores pueden necesitar la capacidad de situar constricciones en las políticas de privilegios con las que se puede utilizar el privilegio asignado.

15.1.2 Campos de extensión de gestión básica de privilegios

Se definen los siguientes campos de extensión:

a) *Especificación de tiempos.*

b) *Información de objetivos.*

c) *Notificación de usuario.*

d) *Políticas aceptables de privilegios;*

e) *Expedidor indirecto;*

f) *Sin afirmación.*

15.1.2.1 Extensión de especificación de tiempos

Una AA puede utilizar una extensión de especificación de tiempos para restringir los periodos de tiempo específicos durante los cuales el titular del privilegio asignado en el certificado que contiene esa extensión, puede afirmar el privilegio. Por ejemplo, una AA puede expedir un certificado asignando privilegios que sólo pueden ser afirmados entre el lunes y el viernes y entre las 9 de la mañana y las 5 de la tarde. Otro ejemplo, en el caso de delegación, podría ser un gestor que delega la autoridad de firma a un subordinado durante el tiempo en el que el gestor esté fuera de vacaciones.

Este campo se define como sigue:

```
timeSpecification EXTENSION  ::= {
    SYNTAX              TimeSpecification
    IDENTIFIED BY       id-ce-timeSpecification }
```

Esta extensión puede estar presente en certificados de atributo o en certificados de clave pública expedidos por las AA, incluidas las SOA, a entidades que pueden actuar como asertores de privilegios, incluidas otras AA y entidades finales. Esta extensión no se incluirá en certificados que contengan la extensión de identificador SOA ni en certificados expedidos a las AA que no puedan actuar también como asertores de privilegios.

Si esta extensión está presente en un certificado expedido a una entidad que es una AA, sólo aplica a la afirmación de entidad de los privilegios contenidos en el certificado. No influyen en el periodo de tiempo durante el cual la AA es capaz de expedir certificados.

Puesto que esta extensión está especificando efectivamente un refinamiento sobre el periodo de validez del certificado que la contiene, esta extensión deberá indicarse como crítica (es decir, al incluir esta extensión el expedidor está definiendo explícitamente que la asignación de privilegios no es válida fuera del periodo de tiempo especificado).

Si esta extensión está presente, pero el verificador de privilegios no la entiende, el certificado tendrá que ser rechazado.

15.1.2.1.1 Concordancia de especificación de tiempos

La regla de concordancia de especificación de tiempos compara la equivalencia de un valor presentado con un valor de atributo del tipo **AttributeCertificate**.

```
timeSpecificationMatch MATCHING-RULE  ::= {
    SYNTAX          TimeSpecification
    ID              id-mr-timeSpecMatch }
```

Esta regla de concordancia devuelve VERDADERO si todos los valores almacenados contienen la extensión **timeSpecification** y si los componentes que están presentes en el valor presentado concuerdan con los componentes correspondientes del valor almacenado.

15.1.2.2 Extensión de información de objetivos

La extensión de información de objetivos permite la determinación de objetivos de un certificado de atributo a un conjunto específico de servidores/servicios. Un certificado de atributo que contiene esta extensión sólo será utilizable en los servidores/servicios especificados.

Este campo se define como sigue:

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

Targets    ::=          SEQUENCE SIZE (1..MAX) OF Target

Target     :    :=      CHOICE {
    targetName              [0]     GeneralName,
    targetGroup             [1]     GeneralName,
    targetCert              [2]     TargetCert }

TargetCert      ::=     SEQUENCE {
    targetCertificate   IssuerSerial,
    targetName          GeneralName OPTIONAL,
    certDigestInfo      ObjectDigestInfo OPTIONAL }
```

El componente **targetName**, si está presente, proporciona el nombre de los servidores/servicios a los cuales se dirige el certificado de atributo que los contiene.

El componente **targetGroup**, si está presente, proporciona el nombre de un grupo objetivo al cual se dirige el certificado de atributo que lo contiene. La forma en que se determina el hecho de ser miembro de un objetivo en un **targetGroup** está fuera del ámbito de esta Especificación.

El componente **targetCert**, si está presente, identifica los servidores/servicios objetivos mediante referencia a su certificado.

Esta extensión puede estar presente en certificados de atributo expedidos por las AA, incluidas las SOA, a entidades que pueden actuar como asertores de privilegios, incluidas otras AA y entidades finales. Esta extensión no se incluirá en certificados de clave pública o en certificados de atributo expedidos por las AA que también puedan actuar como asertores de privilegios.

Si esta extensión está presente en un certificado de atributo expedido a una entidad que es una AA, sólo se aplica a esa aserción de entidad de los privilegios contenidos en el certificado. No influye en la capacidad de la AA para expedir certificados.

Esta extensión siempre es crítica.

Si esta extensión esta presente, pero el verificador de privilegios no se encuentra entre los especificados, el certificado de atributo será rechazado.

Si esta extensión no está presente, el certificado de atributo no tiene objetivo y puede ser aceptado por cualquier servidor.

15.1.2.3 Extensión de notificación de usuario

Esta extensión de notificación de usuario permite que una AA incluya una notificación que se indicará al titular, cuando se afirme su privilegio, y/o a un verificador de privilegios cuando se utilice el certificado de atributo que contiene esa extensión.

Este campo se define como sigue:

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

Esta extensión puede estar presente en certificados de atributo o en certificados de clave pública expedidos por las AA, incluidas las SOA, a entidades que pueden actuar como asertores de privilegios; incluidas otras AA y entidades finales. Esta extensión no se incluirá en certificados que contengan la extensión de identificador SOA o en certificados expedidos a las AA que no puedan también actuar como asertores de privilegios.

Si esta extensión está presente en un certificado expedido a una entidad que es una AA, sólo aplica a esa aserción de entidad de los privilegios contenidos en el certificado. No influye en la capacidad de la AA para expedir certificados.

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

Si esta extensión se señala como crítica mediante banderas, las notificaciones del usuario deberán presentarse a un verificador de privilegios cada vez que se afirme un privilegio. Si el asertor de privilegios proporciona el certificado de atributo al verificador de privilegios (esto es, el verificador de privilegios no lo recupera directamente de un depositario), deberán presentarse también al asertor de privilegios las notificaciones del usuario.

Si esta extensión se indica como no crítica mediante banderas, el privilegio afirmado en el certificado puede estar garantizado por un verificador de privilegios, independientemente de si se indicaron o no las notificaciones de usuario al asertor de privilegios y/o al verificador de privilegios.

15.1.2.4 Extensión de políticas aceptables de privilegios

El campo de políticas aceptables de privilegios se utiliza para constreñir la aserción de privilegios asignados para su utilización con un conjunto específico de políticas de privilegios.

El campo se define como sigue:

```
acceptablePrivilegePolicies EXTENSION ::= {
    SYNTAX              AcceptablePrivilegePoliciesSyntax
    IDENTIFIED BY       id-ce-acceptablePrivilegePolicies }

AcceptablePrivilegePoliciesSyntax       ::=     SEQUENCE SIZE (1..MAX) OF PrivilegePolicy
```

Esta extensión puede estar presente en certificados de atributo o en certificados de clave pública expedidos por las AA, incluidas las SOA, a otras AA o a entidades finales. Si esta extensión está contenida en un certificado de clave pública se refiere únicamente a la capacidad del sujeto para actuar como un asertor de privilegios para los privilegios contenidos en la extensión **subjectDirectoryAttributes**.

Si está presente, esta extensión se indicará como crítica mediante banderas.

Si esta extensión está presente y el verificador de privilegios la entiende, el verificador tendrá que asegurarse de que la política de privilegios con la que se están comparando estos privilegios es una de las identificadas en esta extensión.

Si esta extensión está presente, pero el verificador de privilegios no la entiende, se tendrá que rechazar el certificado.

15.1.2.5 Extensión de expedidor indirecto

En algunos entornos, el privilegio puede delegarse indirectamente. En esos casos, el delegador solicita a una AA que expida un privilegio de delegación de certificado en su nombre a otra entidad. El campo expedidor indirecto se utiliza tanto para un certificado de atributo como para un certificado de clave pública expedidos a una AA por una SOA. La presencia de esta extensión significa que la AA sujeto está autorizada por esa SOA para actuar como mandatario y expedir certificados que puedan delegar el privilegio, en nombre de otros delegadores.

```
indirectIssuer EXTENSION ::= {
  SYNTAX              BOOLEAN
  IDENTIFIED BY       id-ce-indirectIssuer }
```

Esta extensión siempre es no crítica.

La regla de concordancia de expedidor indirecto compara la igualdad entre un valor presentado y un valor de atributo del tipo **AttributeCertificate.**

```
indirectIssuerMatch MATCHING-RULE  ::= {
SYNTAX  BOOLEAN
ID id-mr-indirectIssuerMatch }
```

Esta regla de concordancia devuelve TRUE (verdadero) si el valor almacenado contiene la extensión **indirectIssuer** y si el valor presente en el valor presentado concuerda con el valor almacenado.

15.1.2.6 Extension sin aserción

Si está presente, esta extensión indica que el titular del AC no puede afirmar los privilegios indicados en los atributos del AC. Este campo sólo puede insertarse en AC de AA, y no en AC de entidad final. Si está presente, esta extensión se marcará siempre como crítica.

```
noAssertion EXTENSION ::= {
SYNTAX NULL
IDENTIFIED BY  id-ce-noAssertion }
```

15.2 Extensiones de revocación de privilegios

15.2.1 Requisitos

Los siguientes requisitos se relacionan con la revocación de certificados de atributo:

a) Para controlar los tamaños de las CRL, puede ser necesario asignar subconjuntos del conjunto de todos los certificados expedidos por una AA a diferentes CRL.

b) Los expedidores de certificados de atributo deben ser capaces de indicar, en un certificado de atributo, que no se dispone de información de revocación para ese certificado.

15.2.2 Campos de extensión de revocación de privilegios

Se definen los siguientes campos de extensión:

a) *puntos de distribución de CRL;*

b) *no hay información de revocación.*

15.2.2.1 Extensión de puntos de distribución de CRL

La extensión de puntos de distribución de CRL se define en la sección 2 de esta Especificación para su utilización en certificados de clave pública. Este campo también se puede incluir en un certificado de atributo. Puede estar presente en certificados expedidos a las AA, incluidas las SOA, así como en certificados expedidos a entidades finales.

Si está presente en un certificado, el verificador de privilegios procesará esta extensión exactamente de la misma manera que se describe en la sección 2 para certificados de clave pública.

15.2.2.2 Extensión no hay información de revocación

En algunos entornos (por ejemplo, cuando se expiden certificados de atributo con periodos de validez muy cortos), puede no ser necesario revocar certificados. Una AA puede utilizar esta extensión para indicar que no se proporciona información del estado de revocación para este certificado de atributo. Este campo se define como sigue:

```
noRevAvail EXTENSION ::=  {
   SYNTAX              NULL
   IDENTIFIED BY       id-ce-noRevAvail }
```

Esta extensión puede estar presente en certificados de atributo expedidos por las AA, incluidas las SOA a entidades finales. Esta extensión no se incluirá en certificados de clave pública o en certificados de atributo expedidos a las AA.

Esta extensión siempre es no crítica.

Si esta extensión está presente en un certificado de atributo, un verificador de privilegios no precisa buscar información del estado de revocación.

15.3 Extensiones de fuente de autoridad

15.3.1 Requisitos

Los siguientes requisitos se relacionan con las fuentes de autoridad:

a) En algunos entornos es necesario que una CA controle estrechamente las entidades que pueden actuar como SOA.

b) Es preciso que las SOA responsables establezcan las definiciones de sintaxis y las reglas de dominación válidas para los atributos de privilegios disponibles.

15.3.2 Campos de extensión de SOA

Se definen los siguientes campos de extensión:

a) *Identificador SOA.*

b) *Descriptor de atributo.*

15.3.2.1 Extensión de identificador SOA

La extensión de identificador SOA indica que el sujeto del certificado puede actuar como una SOA para fines de gestión de privilegios. Como tal, el sujeto del certificado puede definir atributos que asignan privilegios, expedir certificados de descriptor de atributos para dichos atributos y utilizar la clave pública correspondiente a la clave pública certificada para expedir certificados que asignan privilegios a los titulares. Estos certificados subsiguientes pueden ser certificados de atributo o certificados de clave pública con una extensión **subjectDirectoryAttributes** que contiene los privilegios.

En algunos entornos no es necesaria esta extensión y se pueden utilizar otros mecanismos para determinar las entidades que pueden actuar como SOA. Esta extensión se requiere únicamente en entornos en los que una CA precise un control centralizado estrecho para gestionar las entidades que actúan como SOA.

Este campo se define como sigue:

```
sOAIdentifier EXTENSION  ::= {
    SYNTAX               NULL
    IDENTIFIED BY        id-ce-sOAIdentifier }
```

Si esta extensión no está presente en un certificado, se deberá determinar mediante otros medios la capacidad del sujeto/titular para actuar como una SOA.

Este campo sólo puede estar presente en un certificado de clave pública expedido a una SOA. No se incluirá en certificados de atributo o en certificados de clave pública expedidos a otras AA o a titulares de privilegios de entidad final.

La certificación cruzada se aplica sólo a certificados de clave pública y no a certificados de atributos. Por consiguiente, un certificado cruzado expedido a la CA que es la expedidora de un certificado que contiene la extensión de identificador de SOA no ofrece confianza transitiva a la SOA identificada en esta extensión.

Esta extensión siempre es no crítica.

15.3.2.1.1 Concordancia de identificador de SOA

La regla de concordancia de identificador de SOA compara un valor presentado con un valor de atributo de tipo **Certificate**.

```
sOAIdentifierMatch MATCHING-RULE ::= {
  SYNTAX      NULL
  ID          id-mr-sOAIdentifierMatch }
```

Esta regla de concordancia arroja VERDADERO si el valor almacenado contiene una extensión de identificador de SOA.

15.3.2.2 Extensión de descriptor de atributo

Los verificadores de privilegios necesitan la definición de un atributo de privilegios y las reglas de dominación que gobiernan la delegación subsiguiente de dicho privilegio para asegurar que la autorización se realiza correctamente. Estas definiciones y reglas las pueden proporcionar los verificadores de privilegios en una diversidad de formas fuera del ámbito de esta Especificación (por ejemplo, pueden estar localmente configuradas en el verificador de privilegios).

Esta extensión proporciona un mecanismo que puede ser utilizado por una SOA para que los verificadores de privilegios dispongan de definiciones de atributo y de las reglas de dominación asociadas. Un certificado de atributo que contiene esta extensión se denomina un certificado de descriptor de atributo y es un tipo especial de certificado de atributo. Aunque es sintácticamente idéntico a un **AttributeCertificate** un certificado descriptor de atributo:

- contiene una **SECUENCIA** vacía en su campo **attributes**;
- es un certificado autoexpedido (es decir, el expedidor y el titular son la misma entidad);
- incluye la extensión de descriptor de atributo.

El campo se define como sigue:

```
attributeDescriptor EXTENSION  ::= {
    SYNTAX              AttributeDescriptorSyntax
    IDENTIFIED BY       {id-ce-attributeDescriptor } }

AttributeDescriptorSyntax  ::= SEQUENCE {
    identifier                  AttributeIdentifier,
    attributeSyntax             OCTET STRING (SIZE(1..MAX)),
    name                [0]     AttributeName   OPTIONAL,
    description         [1]     AttributeDescription    OPTIONAL,
    dominationRule              PrivilegePolicyIdentifier}

AttributeIdentifier  ::= ATTRIBUTE.&id({AttributeIDs})

AttributeIDs ATTRIBUTE  ::= {...}

AttributeName  ::= UTF8String(SIZE(1..MAX))

AttributeDescription  ::= UTF8String(SIZE(1..MAX))

PrivilegePolicyIdentifier ::=   SEQUENCE {
    privilegePolicy             PrivilegePolicy,
    privPolSyntax               InfoSyntax }
```

El componente **identifier** de un valor de la extensión **attributeDescriptor** es el identificador de objeto que identifica el tipo de atributo.

El componente **attributeSyntax** contiene la definición ASN.1 de la sintaxis de atributo. Dicha definición ASN.1 deberá darse como especificada para el componente de información del atributo operacional Matching Rules definido en la Rec. UIT-T X.501 | ISO/CEI 9594-2.

El componente **name** contiene como opción un nombre fácil de utilizar gracias al cual puede reconocerse el atributo.

El componente **description** contiene como opción una descripción del atributo fácil de utilizar.

El componente **dominationRule** especifica, para el atributo, lo que significa para un privilegio delegado ser "menos que" el privilegio correspondiente que tiene el delegador. El componente **privilegePolicy** identifica, mediante su identificador de objeto, el ejemplar de la política de privilegios que contiene las reglas. El componente **privPolSyntax** contiene la propia política de privilegios o un puntero hacia una ubicación en la que puede situarse. Si se incluye un puntero, también se puede incluir un troceo facultativo de la política de privilegios para permitir una comprobación de integridad en la política de privilegios referenciada.

Esta extensión sólo puede estar presente en certificados de descriptor de atributos. Esta extensión no estará presente en certificados de clave pública o en certificados de atributo distintos de los certificados autoexpedidos de las SOA.

Esta extensión siempre será no crítica.

El certificado de descriptor de atributos, creado por la SOA en el instante de creación/definición del tipo de atributo correspondiente, constituye un medio mediante el cual se puede entender y reforzar en la infraestructura la constricción universal de delegación "hacia abajo". En un directorio los certificados de atributo que contienen esta extensión se almacenarían en el atributo **attributeDescriptorCertificate** del asiento de directorio de la SOA.

15.3.2.2.1 Concordancia de descriptores de atributo

La regla de concordancia de descriptores de atributo compara la equivalencia de un valor presentado con un valor de atributo del tipo **AttributeCertificate**.

```
attDescriptor MATCHING-RULE  ::= {
    SYNTAX          AttributeDescriptorSyntax
    ID              id-mr-attDescriptorMatch }
```

Esta regla de concordancia devuelve VERDADERO si todos los valores almacenados contienen la extensión **attributeDescriptor** y si los componentes que están presentes en el valor presentado concuerdan con los componentes correspondientes del valor almacenado.

15.4 Extensiones de cometidos

15.4.1 Requisitos

El siguiente requisito se relaciona con los cometidos:

– Si un certificado es un certificado de asignación de cometidos, un verificador de privilegios necesita ser capaz de ubicar el certificado de especificación de cometidos correspondiente que contiene los privilegios específicos asignados al propio cometido.

15.4.2 Campos de extensión de cometidos

Se define el siguiente campo de extensión:

– *Identificador de certificado de especificación de cometidos.*

15.4.2.1 Extensión de identificador de certificados de especificación de cometidos

Esta extensión la puede utilizar una AA como puntero hacia un certificado de especificación de cometidos que contiene la asignación de privilegios a un cometido. Puede estar presente en un certificado de asignación de cometidos (es decir, un certificado que contiene el atributo **role**).

Un verificador de privilegios, cuando trata con un certificado de asignación de cometidos, necesita obtener el conjunto de privilegios de dicho cometido para determinar si tiene éxito o fracasa en la verificación. Si los privilegios se asignaron al cometido en un certificado de especificación de cometido, este campo se puede utilizar para ubicar dicho certificado.

Este campo se define como sigue:

```
roleSpecCertIdentifier EXTENSION ::=
    {
    SYNTAX              RoleSpecCertIdentifierSyntax
    IDENTIFIED BY       { id-ce-roleSpecCertIdentifier }
}

RoleSpecCertIdentifierSyntax  ::= SEQUENCE SIZE (1..MAX) OF RoleSpecCertIdentifier

RoleSpecCertIdentifier  ::= SEQUENCE {
    roleName                    [0]     GeneralName,
    roleCertIssuer              [1]     GeneralName,
    roleCertSerialNumber        [2]     CertificateSerialNumber   OPTIONAL,
    roleCertLocator             [3]     GeneralNames              OPTIONAL
    }
```

roleName identifica el cometido. Este nombre sería el mismo que el del componente **holder** del certificado de especificación de cometidos al que se hace referencia en esta extensión.

roleCertIssuer identifica la AA que expidió el certificado de especificación de cometidos referenciado.

roleCertSerialNumber, si está presente, contiene el número de serie del certificado de especificación de cometidos. Hay que destacar que si los privilegios asignados al propio cometido cambian, se expediría al cometido un nuevo certificado de especificación de cometidos. Sería entonces necesario sustituir cualquier certificado que contenga esta extensión, incluido el componente **roleCertSerialNumber**, por certificados que se refieran al nuevo número de serie. Aunque se precisa este comportamiento en algunos entornos, en muchos otros no es deseable. Normalmente este componente estaría ausente, permitiendo la actualización automática de los privilegios asignados al propio cometido, sin influir en los certificados de asignación de cometidos.

roleCertLocator, si está presente, contiene información que se puede utilizar para localizar el certificado de especificación de cometidos.

Esta extensión puede estar presente en los certificados de asignación de cometidos que son certificados de atributo o certificados de clave pública expedidos por las AA, incluidas las SOA, a otras AA o a los titulares de privilegios de entidad final. Esta extensión no se incluirá en certificados que contiene una extensión de identificador de SOA.

Si está presente, esta extensión puede ser utilizada por un verificador de privilegios para ubicar el certificado de especificación de cometidos.

Si esta extensión no está presente:

a) deben utilizarse otros medios para ubicar el certificado de especificación de cometidos; o

b) se utilizaron mecanismos diferentes del certificado de especificación de cometidos para asignar privilegios al cometido (por ejemplo, los privilegios de cometido se pueden configurar localmente en el verificador de privilegios).

Esta extensión es siempre no crítica.

15.4.2.1.1 Concordancia de ID de certificado de especificación de cometidos

La regla de concordancia de identificadores de certificado de especificación de cometidos compara la equivalencia de un valor presentado con un valor de atributo del tipo **AttributeCertificate.**

```
roleSpecCertIdMatch MATCHING-RULE  ::= {
    SYNTAX          RoleSpecCertIdentifierSyntax
    ID              id-mr-roleSpecCertIdMatch }
```

Esta regla de concordancia devuelve VERDADERO si el valor almacenado contiene la extensión **roleSpecCertIdentifier** y si los componentes que están presentes en el valor presentado concuerdan con los componentes correspondientes del valor almacenado.

15.5 Extensiones de delegación

15.5.1 Requisitos

Los siguientes requisitos se relacionan con la delegación de privilegios:

a) los certificados de privilegios de entidad final necesitan poderse distinguir de los certificados de AA, para evitar que las entidades finales se constituyan ellas mismas en AA sin autorización. También es necesario que una AA sea capaz de limitar la longitud de un trayecto de delegación subsiguiente;

b) una AA necesita ser capaz de especificar el espacio de nombre adecuado en el que puede producirse la delegación de privilegios. El verificador de privilegios necesita poder comprobar la adhesión a dichas constricciones;

c) una AA necesita ser capaz de especificar las políticas de certificado aceptables que deberán utilizar los asertores de privilegios a lo largo del trayecto de delegación para autenticarse ellos mismos cuando esta AA afirma una delegación de privilegios;

d) un verificador de privilegios necesita ser capaz de ubicar el certificado de atributo correspondiente para que un expedidor asegure que el expedidor tiene suficiente privilegios para delegar el privilegio en el certificado actual;

e) existe el requisito de un servicio de delegación (DS) independiente para expedir certificados que puedan delegar privilegios, aunque el servidor DS no puede actuar como detentador de esos privilegios.

15.5.2 Campos de extensión de delegación

Se definen los siguientes campos de extensión:

a) *constricciones de atributo básico;*

b) *constricciones de nombre delegado;*

c) *políticas de certificado aceptable;*

d) *identificador de atributo de autoridad;*

e) *expedidor indirecto;*

f) *expedido en nombre de.*

15.5.2.1 Extensión de constricciones de atributo básico

Este campo indica si está permitida la delegación subsiguiente de los privilegios asignados en el certificado que contiene esta extensión. Si es así, se debe especificar también una constricción de longitud del trayecto de delegación.

Este campo se define como sigue:

```
basicAttConstraints EXTENSION ::=
{
        SYNTAX                  BasicAttConstraintsSyntax
        IDENTIFIED BY           { id-ce-basicAttConstraints }
}
```

```
BasicAttConstraintsSyntax ::= SEQUENCE
   {
      authority              BOOLEAN DEFAULT FALSE,
      pathLenConstraint      INTEGER (0..MAX) OPTIONAL
   }
```

El componente **authority** indica si el titular está autorizado o no para delegar el privilegio. Si **authority** es **VERDADERO** el titular es también una AA y tiene autorización para delegar ulteriormente el privilegio, dependiendo de las constricciones pertinentes. Si **authority** es **FALSO**, el titular es una entidad final y no está autorizado a delegar el privilegio.

El componente **pathLenConstraint** tiene sentido únicamente si **authority** está fijado a **VERDADERO**. Da el número máximo de certificados de AA que pueden seguir a este certificado en un trayecto de delegación. El valor **0** indica que el sujeto de este certificado puede expedir certificados sólo a entidades finales y no a AA. Si no aparece el campo **pathLenConstraint** en ningún certificado del trayecto de delegación, no hay límite a la longitud permitida del trayecto de delegación. Hay que destacar que la constricción entra en vigor con el siguiente certificado en el trayecto. La constricción controla el número de certificados de AA entre el certificado de AA que contiene la constricción y el certificado de entidad final. La constricción restringe la longitud del segmento del trayecto de delegación entre el certificado que contiene esta extensión y el certificado de la entidad final. No influye en el número de certificados AA en el trayecto de delegación entre el ancla de confianza y el certificado que contiene esta extensión. Por lo tanto, la longitud de un trayecto de delegación completo puede ser superior a la longitud máxima del segmento constreñido por esta extensión. La constricción controla el número de certificados AA entre el certificado AA que contiene la constricción y el certificado de la entidad final. Por lo tanto, la longitud total de este segmento del trayecto puede ser superior al valor de la constricción como mucho en dos certificados. (Esto incluye el certificado de los dos extremos del segmento más los certificados de AA entre los dos puntos extremos que están constreñidos por el valor de esta extensión.)

Esta extensión puede estar presente en certificados de atributo o en certificados de clave pública expedidos por las AA, incluidas las SOA, otras AA y entidades finales. Esta extensión no se incluirá en certificados que contienen la extensión de identificador de SOA.

Si esta extensión está presente en un certificado de atributo, y **authority** es **VERDADERO**, se autoriza al titular a expedir certificados de atributo subsiguientes, delegando los privilegios contenidos a otras entidades, pero no certificados de clave pública.

Si esta extensión está presente en un certificado de clave pública, y si la extensión **basicConstraints** indica que el sujeto es también una CA, se autoriza al sujeto a expedir certificados de clave pública subsiguientes que delegan estos privilegios a otras entidades, pero no certificados de atributo. Si se incluye la constricción de longitud de trayecto, el sujeto sólo puede delegar en la intersección de la constricción especificada en esta extensión y de cualquier otra especificada en la extensión **basicConstraints**. Si esta extensión está presente en un certificado de clave pública pero la extensión **basicConstraints** está ausente, o indica que el sujeto es una entidad final, el sujeto no está autorizado a delegar los privilegios.

A opción del expedidor de certificados, esta extensión puede ser crítica o no crítica. Se recomienda que se indique como crítica mediante banderas, en otro caso un titular que no esté autorizado a ser una AA puede expedir certificados y el verificador de privilegios puede utilizar un certificado de este tipo involuntariamente.

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

- si el valor de **authority** no está fijado a **VERDADERO,** entonces el atributo delegado no se debe utilizar para delegarlo ulteriormente;
- si el valor de **authority** está fijado a **VERDADERO** y está presente **pathLenConstraint**, entonces el verificador de privilegios comprobará que el trayecto de delegación que se está procesando es coherente con el valor de **pathLenConstrain**.

Si esta extensión está presente, indicada como no crítica mediante banderas, y el verificador de privilegios no la reconoce, entonces dicho sistema utilizará otros medios para determinar si el atributo delegado puede utilizarse para su posterior delegación.

Si la extensión no está presente, o si la extensión está presente con un valor **SECUENCIA** vacío el titular estará obligado a ser sólo una entidad final y no una autoridad de atributo y el titular no permitirá la delegación de los privilegios contenidos en el certificado de atributo.

15.5.2.1.1 Concordancia de constricciones de atributo básico

La regla de concordancia de constricciones de atributo básico compara un valor presentado con un valor de atributo del tipo **AttributeCertificate.**

```
basicAttConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX          BasicAttConstraintsSyntax
    ID              id-mr-basicAttConstraintsMatch }
```

Esta regla de concordancia devuelve VERDADERO si el valor almacenado contiene la extensión **basicAttConstraints** y si los componentes que están presentes en el valor presentado concuerdan con los componentes correspondientes del valor almacenado.

15.5.2.2 Extensión de constricciones de nombre delegado

El campo de constricciones de nombre delegado indica un espacio de nombre en el que necesitan ubicarse todos los nombres de titulares de certificados subsiguientes en un trayecto de delegación.

Este campo se define como sigue:

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

Esta extensión se procesa de la misma forma que la extensión **nameConstraints** para certificados de clave pública. Si **permittedSubtrees** está presente, de todos los certificados de atributo expedidos por la AA titular y por las AA subsiguientes en el trayecto de delegación sólo son aceptables aquellos certificados de atributo con nombres de titular dentro de estos subárboles. Si **excludedSubtrees** está presente, cualquier certificado de atributo expedido por la AA titular o las AA subsiguientes en el trayecto de delegación que tenga un nombre de titular en estos subárboles es inaceptable. Si tanto **permittedSubtrees** como **excludedSubtrees** están presentes y los espacios de nombre se superponen, la declaración de exclusión tiene precedencia.

Esta extensión puede estar presente en certificados de atributo o en certificados de clave pública expedidos por las AA, incluidas las SOA, a otras AA. Esta extensión no se incluirá en certificados expedidos a entidades finales o en certificados que contienen la extensión de identificador de SOA.

Si esta extensión está presente en un certificado de clave pública, y si la extensión **nameConstraints** está presente, el sujeto sólo puede delegar en la intersección entre la constricción especificada en esta extensión y la especificada en la extensión **nameConstraints**.

A opción del expedidor del certificado de atributo, esta extensión puede ser crítica o no crítica. Se recomienda que se indique como crítica mediante banderas, en otro caso un usuario de certificado de atributo puede comprobar qué certificados de atributo subsiguientes en un trayecto de delegación están situados en el espacio de nombre pretendido por la AA expedidora.

15.5.2.2.1 Concordancia de constricciones de nombre delegado

La regla de concordancia de constricciones de nombre delegado compara la equivalencia de un valor presentado con un valor de atributo del tipo **AttributeCertificate.**

```
delegatedNameConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX          NameConstraintsSyntax
    ID              id-mr-delegatedNameConstraintsMatch}
```

Esta regla de concordancia devuelve VERDADERO si el valor almacenado contiene la extensión **attributeNameConstraints** y si los componentes que están presentes en el valor presentado concuerdan con los componentes correspondientes del valor almacenado.

15.5.2.3 Extensión de políticas de certificados aceptables

El campo de políticas de certificados aceptables se utiliza, por delegación de certificados de atributo, para controlar las políticas de certificados aceptables mediante las que se necesitan expedir los certificados de clave pública para titulares subsiguientes en un trayecto de delegación. Al enumerar un conjunto de políticas en este campo una AA está solicitando que expedidores subsiguientes en un trayecto de delegación deleguen únicamente los privilegios contenidos a titulares que tienen certificados de clave pública expedidos mediante una o más de las políticas de certificados enumeradas. Las políticas enumeradas aquí no son políticas con las que se expidió el certificado de atributo, sino políticas con las que necesitan haberse expedido los certificados de clave pública aceptables para titulares subsiguientes.

Este campo se identifica como sigue:

```
acceptableCertPolicies EXTENSION ::= {
    SYNTAX              AcceptableCertPoliciesSyntax
    IDENTIFIED BY       id-ce-acceptableCertPolicies }

AcceptableCertPoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId

CertPolicyId ::= OBJECT IDENTIFIER
```

Esta extensión puede estar sólo presente en certificados de atributo expedidos por las AA, incluidas las SOA, a otras AA. Esta extensión no se incluirá en certificados de atributo de entidad final ni en certificados de clave pública. En el caso de una delegación que utiliza certificados de clave pública, esta misma funcionalidad se proporciona mediante **certificatePolicies** y otras extensiones relacionadas.

Si está presente esta extensión, se indicará como crítica mediante banderas.

Si esta extensión está presente y el verificador de privilegios la comprende, el verificador tendrá que asegurar que todos los asertores de privilegios subsiguientes en el trayecto de delegación son autenticados con un certificado de clave pública según una o más de las políticas de certificado enumeradas.

Si esta extensión está presente, pero el verificador de privilegios no la entiende, se tendrá que rechazar el certificado.

15.5.2.3.1 Concordancia de políticas de certificados aceptables

La regla de concordancia de las políticas de certificados aceptables compara la equivalencia de un valor presentado con un valor de atributo del tipo **AttributeCertificate.**

```
acceptableCertPoliciesMatch MATCHING-RULE  ::= {
    SYNTAX          AcceptableCertPoliciesSyntax
    ID              id-mr-acceptableCertPoliciesMatch }
```

Esta regla de concordancia devuelve VERDADERO si el valor almacenado contiene la extensión **acceptableCertPolicies** y si los componentes que están presentes en el valor presentado concuerdan con los componentes correspondientes del valor almacenado.

15.5.2.4 Extensión de identificador de atributos de autoridad

En la delegación de privilegios, una AA que delega privilegios tendrá que tener por lo menos el mismo privilegio y la autoridad para delegar dicho privilegio. Una AA que está delegando privilegios a otra AA o a una entidad final puede situar esta extensión en el certificado de AA o de entidad final que expide. La extensión es un puntero hacia el certificado en el que se asignó el privilegio correspondiente al expedidor del certificado que contiene la extensión. El verificador de privilegios puede utilizar la extensión para asegurar que la AA expedidora tenía un privilegio suficiente para delegar al titular del certificado que contiene esta extensión.

Este campo se define como sigue:

```
authorityAttributeIdentifier EXTENSION  ::=
    {
    SYNTAX              AuthorityAttributeIdentifierSyntax
    IDENTIFIED BY       { id-ce-authorityAttributeIdentifier }
    }
AuthorityAttributeIdentifierSyntax   ::=      SEQUENCE SIZE (1..MAX) OF AuthAttId
AuthAttId       ::=     IssuerSerial
```

Un certificado que contiene esta extensión puede incluir la delegación de múltiples privilegios al titular del certificado. Si la asignación de esos privilegios a la AA que expidió este certificado se realizó en más de un certificado, entonces esta extensión suele incluir más de un puntero.

Esta extensión puede estar presente en certificados de atributo o en certificados de clave pública expedidos por las AA a otras AA o a titulares de privilegios de entidad final. Esta extensión no se incluirá en certificados expedidos por una SOA o en certificados de clave pública que contengan la extensión de identificador de SOA.

Esta extensión siempre es no crítica.

15.5.2.4.1 Concordancia de identificadores de AA

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

```
authAttIdMatch MATCHING-RULE  ::= {
    SYNTAX          AuthorityAttirbuteIdentifierSyntax
    ID              id-mr-authAttIdMatch }
```

Esta regla de concordancia devuelve VERDADERO si el valor almacenado contiene la extensión **authorityAttributeIdentifier** y si los componentes que están presentes en el valor presentado concuerdan con los componentes correspondientes del valor almacenado.

15.5.2.5 Extensión de expedidor indirecto

En algunos entornos, el privilegio puede delegarse indirectamente. En esos casos, el delegador solicita a un servidor DS que expida un privilegio de delegación de certificado en su nombre a otra entidad. El campo expedidor indirecto se utiliza tanto para un certificado de atributo como para un certificado de clave pública expedidos a un servidor DS por una SOA. La presencia de esta extensión significa que la AA sujeto (el servidor DS) está autorizada por esa SOA para actuar como mandatario y expedir certificados que puedan delegar el privilegio, en nombre de otros delegadores.

```
indirectIssuer EXTENSION ::= {
    SYNTAX          NULL
    IDENTIFIED BY       id-ce-indirectIssuer }
```

Esta extensión siempre es no crítica.

La regla de concordancia de expedidor indirecto compara la igualdad de un valor presentado con un valor de atributo del tipo **AttributeCertificate.**

```
indirectIssuerMatch MATCHING-RULE  ::= {
    SYNTAX   NULL
    ID  id-mr-indirectIssuerMatch }
```

Esta regla de concordancia devuelve VERDADERO si el valor almacenado contiene la extensión **indirectIssuer** y si el valor presente en el valor presentado concuerda con el valor almacenado.

15.5.2.6 Expedido en nombre de

Esta extensión es insertada en un AC por un expedidor indirecto (servidor DS). Indica la AA que solicitó al servidor DS expedir el AC, y permite la construcción y la validación de la cadena de delegación.

```
issuedOnBehalfOf EXTENSION ::= {
    SYNTAX GeneralName
    ID id-ce-issuedOnBehalfOf }
```

`GeneralName` es el nombre de la AA que solicitó al usuario indirecto (servidor DS) expedir este AC.

El expedidor de esta AC debe haber recibido de una SOA el privilegio para expedir los AC en nombre de otras AA, a través de la extensión IndirectIssuer en su AC.

Esta extensión puede ser crítica o no crítica según proceda para garantizar la validación del trayecto de delegación.

16 Procedimiento de procesamiento de trayectos de privilegios

El procesamiento de trayectos de privilegios lo realiza un verificador de privilegios. Las reglas de procesamiento de trayectos para certificados de atributo son de alguna manera análogas a las de los certificados de clave pública.

Otros componentes del procesamiento de trayectos que no se tratan en esta cláusula incluyen la verificación de firmas de certificado, la validación de periodos de validez de certificados, etc.

Para trayectos de privilegios constituidos por un único certificado, (es decir, la SOA asignó los privilegios directamente al asertor de privilegios) sólo se requiere el procedimiento básico descrito en 16.1, a menos que el privilegio sea asignado a un cometido. En ese caso, si el verificador de privilegios no está configurado con los privilegios específicos del cometido, puede necesitar obtener el certificado de especificación de cometidos que asigna los privilegios específicos al cometido, según se describe en 16.2. Si una AA intermediaria delegó su privilegio al asertor de privilegios, entonces se requiere también el procedimiento de trayecto de delegación descrito en 16.3. Estos procedimientos no se realizan secuencialmente. El procedimiento de procesamiento de cometidos y el procedimiento de procesamiento de delegación se realizan antes de determinar si los privilegios afirmados son suficientes para el contexto de utilización en el procedimiento básico.

16.1 Procedimiento de procesamiento básico

Es imprescindible verificar la firma de todos los certificados en el trayecto. En esta cláusula no se repiten procedimientos relativos a la validación de firmas y de certificados de clave pública. El verificador de privilegios tendrá que verificar la identidad de todas las entidades en el trayecto, utilizando los procedimientos de la cláusula 10. Hay que destacar que la comprobación de la firma en un certificado de atributo implica necesariamente la comprobación de la validez del certificado de clave pública referenciado. Cuando se asignan privilegios utilizando certificados de atributo,

el procesamiento de trayectos necesitará considerar elementos tanto de la PMI como de la PKI para determinar la validez última de un certificado de atributo de asertor de privilegios. Una vez confirmada esa validez, se *pueden* utilizar los privilegios contenidos en ese certificado mediante una comparación con la política de privilegios pertinente y otra información asociada con el contexto en el que se está utilizando el certificado.

El contexto de utilización tendrá que determinar si el titular del privilegio pretende realmente afirmar el privilegio contenido para su utilización en ese contexto. El hecho de que exista una cadena de certificados hacia una SOA no es en si mismo suficiente para tomar esta determinación. El deseo del titular del privilegio de utilizar ese certificado tendrá que estar claramente indicado y verificado. Sin embargo, los mecanismos para asegurar que el titular del privilegio ha demostrado adecuadamente este tipo de aserción de privilegios está fuera del ámbito de esta Especificación. Por ejemplo, este tipo de aserción de privilegios se puede verificar si el titular del privilegio firmó una referencia a ese certificado indicando así su voluntad de utilizar ese certificado para ese contexto.

Para cada certificado de atributo en el trayecto que no contenga la extensión **noRevAvail**, el verificador de privilegios tendrá que asegurar que el certificado de atributo no ha sido revocado.

El verificador de privilegios tendrá que asegurar que el privilegio afirmado es válido para el intervalo de tiempo denominado "tiempo de evaluación" lo que se puede realizar en *cualquier* instante, es decir, el instante actual de la comprobación o cualquier instante en el pasado. En el contexto de un servicio de control de acceso, la comprobación se realiza siempre para el tiempo presente. Sin embargo, en el contexto de no repudio, la comprobación se puede realizar durante un instante en el pasado o en el instante actual. Cuando se validan certificados, el verificador de privilegios tendrá que asegurar que el instante de evaluación se encuentra dentro de todos los periodos de validez de todos los certificados utilizados en el trayecto. Asimismo, si algunos de los certificados en el trayecto contiene la extensión **timeSpecification**, las constricciones establecidas durante los intervalos de tiempo en los que necesita afirmar el privilegio tendrán que permitir también que la aserción de privilegios sea válida en el instante de evaluación.

Si la extensión **targetingInformation** está presente en el certificado utilizado para afirmar un privilegio, el verificador de privilegios tendrá que comprobar que el servidor/servicio para el cual está verificando está incluido en la lista de objetivos.

Si el certificado es un certificado de asignación de cometidos, se necesita el procedimiento de procesamiento descrito en 16.2 para asegurar que se identifican los privilegios adecuados. Si el privilegio se delega a la entidad en lugar de ser asignado directamente por la SOA en la que confía el verificador de privilegios, se necesita el procedimiento de procesamiento descrito en 16.3 para asegurar que la delegación se realiza adecuadamente.

El verificador de privilegios tendrá que determinar también si los privilegios que se están afirmando son suficientes o no para el contexto de utilización. La política de privilegios establece las reglas para tomar esta decisión e incluye la especificación de cualesquiera variables ambientales que sea preciso considerar. Los privilegios afirmados, incluidos aquellos provenientes del procedimiento de cometidos de 16.2 y del procedimiento de delegación de 16.3 y cualquier variable ambiental pertinente (por ejemplo, hora del día o balance de cuenta corriente) se comparan con la política de privilegios para determinar si son o no suficientes para el contexto de utilización. Si está presente la extensión **acceptablePrivilegePolicies**, la aserción de privilegios sólo puede tener éxito si la política de privilegios con la que el verificador de privilegios la está comparando es una de las contenidas en esta extensión.

Si la comparación tiene éxito, se proporciona al verificador de privilegios cualquier notificación de usuario pertinente.

16.2 Procedimiento de procesamiento de cometidos

Si el certificado afirmado es un certificado de asignación de cometidos, el verificador de privilegios tendrá que conseguir los privilegios específicos asignados a su cometido. El nombre del cometido al que está asignado el asertor de privilegios está contenido en el atributo **role** del certificado. El verificador de privilegios, si todavía no está configurado con los privilegios del cometido denominado, puede necesitar localizar el certificado de especificación de cometidos que asigna los privilegios a ese cometido. La información del atributo **role** y de la extensión **roleSpecCertIdentifier** se puede utilizar para localizar ese certificado.

Los privilegios asignados al cometido están asignados implícitamente al asertor de privilegios y están por tanto incluidos entre los privilegios afirmados que se comparan con la política de privilegios en el procedimiento básico de 16.1 para determinar si los privilegios afirmados son o no suficientes para el contexto de utilización.

16.3 Procedimiento de procesamiento de delegaciones

Si los privilegios afirmados los delega por una AA intermediaria al asertor de privilegios, el verificador de privilegios tendrá que asegurar que el trayecto es un trayecto de delegación válido, comprobando que:

– cada AA que expidió un certificado en el trayecto de delegación estaba autorizada a hacerlo;

- cada certificado en el trayecto de delegación es válido respecto de las constricciones de trayecto y de nombre que en él se imponen;
- cada entidad en el trayecto de delegación está autenticada con un certificado de clave pública que es válido de conformidad con cualesquiera constricciones de política impuestas;
- ningún privilegio de delegación de AA sea superior al privilegio que tiene dicha AA.

Antes de comenzar la validación del trayecto de delegación, el verificador de privilegios tendrá que obtener lo siguiente. El asertor de privilegios puede proporcionar cualesquiera de ellos, o los puede obtener a partir de alguna otra fuente, como el directorio. Los atributos del servicio se pueden proporcionar al verificador de privilegios en un documento estructurado o mediante otros medios.

- Confianza establecida en la clave pública de verificación utilizada para validar la firma de la SOA en la que se confía. Esta confianza puede establecerse mediante medios fuera de banda o mediante un certificado de clave pública expedido a la SOA por la CA, en la que el verificador de privilegios ha establecido su confianza. Este tipo de certificado suele contener la extensión **sOAIdentifier**.
- El privilegio del asertor de privilegios, codificado en su extensión de certificado de atributo o de atributos de directorio de sujeto de su certificado de clave pública.
- Trayecto de delegación de certificados del asertor de privilegios a la SOA en la que confía.
- Regla de dominación para el privilegio que se está afirmando que se puede obtener a partir del descriptor de atributos expedido por la SOA responsable del atributo en cuestión o se puede obtener mediante medios fuera de banda.
- Política de privilegios; se puede obtener a partir del directorio o mediante algunos medios fuera de banda.
- Variables ambientales incluidas, por ejemplo, fecha/hora actual, balance de cuenta corriente, etc.

Una implementación será funcionalmente equivalente al comportamiento externo resultante de este procedimiento, aunque el algoritmo utilizado por una implementación determinada no está normalizado para derivar la salida o salidas correctas a partir de entradas dadas.

En caso de que los certificados de atributos sean expedidos por un expedidor indirecto (DS) la parte confiante debería validar plenamente la cadena de delegación de la siguiente manera.

i) Comenzando por la entidad final AC, la parte confiante (RP) extrae el nombre del expedidor y el nombre issuedOnBehalfOf.

ii) La RP recupera el AC del expedidor y valida que se trate de un expedidor de la SOA (es decir, tiene la extensión indirectIssuer).

iii) La RP recupera el AC de la AA issuedOnBehalfOf y valida que la AA disponga de un superconjunto de los atributos de privilegios expedidos a la entidad final.

La RP repite el paso ii) utilizando el AC de la AA, ascendiendo así en la cadena hasta que llega al AC de un AA expedido por la SOA.

16.3.1 Verificación de la integridad de la regla de dominación

La regla de dominación está asociada con el privilegio que se está delegando. La sintaxis y el método para obtener la regla de dominación no está normalizada. Si embargo, se puede verificar la integridad de la regla de dominación que se obtiene. El certificado de descriptor de atributos expedido por la SOA responsable que se está delegando puede contener un TROCEO de la regla de dominación. El verificador de privilegios puede reproducir la función de TROCEO en la copia recuperada de la regla de dominación y comparar los dos troceos. Si son idénticos, el verificador de privilegios tiene la regla de dominación precisa.

16.3.2 Establecimiento del trayecto de delegación válido

El verificador de privilegios tendrá que encontrar el trayecto de delegación y obtener certificados para cada entidad en el trayecto. El trayecto de delegación se extiende desde el asertor directo de privilegios hasta la SOA. Cada certificado intermediario en el trayecto de delegación tendrá que contener la extensión **basicAttConstraints** con el componente de autoridad fijado en **VERDADERO**. El expedidor de cada certificado será el mismo que el titular/sujeto del certificado adyacente a él en el trayecto de delegación. La extensión **authorityAttributeIdentifier** se utiliza para localizar el certificado adecuado de la entidad adyacente en el trayecto de delegación. El número de certificados en el trayecto desde cada entidad hasta el asertor de privilegios directo (inclusive) no superará en más de dos el valor de **pathLenConstraint** en la extensión **basicAttConstraints** de la entidad. Esto se debe a que **pathLenConstraint** limita el número de certificados intermedios entre los dos puntos extremos (es decir, el certificado que contiene la constricción y el certificado de entidad final) de forma que la longitud máxima es el valor de esta constricción más los certificados que están en los puntos extremos.

Si la extensión **delegatedNameConstraints** está presente en cualquiera de los certificados en el trayecto de delegación, las constricciones se procesan de la misma manera que se procesa la extensión **nameConstraints** en el procedimiento de procesamiento de trayectos de certificación de la cláusula 10.

Si la extensión **acceptableCertPolicies** está presente en cualquiera de los certificados en el trayecto de delegación, el verificador de privilegios asegurará que se realiza la autenticación de cada entidad subsiguiente en el trayecto de delegación con un certificado de clave pública que contiene por lo menos una de las políticas aceptables.

16.3.3 Verificación de la delegación de privilegios

Ningún delegador puede delegar un privilegio que sea superior al privilegio que posee. La regla de dominación en el atributo descriptor de atributo proporciona las reglas que se indican si un determinado valor es 'inferior que' otro valor para el atributo que se está delegando.

Para cada certificado en el trayecto de delegación, incluido el certificado de asertor de privilegios directo, el verificador de privilegios tendrá que asegurar que el delegador tenía autorización para delegar el privilegio que poseía y que al privilegio delegado no era superior al privilegio propio.

Para cada uno de estos certificados, el verificador de privilegios tendrá que comparar el privilegio delegado con el privilegio que tiene ese delegador, de conformidad con la regla de dominación del privilegio. El privilegio que tiene el delegador se obtiene a partir del certificado adyacente en el trayecto de delegación, como se describe en 16.2. La comparación de los dos privilegios se realiza basándose en la regla de dominación tratada en 16.3.1.

16.3.4 Determinación de éxito/fracaso

Suponiendo que se establece un trayecto de delegación válido, los privilegios del asertor de privilegios directo se consideran como entradas para la comparación con la política de privilegios, como se trata en 16.1, para determinar si el asertor de privilegios directo tiene o no un privilegio suficiente para el contexto de utilización.

17 Esquema de directorio PMI

Esta cláusula define los elementos del esquema de directorio utilizados para representar información de la PMI en el directorio. Incluye la especificación de clases de objeto pertinentes, atributos y reglas de concordancia de valores de atributos.

17.1 Clases de objeto de directorio PMI

Esta subcláusula define clases de objeto para la representación de objetos PMI en el directorio.

17.1.1 Clase de objeto usuario PMI

La clase de objeto usuario PMI se utiliza para definir inserciones para objetos que pueden ser titulares de certificados de atributo.

```
pmiUser OBJECT-CLASS ::= {
-- a PMI user (i.e., a "holder")
  SUBCLASS OF    {top}
  KIND           auxiliary
  MAY CONTAIN    {attributeCertificateAttribute}
  ID             id-oc-pmiUser }
```

17.1.2 Clase de objeto AA de PMI

La clase de objeto AA de PMI se utiliza en la definición de inserciones para objetos que actúan como autoridades de atributo.

```
pmiAA OBJECT-CLASS ::= {
-- a PMI AA
  SUBCLASS OF    {top}
  KIND           auxiliary
  MAY CONTAIN    {aACertificate |
                 attributeCertificateRevocationList |
                 attributeAuthorityRevocationList}
  ID             id-oc-pmiAA }
```

17.1.3 Clase de objeto SOA de PMI

La clase de objeto SOA de PMI se utiliza en la definición de inserciones para objetos que actúan como fuentes de autoridad. Hay que destacar que, si el objeto tiene autorización para actuar como una SOA mediante la expedición de un certificado de clave pública que contiene la extensión **sOAIdentifier**, una inserción de directorio que represente dicho objeto tendría también la clase de objeto **pkiCA**.

```
pmiSOA OBJECT-CLASS ::= {  -- a PMI Source of Authority
  SUBCLASS OF    {top}
  KIND           auxiliary
  MAY CONTAIN    {attributeCertificateRevocationList |
                 attributeAuthorityRevocationList |
                 attributeDescriptorCertificate}
  ID             id-oc-pmiSOA }
```

17.1.4 Clase de objeto punto de distribución de CRL de certificado de atributo

La clase de objeto punto de distribución de CRL de certificado de atributo se utiliza en la definición de asientos para objetos que contienen segmentos de listas de revocación de certificados de atributo y/o de autoridad de atributo. Esta clase auxiliar suele estar combinada con la clase de objeto estructural **crlDistributionPoint**, cuando se introducen asientos. Puesto que los atributos **certificateRevocationList** y **authorityRevocationList** son facultativos en esta clase, es posible crear asientos que obtienen, por ejemplo, sólo una lista de revocación de autoridad de atributo o asientos que contienen listas de revocación de múltiples tipos, dependiendo de los requisitos.

```
attCertCRLDistributionPt     OBJECT-CLASS ::= {
  SUBCLASS OF    {top}
  KIND           auxiliary
  MAY CONTAIN    { attributeCertificateRevocationList |
                 attributeAuthorityRevocationList }
  ID             id-oc-attCertCRLDistributionPts }
```

17.1.5 Trayecto de delegación de PMI

La clase de objeto trayecto de delegación de PMI se utiliza en la definición de inserciones para objetos que pueden contener trayectos de delegación. Generalmente se utilizará junto con inserciones de la clase de objeto estructural **pmiAA**.

```
pmiDelegationPath      OBJECT-CLASS   ::= {
    SUBCLASS OF        {top}
    KIND               auxiliary
    MAY CONTAIN        { delegationPath }
    ID                 id-oc-pmiDelegationPath }
```

17.1.6 Clase de objeto política de privilegios

La clase de objeto política de privilegios se utiliza en la definición de inserciones para objetos que contienen información de políticas de privilegios.

```
privilegePolicy      OBJECT-CLASS   ::= {
    SUBCLASS OF      {top}
    KIND             auxiliary
    MAY CONTAIN      {privPolicy }
    ID               id-oc-privilegePolicy }
```

17.1.7 Clase de objeto política de privilegios protegidos

La clase de objeto política de privilegios protegidos se utiliza en la definición de inserciones para objetos que contienen información de políticas de privilegios protegidas dentro de los certificados de atributos.

```
protectedPrivilegePolicy          OBJECT-CLASS   ::= {
    SUBCLASS OF      {top}
    KIND             auxiliary
```

```
    MAY CONTAIN         {protPrivPolicy }
    ID                  id-oc-protectedPrivilegePolicy }
```

17.2 Atributos de directorio de PMI

Esta subcláusula define atributos de directorio utilizados para almacenar datos PMI en asientos de directorio.

17.2.1 Atributo de certificado de atributos

El atributo siguiente contiene los certificados de atributo expedidos a un titular específico y se almacena en el asiento de directorio de ese titular.

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

17.2.2 Atributo de certificado de AA

El atributo siguiente contiene certificados de atributo expedidos a una AA y está almacenado en el asiento de directorio de la AA titular.

```
aACertificate       ATTRIBUTE   ::=  {
    WITH SYNTAX                 AttributeCertificate
    EQUALITY MATCHING RULE      attributeCertificateExactMatch
    ID                          id-at-aACertificate }
```

17.2.3 Atributo de certificado de descriptor de atributos

El atributo siguiente contiene certificados de atributo, expedidos por una SOA, que contiene la extensión **attributeDescriptor**. Estos certificados de atributo contienen la sintaxis válida y la especificación de la regla de dominación de los atributos de privilegio y están almacenados en el asiento de directorio de la SOA expedidora.

```
attributeDescriptorCertificate      ATTRIBUTE  ::= {
    WITH SYNTAX                     AttributeCertificate
    EQUALITY MATCHING RULE          attributeCertificateExactMatch
    ID                              id-at-attributeDescriptorCertificate }
```

17.2.4 Atributo de lista de revocación de certificados de atributo

El atributo siguiente contiene una lista de certificados de atributo revocados. Estas listas se pueden almacenar en el asiento de directorio de la autoridad expedidora o en otros asientos de directorio (por ejemplo, en un punto de distribución).

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

17.2.5 Atributo de lista de revocación de certificados de AA

El atributo siguiente contiene una lista de certificados de atributo revocados expedida a las AA. Estas listas se pueden almacenar en el asiento de directorio de la autoridad expedidora o en otros asientos de directorio (por ejemplo, en un punto de distribución).

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

17.2.6 Atributo de trayecto de delegación

El atributo trayecto de delegación contiene los trayectos de delegación, constituidos por una secuencia de certificados de atributo.

```
delegationPath      ATTRIBUTE   ::= {
    WITH SYNTAX                 AttCertPath
    ID                          id-at-delegationPath }

AttCertPath     ::=  SEQUENCE OF AttributeCertificate
```

Este atributo se puede almacenar en el asiento de directorio de AA y contendrá algunos trayectos de delegación desde las AA a otras AA. Este atributo, si se utiliza, permite la recuperación más eficiente de certificados de atributo delegados que forman trayectos de delegación frecuentemente utilizados. No existen requisitos específicos como tales para utilizar este atributo y es poco probable que el conjunto de valores que están almacenados en el atributo no representarán al conjunto completo de los trayectos de delegación para ninguna AA dada.

17.2.7 Atributo de política de privilegios

El atributo política de privilegios contiene información sobre políticas de privilegios.

```
privPolicy ATTRIBUTE  ::= {
    WITH SYNTAX       PolicySyntax
    ID                id-at-privPolicy }
```

El componente **policyIdentifier** incluye el identificador de objeto registrado para esa política de privilegios en particular.

Si **content** está presente, se incluye el contenido completo de la política de privilegios.

Si está presente **pointer**, el componente **name** hace referencia a una o más ubicaciones en las que se puede encontrar una copia de la política de privilegios. Si el componente **hash** está presente, contiene un TROCEO del contenido de la política de privilegios que se encontrará en una ubicación referenciada. Este troceo se puede utilizar para realizar una comprobación de integridad del documento referenciado.

17.2.8 Atributo de política de privilegios protegidos

El atributo de política de privilegios contiene políticas de privilegios, protegidos dentro de los certificados de atributos.

```
protPrivPolicy      ATTRIBUTE   ::= {
 WITH SYNTAX                    AttributeCertificate
 EQUALITY MATCHING RULE         attributeCertificateExactMatch
 ID                             id-at-protPrivPolicy }
```

Cabe señalar que a diferencia de los certificados de atributos convencionales, los que están dentro del atributo **protPrivPolicy** contienen políticas de privilegios, y no privilegios. Los componentes expedidor y titular de estos certificados de atributos permiten identificar la misma unidad. El atributo que se incluye en el certificado de atributo contenido en el atributo **protPrivPolicy** es el atributo **privPolicy** o el atributo **xmlPrivPolicy**.

17.2.9 Atributo de política de privilegios protegidos XML

El atributo de política de privilegios protegidos XML contiene información de política de privilegios codificada en XML.

```
xmlPrivPolicy      ATTRIBUTE   ::= {
 WITH SYNTAX       UTF8String -- contains XML-encoded privilege policy information
 ID                id-at-xMLPprotPrivPolicy }
```

17.3 Reglas de concordancia en el directorio PMI general

Esta subcláusula define las reglas de concordancia para atributos de directorio PMI.

17.3.1 Concordancia exacta de certificados de atributo

La regla de concordancia exacta de certificados de atributo compara la equivalencia de un valor presentado con un valor de atributo del tipo **AttributeCertificate**.

```
attributeCertificateExactMatch MATCHING-RULE  ::= {
    SYNTAX          AttributeCertificateExactAssertion
    ID              id-mr-attributeCertificateExactMatch }

AttributeCertificateExactAssertion      ::=  SEQUENCE {
    serialNumber            CertificateSerialNumber,
    issuer                  AttCertIssuer }
```

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

17.3.2 Concordancia de certificados de atributo

La regla de concordancia de certificados de atributo compara la equivalencia de un valor presentado con un valor de atributo del tipo **AttributeCertificate**. Estas reglas de concordancia permiten una concordancia mas compleja que en **certificateExactMatch**.

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

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

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

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

17.3.3 Concordancia de expedidores titulares

La regla de concordancia de expedidores titulares de certificado de atributo compara la equivalencia de un valor presentado de los componentes titular y/o expedidor del valor presentado con un valor de atributo del tipo **AttributeCertificate**.

```
holderIssuerMatch MATCHING-RULE  ::= {
    SYNTAX          HolderIssuerAssertion
    ID              id-mr-holderIssuerMatch }

HolderIssuerAssertion  ::=  SEQUENCE {
    holder          [0]     Holder          OPTIONAL,
    issuer          [1]     AttCertIssuer   OPTIONAL }
```

Esta regla de concordancia devuelve VERDADERO si todos los componentes que están presentes en el valor presentado concuerdan con los componentes correspondientes en el valor de atributo.

17.3.4 Concordancia de trayectos de delegación

La regla de concordancia **delegationPathMatch** compara la equivalencia de un valor presentado con un valor de atributo del tipo **delegationPath**. Un verificador de privilegios puede utilizar esta regla de concordancia para seleccionar un trayecto que empieza con un certificado expedido por una SOA y que termina con un certificado expedido a una AA que expidió el certificado de titular de entidad final que se está validando.

```
delegationPathMatch MATCHING-RULE  ::= {
    SYNTAX          DelMatchSyntax
    ID              id-mr-delegationPathMatch }

DelMatchSyntax      ::=  SEQUENCE {
    firstIssuer     AttCertIssuer,
    lastHolder      Holder }
```

Esta regla de concordancia devuelve VERDADERO si el valor presentado en el componente **firstIssuer** concuerda con los elementos correspondientes del campo expedidor del primer certificado en la **SECUENCIA** en el valor almacenado, y el valor presentado en el componente **lastHolder** concuerda con los elementos correspondientes del campo titular del último certificado en la **SECUENCIA** en el valor almacenado. Esta regla de concordancia devuelve **FALSO** si cualquiera de las concordancias fracasa.

SECCIÓN 4 – UTILIZACIÓN POR EL DIRECTORIO DE LOS MARCOS DE CERTIFICADOS DE CLAVE PÚBLICA Y DE ATRIBUTO

El directorio utiliza el marco de certificados de clave pública como base para varios servicios de seguridad, incluidas la autenticación fuerte y la protección de operaciones de directorio así como la protección de datos almacenados. El directorio utiliza el marco de certificados de atributo como base para el esquema de control de accesos basado en reglas. Aquí se define la relación entre los elementos del marco de certificados de clave pública y del marco de certificados de atributo con los diversos servicios de seguridad del directorio. Los servicios de seguridad específicos que proporciona el directorio se especifican totalmente mediante el conjunto completo de especificaciones de directorio.

18 Autenticación de directorio

El directorio soporta la autenticación de usuarios que acceden al directorio a través de un DUA y la autenticación de sistemas de directorio (DSA, *directory system agent*) a usuarios y a otros DSA. En función del entorno, se puede utilizar autenticación simple o fuerte. En las siguientes subcláusulas se describen los procedimientos a utilizar para la autenticación simple y fuerte en el directorio.

18.1 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:

a) la transferencia del nombre distinguido del usuario y la contraseña (opcional) en lenguaje ordinario (no protegido) al receptor, para su evaluación;

b) 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;

c) 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 este documento.

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 5.

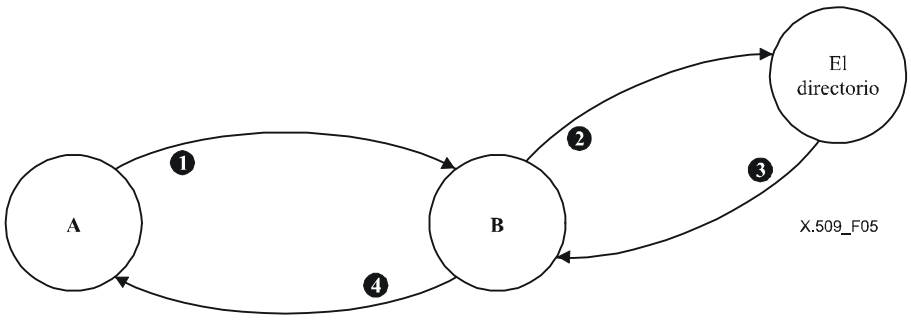


Figura 5 – Procedimiento de autenticación simple no protegida

Comprende los siguientes pasos:

1) un usuario originador A envía su nombre distinguido y contraseña a un usuario receptor (o destinatario) B;

2) 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);

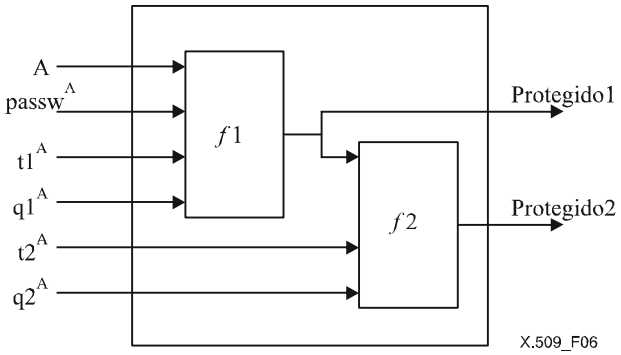
3) el directorio confirma (o rechaza) a B que las credenciales son válidas;

4) 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).

18.1.1 Generación de información de identificación protegida

La figura 6 muestra dos métodos que pueden emplearse para generar información de identificación protegida. $f1$ y $f2$ 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.



A	Nombre distinguido de usuario
t^A	Indicaciones de tiempo
$passw^A$	Contraseña de A
q^A	Números aleatorios, y opcionalmente con contador incluido

Figura 6 – Autenticación simple protegida

18.1.2 Procedimiento de autenticación simple protegida

La figura 7 ilustra el procedimiento de autenticación simple protegida.

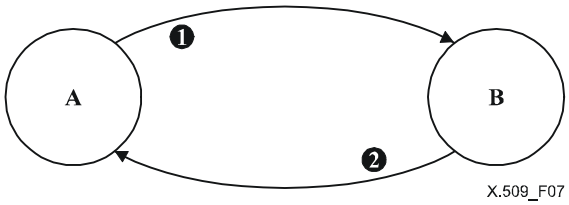


Figura 7 – El procedimiento de autenticación simple protegida

Comprende las siguientes frases (inicialmente utilizando solamente $f1$):

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 ($f1$) de la figura 6, 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:

   $$\text{Protegido1} = f1\ (t1^A, q1^A, A, passw^A)$$

   La información transportada a B tiene la forma siguiente:

   $$\text{Autenticador1} = t1^A, q1^A, A, \text{protegido1}$$

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 $f1$ y $f2$. Las diferencias principales son las siguientes:

1) A envía adicionalmente su información de identificación protegida (autenticador2) a B. Una protección adicional se obtiene aplicando una segunda función unidireccional, $f2$, como se ilustra en la figura 6. Esta mayor protección adopta la forma siguiente:

   Protegido2 = $f2$ ($t2^A$, $q2^A$, protegido1)

   La información transportada a B tiene la forma:

   Autenticador2 = $t1^A$, $t2^A$, $q1^A$, $q2^A$, A, 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.

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

NOTA – Los procedimientos definidos en estas cláusulas 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; alternativamente A, podría ser un DSA vinculado a otro DSA, B.

18.1.3 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 }
```

18.2 Autenticación fuerte

Los procedimientos descritos en esta subcláusula se utilizan en la autenticación entre un DUA y un DSA, así como entre pares de DSA. Los procedimientos utilizan el marco de certificados de clave pública definido en esta Especificación. Además, los procedimientos utilizan el propio directorio como depósito de la información de clave pública necesaria para realizar la autenticación. La inclusión de parámetros pertinentes en los protocolos de directorio está definida en las propias especificaciones de protocolo. Los procedimientos definidos aquí para la autenticación fuerte pueden también ser utilizados para aplicaciones distintas de las del directorio, que también utilizan este tipo de depósitos. Para la utilización por el directorio de estos procedimientos, el término 'usuario' en estos procedimientos puede referirse tanto a un DUA como a un DSA.

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 E 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 tendrán 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.

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 copartícipe en la comunicación está en posesión de la clave secreta, y puede usar esto para corroborar que su copartícipe 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 copartícipe 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 subcláusula 18.2.1 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.

18.2.1 Obtención de certificados de clave pública a partir del directorio

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 está definida en 11.2.

En el caso general, antes de que los usuarios puedan autenticar mutuamente, el directorio tendrá 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 un ejemplar particular de autenticación se puede reducir por los medios siguientes:

a) si los dos 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;

b) 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;

c) 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;

d) las autoridades de certificación pueden certificarse mutuamente unas a otras, por acuerdos bilaterales. Como resultado de esto se acorta el trayecto de certificación;

e) 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.

18.2.1.1 Ejemplo

La figura 8 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.

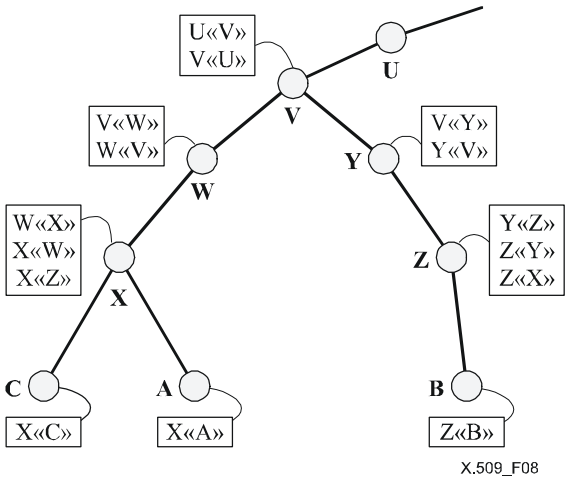


Figura 8 – Jerarquía de CA – Ejemplo teórico

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<<W>>, W<<V>>, V<<Y>>, Y<<Z>>, Z<<B>>

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:

Bp = Xp • X<<W>> W<<V>> V<<Y>> Y<<Z>> Z<<B>>

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<<Y>>, Y<<V>>, V<<W>>, W<<X>>, X<<A>>

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:

Ap = Zp • Z<<Y>> Y<<V>> V<<W>> W<<X>> X<<A>>

Aplicando las optimizaciones de 18.2.1:

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

   Cp = Xp • X<<C>>

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

   Ap = Xp • X<<A>>

b) suponiendo que A conociera así W<<X>>, Wp, V<<W>>, Vp, U<<V>>, hacia arriba, etc., la información que A tendrá que obtener del directorio para formar el trayecto de autenticación se reduce a:

   V<<Y>>, Y<<Z>>, Z<<B>>

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

   Z<<Y>>, Y<<V>>

c) suponiendo que A comunica frecuentemente con usuarios certificados por Z, él puede aprender (además de las claves públicas aprendidas en b)) V<<Y>>, Y<<V>>, Y<<Z>> y Z<<Y>>. Para comunicar con B, sólo necesita por consiguiente obtener Z<<B>> del directorio;

d) suponiendo que los usuarios certificados por X y Z comunican frecuentemente, entonces X<<Z>> estaría contenido en la inserción de directorio para X, y viceversa (esto se muestra en la figura 8). Si A quiere autenticar hacia B, sólo necesita obtener:

X<<Z>>, Z<<B>>

para formar el trayecto de certificación, y:

Z<<X>>

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:

Cp = Xp • X<<C>>

y

Ap = Xp • X<<A>>

En el caso más general, las autoridades de certificación no guardan una relación jerárquica. En el ejemplo hipotético de la figura 9, 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<<D>> y la inserción del usuario E contendrá el certificado W<<E>>.

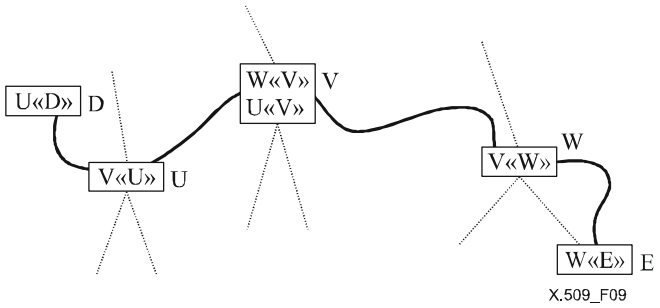


Figura 9 – Ejemplo de trayecto de certificación no jerárquico

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<<V>>, V<<U>>, W<<V>> y V<<W>>. Supóngase que U<<V>> y W<<V>> están almacenados en la inserción de V, V<<U>> está almacenado en el asiento de U, y V"W" está almacenado en la inserción de W.

El usuario D necesita 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<<V>>, V<<W>>, W<<E>>. Una vez descubierto este trayecto, se puede construir también el trayecto inverso W<<V>>, V<<U>>, U<<D>>.

18.2.2 Procedimientos de autenticación fuerte

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 subcláusula 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 normas 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 18.2.2.1 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 18.2.2.2, 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 18.2.2.3, 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 tendrá 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 18.2 anteriormente. Tal tipo de acceso no se vuelve a mencionar en la descripción de los procedimientos que siguen.

La comprobación de las indicaciones de tiempo mencionadas en las siguientes cláusulas 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.

18.2.2.1 Autenticación unidireccional

Se siguen los pasos indicados en la figura 10:

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:

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

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:

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

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"):

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

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:
a) obtiene Ap de BA, comprobando que el certificado de A no ha expirado;
b) verifica la firma, y por consiguiente la integridad de la información firmada;
c) comprueba que el mismo B es el receptor deseado;
d) comprueba que la indicación de tiempo está "actual";
e) opcionalmente, comprueba que r^A no ha sido reactuada. 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 tiempo t^A en lenguaje ordinario junto con la parte del testigo a que se aplicó la función hash, durante el rango de tiempo r^A.

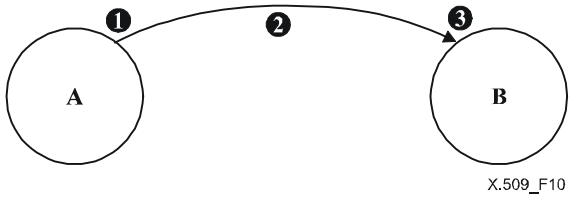


Figura 10 – Autenticación unidireccional

18.2.2.2 Autenticación bidireccional

Se siguen los pasos indicados en la figura 11:

1) como en 18.2.2.1;
2) como en 18.2.2.1;
3) como en 18.2.2.1;
4) B genera r^B, un número no repetitivo, utilizado para fines similares a los de r^A;
5) 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:

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

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"):

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

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 tiempo t^B es "corriente";
   d) opcionalmente, comprueba que r^B no ha sido reactuado (véase 18.2.2.1, paso 3) d)).

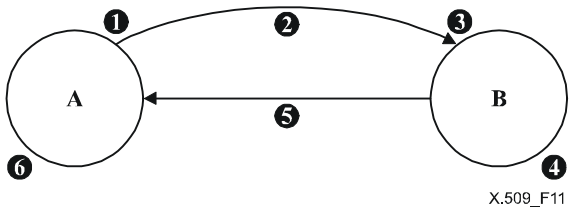


Figura 11 – Autenticación bidireccional

18.2.2.3 Autenticación tridireccional

Se siguen los pasos indicados en la figura 12:

1) Como en 18.2.2.2.
2) Como en 18.2.2.2. La indicación de tiempo t^A puede ser cero.
3) Como en 18.2.2.2, excepto que la indicación de tiempo no necesita ser comprobada.
4) Como en 18.2.2.2.
5) Como en 18.2.2.2. La indicación de tiempo t^B puede ser cero.
6) Como en 18.2.2.2, 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.

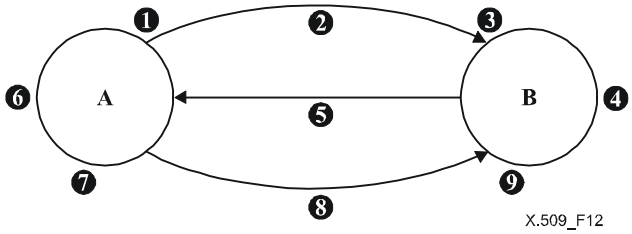


Figura 12 – Autenticación tridireccional

19 Control de acceso

El directorio existe en un entorno en el que diversas autoridades administrativas controlan el acceso a su porción de la DIB. La definición de un esquema de control de acceso en el contexto del directorio incluye métodos para:

- especificar la información de control de acceso (ACI, *access control information*);
- imponer los derechos de acceso definidos por esa información de control de acceso;
- mantener la información de control de acceso.

La imposición de los derechos de acceso aplica al control de acceso para:

- información de directorio relativa a nombres;
- información de usuario de directorio;
- información de explotación de directorio, incluida información de control de acceso.

Las autoridades administrativas utilizan todo o partes de un esquema de control de acceso normalizado al implementar sus políticas de seguridad, o pueden definir libremente sus propios esquemas a discreción.

El esquema de control de acceso básico (BAC, *basic access control*) definido en la Rec. UIT-T X.501 | ISO/CEI 9594-2 es un esquema basado en una lista de control de acceso que permite a los administradores de directorio establecer permisos al nivel de autenticación realizado para vincular al directorio. El marco de certificados de clave pública definido en esta Especificación se utiliza para proporcionar el esquema de autenticación fuerte utilizado para esta vinculación.

El esquema de control de acceso basado en reglas (RBAC, *rules based access control*) definidas en la Rec. UIT-T X.501 | ISO/CEI 9594-2, utiliza el marco de certificados de atributo definido en esta Especificación para transportar atributos de liberación utilizados al tomar decisiones de control de acceso. El RBAC también puede utilizarse junto con el BAC.

20 Protección de operaciones de directorio

El marco de certificados de clave pública definido en esta Especificación se utiliza en todos los protocolos de directorio definidos en esta serie de Recomendaciones para proteger de forma optativa las operaciones que incluyen peticiones, respuestas y errores. La protección de integridad se proporciona mediante la firma digital del emisor y la verificación de esa firma por el receptor, utilizando el certificado de clave pública correspondiente al emisor. La protección de privacidad se proporciona mediante la utilización de criptado de clave pública en la que se cripta el contenido con la clave pública obtenida del pretendido certificado de clave pública del receptor y se descripta por el receptor utilizando su correspondiente clave privada.

Los mecanismos y sintaxis específicos para solicitar e incluir los elementos de protección en los intercambios de protocolo se definen dentro de cada protocolo de directorio en esta serie de Especificaciones.

Anexo A

Marcos para certificados de claves públicos y atributos

(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**, **CertificateExtensions** y **AttributeCertificateDefinitions**.

-- A.1 Authentication framework module

```
AuthenticationFramework {joint-iso-itu-t ds(5) module(1) authenticationFramework(7) 5}
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, id-nf, id-oc, informationFramework, upperBounds, selectedAttributeTypes, basicAccessControl,
    certificateExtensions
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 5}

    Name, ATTRIBUTE, OBJECT-CLASS, NAME-FORM, top
        FROM InformationFramework informationFramework

    ub-user-password, ub-content
        FROM UpperBounds upperBounds

    UniqueIdentifier, octetStringMatch, DirectoryString{}, commonName
        FROM SelectedAttributeTypes selectedAttributeTypes

    certificateExactMatch, certificatePairExactMatch, certificateListExactMatch, KeyUsage, GeneralNames,

       CertificatePoliciesSyntax, algorithmIdentifierMatch, CertPolicyId
        FROM CertificateExtensions certificateExtensions ;

-- public-key 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 shall be v2 or v3
    subjectUniqueIdentifier [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                                 -- if present, version shall be v2 or v3
    extensions              [3]  Extensions OPTIONAL
                                 -- If present, version shall 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 PKI 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 -- }
    (WITH COMPONENTS { ..., forward PRESENT} |
    WITH COMPONENTS { ..., reverse PRESENT})

-- certificate revocation list (CRL)

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

```
-- information object classes --

ALGORITHM ::=  TYPE-IDENTIFIER

-- parameterized types --
HASH {ToBeHashed}       ::=     SEQUENCE {
    algorithmIdentifier             AlgorithmIdentifier,
    hashValue                       BIT STRING ( CONSTRAINED BY {
        -- shall be the result of applying a hashing procedure to the DER-encoded octets --

        -- of a value of --ToBeHashed } ) }

ENCRYPTED-HASH { ToBeSigned }       ::=     BIT STRING ( CONSTRAINED BY {
    -- shall be the result of applying a hashing procedure to the DER-encoded (see 6.1) octets --
    -- of a value of -- ToBeSigned -- and then applying an encipherment procedure to those octets -- })

ENCRYPTED { ToBeEnciphered }        ::=     BIT STRING ( CONSTRAINED BY {
    -- shall 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 }}

-- PKI object classes --

pkiUser  OBJECT-CLASS       ::= {
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {userCertificate}
    ID              id-oc-pkiUser }


pkiCA OBJECT-CLASS          ::= {
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {cACertificate |
                    certificateRevocationList |
                    authorityRevocationList |
                    crossCertificatePair }
    ID              id-oc-pkiCA }


cRLDistributionPoint        OBJECT-CLASS        ::= {
    SUBCLASS OF             { top }
    KIND                    structural
    MUST CONTAIN            { commonName }
    MAY CONTAIN             { certificateRevocationList |
                            authorityRevocationList |
                            deltaRevocationList }
    ID                      id-oc-cRLDistributionPoint }


cRLDistPtNameForm  NAME-FORM        ::= {
    NAMES                   cRLDistributionPoint
    WITH ATTRIBUTES         { commonName}
    ID                      id-nf-cRLDistPtNameForm }


deltaCRL    OBJECT-CLASS        ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         {deltaRevocationList}
    ID                  id-oc-deltaCRL }
```

```
cpCps    OBJECT-CLASS        ::= {
    SUBCLASS OF             {top}
    KIND                    auxiliary
    MAY CONTAIN             {certificatePolicy |
                            certificationPracticeStmt}
    ID                      id-oc-cpCps }

pkiCertPath         OBJECT-CLASS        ::= {
    SUBCLASS OF             {top}
    KIND                    auxiliary
    MAY CONTAIN             { pkiPath }
    ID                      id-oc-pkiCertPath }
```

-- PKI directory attributes --

```
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 }

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 }

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

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 }

certificationPracticeStmt ATTRIBUTE    ::=         {
    WITH SYNTAX             InfoSyntax
    ID                      id-at-certificationPracticeStmt }
```

```
InfoSyntax    ::=  CHOICE {
    content         DirectoryString {ub-content},
    pointer         SEQUENCE {
        name                GeneralNames,
        hash                HASH { HashedPolicyInfo } OPTIONAL } }

POLICY  ::=  TYPE-IDENTIFIER

HashedPolicyInfo      ::=    POLICY.&Type( {Policies} )

Policies POLICY ::= {...} -- Defined by implementors --


certificatePolicy   ATTRIBUTE   ::=  {
    WITH SYNTAX       PolicySyntax
    ID                id-at-certificatePolicy }


PolicySyntax ::=  SEQUENCE {
    policyIdentifier      PolicyID,
    policySyntax          InfoSyntax
    }


PolicyID      ::=  CertPolicyId


pkiPath       ATTRIBUTE   ::= {
    WITH SYNTAX       PkiPath
    ID                id-at-pkiPath }


PkiPath       ::=  SEQUENCE OF Certificate


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

-- object identifier assignments --
-- object classes --

id-oc-cRLDistributionPoint          OBJECT IDENTIFIER ::=          {id-oc 19}
id-oc-pkiUser                       OBJECT IDENTIFIER ::=          {id-oc 21}
id-oc-pkiCA                         OBJECT IDENTIFIER ::=          {id-oc 22}
id-oc-deltaCRL                      OBJECT IDENTIFIER ::=          {id-oc 23}
id-oc-cpCps                         OBJECT IDENTIFIER ::=          {id-oc 30}
id-oc-pkiCertPath                   OBJECT IDENTIFIER ::=          {id-oc 31}

-- name forms--

id-nf-cRLDistPtNameForm             OBJECT IDENTIFIER ::=          {id-nf 14}

-- directory attributes--

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-supportedAlgorithms           OBJECT IDENTIFIER ::=          {id-at 52}
id-at-deltaRevocationList           OBJECT IDENTIFIER ::=          {id-at 53}
id-at-certificationPracticeStmt     OBJECT IDENTIFIER ::=          {id-at 68}
id-at-certificatePolicy             OBJECT IDENTIFIER ::=          {id-at 69}
id-at-pkiPath                       OBJECT IDENTIFIER ::=          {id-at 70}
```

```
END
```

-- A.2 Certificate extensions module

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

-- EXPORTS ALL --

IMPORTS
    id-at, id-ce, id-mr, informationFramework, authenticationFramework,
        selectedAttributeTypes, upperBounds
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1)
        usefulDefinitions(0) 5}

    Name, RelativeDistinguishedName, ATTRIBUTE, Attribute, MATCHING-RULE
        FROM InformationFramework informationFramework

    CertificateSerialNumber, CertificateList, AlgorithmIdentifier,
        EXTENSION, Time, PolicyID
        FROM AuthenticationFramework authenticationFramework

    DirectoryString {}
        FROM SelectedAttributeTypes selectedAttributeTypes

    ub-name
        FROM UpperBounds upperBounds

    ORAddress
        FROM MTSAbstractService {joint-iso-itu-t mhs(6) mts(3)
        modules(0) mts-abstract-service(1) version-1999 (1) } ;
-- Unless explicitly noted otherwise, there is no significance to the ordering
-- of components of a SEQUENCE OF construct in this Specification.

-- public-key certificate and CRL 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),
    contentCommitment       (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 ::= { ... }

anyPolicy      OBJECT IDENTIFIER    ::=   { 2 5 29 32 0 }

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 }


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


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-nameConstraint }

NameConstraintsSyntax ::= SEQUENCE {
    permittedSubtrees       [0]  GeneralSubtrees OPTIONAL,
    excludedSubtrees        [1]  GeneralSubtrees OPTIONAL,
    requiredNameForms       [2]  NameForms 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)


NameForms  ::= SEQUENCE {
   basicNameForms    [0]  BasicNameForms OPTIONAL,
   otherNameForms    [1]  SEQUENCE SIZE (1..MAX) OF OBJECT IDENTIFIER OPTIONAL }
(ALL EXCEPT ({ -- none; i.e., at least one component shall be present -- }))


BasicNameForms  ::= BIT STRING {
    rfc822Name              (0),
    dNSName             (1),
    x400Address         (2),
    directoryName       (3),
    ediPartyName        (4),
    uniformResourceIdentifier    (5),
    iPAddress           (6),
    registeredID            (7) }  (SIZE (1..MAX))


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

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

SkipCerts ::= INTEGER (0..MAX)


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),
    privilegeWithdrawn            (9),
    aaCompromise                  (10) }


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

HoldInstruction ::= OBJECT IDENTIFIER


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


crlScope EXTENSION ::= {
    SYNTAX                    CRLScopeSyntax
    IDENTIFIED BY             id-ce-cRLScope }

CRLScopeSyntax ::=            SEQUENCE SIZE (1..MAX) OF PerAuthorityScope

PerAuthorityScope ::= SEQUENCE {
    authorityName             [0]     GeneralName OPTIONAL,
    distributionPoint         [1]     DistributionPointName OPTIONAL,
    onlyContains              [2]     OnlyCertificateTypes OPTIONAL,
    onlySomeReasons           [4]     ReasonFlags OPTIONAL,
    serialNumberRange         [5]     NumberRange OPTIONAL,
    subjectKeyIdRange         [6]     NumberRange OPTIONAL,
    nameSubtrees              [7]     GeneralNames OPTIONAL,
    baseRevocationInfo        [9]     BaseRevocationInfo OPTIONAL
    }

OnlyCertificateTypes ::= BIT STRING {
    user          (0),
    authority     (1),
    attribute     (2) }

NumberRange ::= SEQUENCE {
    startingNumber    [0]     INTEGER OPTIONAL,
    endingNumber      [1]     INTEGER OPTIONAL,
    modulus                   INTEGER OPTIONAL }

BaseRevocationInfo ::= SEQUENCE {
    cRLStreamIdentifier       [0]     CRLStreamIdentifier    OPTIONAL,
    cRLNumber                 [1]     CRLNumber,
    baseThisUpdate            [2]     GeneralizedTime }
```

```
statusReferrals EXTENSION  ::= {
    SYNTAX              StatusReferrals
    IDENTIFIED BY       id-ce-statusReferrals }

StatusReferrals ::= SEQUENCE SIZE (1..MAX) OF StatusReferral

StatusReferral  ::=  CHOICE {
    cRLReferral         [0]     CRLReferral,
    otherReferral       [1]     INSTANCE OF OTHER-REFERRAL}

CRLReferral ::= SEQUENCE  {
    issuer              [0]     GeneralName OPTIONAL,
    location            [1]     GeneralName OPTIONAL,
    deltaRefInfo        [2]     DeltaRefInfo OPTIONAL,
    cRLScope                    CRLScopeSyntax,
    lastUpdate          [3]     GeneralizedTime OPTIONAL,
    lastChangedCRL      [4]     GeneralizedTime OPTIONAL}

DeltaRefInfo  ::=  SEQUENCE {
    deltaLocation       GeneralName,
    lastDelta           GeneralizedTime OPTIONAL }

OTHER-REFERRAL ::= TYPE-IDENTIFIER


cRLStreamIdentifier EXTENSION       ::= {
    SYNTAX              CRLStreamIdentifier
    IDENTIFIED BY       id-ce-cRLStreamIdentifier }


CRLStreamIdentifier  ::=  INTEGER (0..MAX)


orderedList EXTENSION      ::= {
    SYNTAX              OrderedListSyntax
    IDENTIFIED BY       id-ce-orderedList }

OrderedListSyntax  ::= ENUMERATED {
ascSerialNum        (0),
ascRevDate          (1) }


deltaInfo EXTENSION  ::=   {
    SYNTAX              DeltaInformation
    IDENTIFIED BY       id-ce-deltaInfo }

DeltaInformation   ::=   SEQUENCE {
    deltaLocation       GeneralName,
    nextDelta           GeneralizedTime OPTIONAL }


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

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

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

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

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

```
    privilegeWithdrawn      (7),
    aACompromise            (8) }

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


IssuingDistPointSyntax ::= SEQUENCE {
    -- If onlyContainsUserPublicKeyCerts and onlyContainsCACerts are both FALSE,
    -- the CRL covers both certificate types
    distributionPoint                   [0] DistributionPointName OPTIONAL,
    onlyContainsUserPublicKeyCerts      [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

toBeRevoked  EXTENSION  ::= {
    SYNTAX   ToBeRevokedSyntax
    IDENTIFIED BYid-ce-toBeRevoked }

ToBeRevokedSyntax::=  SEQUENCE SIZE(1..MAX) OF ToBeRevokedGroup

ToBeRevokedGroup      ::=       SEQUENCE {
    certificateIssuer       [0]       GeneralName OPTIONAL,
    reasonInfo              [1]       ReasonInfo     OPTIONAL,
    revocationTime                    GeneralizedTime,
    certificateGroup                  CertificateGroup }

ReasonInfo          ::=     SEQUENCE {
    reasonCode                        CRLReason,
    holdInstructionCode     HoldInstruction OPTIONAL }

CertificateGroup   ::=      CHOICE {
    serialNumbers           [0]       CertificateSerialNumbers,
    serialNumberRange       [1]       CertificateGroupNumberRange,
    nameSubtree             [2]       GeneralName }

CertificateGroupNumberRange ::= SEQUENCE {
    startingNumber          [0]       INTEGER,
    endingNumber            [1]       INTEGER }

CertificateSerialNumbers ::=      SEQUENCE SIZE(1..MAX) OF CertificateSerialNumber

revokedGroups  EXTENSION  ::= {
    SYNTAX   RevokedGroupsSyntax
    IDENTIFIED BYid-ce-RevokedGroups }

RevokedGroupsSyntax ::=           SEQUENCE SIZE (1..MAX) OF RevokedGroup

RevokedGroup ::=SEQUENCE {
    certificateIssuer               [0]     GeneralName OPTIONAL,
    reasonInfo                      [1]     ReasonInfo OPTIONAL,
    invalidityDate                  [2]     GeneralizedTime OPTIONAL,
    revokedcertificateGroup         [3]     RevokedCertificateGroup }

RevokedCertificateGroup ::=       CHOICE {
    serialNumberRange   NumberRange,
    nameSubtree         GeneralName }
```

```
expiredCertsOnCRL EXTENSION  ::= {
    SYNTAXExpiredCertsOnCRL
    IDENTIFIED BYid-ce-expiredCertsOnCRL }

ExpiredCertsOnCRL       ::= GeneralizedTime

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


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


aAissuingDistributionPoint EXTENSION ::= {
    SYNTAX AAIssuingDistPointSyntax
    IDENTIFIED BY id-ce-aAissuingDistributionPoint }

AAIssuingDistPointSyntax ::= SEQUENCE {
    distributionPoint                   [ 0 ] DistributionPointName OPTIONAL,
    onlySomeReasons                     [ 1 ] ReasonFlags OPTIONAL,
    indirectCRL                         [ 2 ] BOOLEAN DEFAULT FALSE,
    containsUserAttributeCerts          [ 3 ] BOOLEAN DEFAULT TRUE,
    containsAACerts                     [ 4 ] BOOLEAN DEFAULT TRUE,
    containsSOAPublicKeyCerts           [ 5 ] BOOLEAN DEFAULT TRUE }


inhibitAnyPolicy   EXTENSION   ::= {
    SYNTAX         SkipCerts
IDENTIFIED BY      id-ce-inhibitAnyPolicy }
```

-- PKI 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,
    subject                 [11] Name                       OPTIONAL,
    nameConstraints         [12] NameConstraintsSyntax      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 SIZE (1..MAX) OF CertPolicyId


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

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


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

CertificatePairAssertion ::= SEQUENCE {
    issuedToThisCAAssertion    [0] CertificateAssertion OPTIONAL,
    issuedByThisCAAssertion    [1] CertificateAssertion OPTIONAL }
    ( WITH COMPONENTS   {..., issuedToThisCAAssertion PRESENT} |
      WITH COMPONENTS   {..., issuedByThisCAAssertion 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,
    authorityKeyIdentifier  [3]     AuthorityKeyIdentifier OPTIONAL}


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


policyMatch MATCHING-RULE   ::= {
    SYNTAX          PolicyID
    ID              id-mr-policyMatch }


pkiPathMatch MATCHING-RULE  ::= {
SYNTAX      PkiPathMatchSyntax
ID          id-mr-pkiPathMatch }

PkiPathMatchSyntax      ::= SEQUENCE {
firstIssuer     Name,
lastSubject     Name }


enhancedCertificateMatch MATCHING-RULE  ::= {
    SYNTAX  EnhancedCertificateAssertion
    ID      id-mr-enhancedCertificateMatch }
```

```
EnhancedCertificateAssertion  ::= 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]  AltName                    OPTIONAL,
    policy                  [9]  CertPolicySet              OPTIONAL,
    pathToName              [10] GeneralNames               OPTIONAL,
    subject                 [11] Name                       OPTIONAL,
    nameConstraints         [12] NameConstraintsSyntax      OPTIONAL
}

(ALL EXCEPT ({ -- none; at least one component shall be present -- }))

AltName  ::= SEQUENCE {
        altnameType         AltNameType,
        altNameValue        GeneralName OPTIONAL }

-- Object identifier assignments --

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-nameConstraint                    OBJECT IDENTIFIER  ::=  {id-ce 30 1}

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-ce-cRLStreamIdentifier               OBJECT IDENTIFIER  ::=  {id-ce 40}
id-ce-cRLScope                          OBJECT IDENTIFIER  ::=  {id-ce 44}
id-ce-statusReferrals                   OBJECT IDENTIFIER  ::=  {id-ce 45}
id-ce-freshestCRL                       OBJECT IDENTIFIER  ::=  {id-ce 46}
id-ce-orderedList                       OBJECT IDENTIFIER  ::=  {id-ce 47}
id-ce-baseUpdateTime                    OBJECT IDENTIFIER  ::=  {id-ce 51}
id-ce-deltaInfo                         OBJECT IDENTIFIER  ::=  {id-ce 53}
id-ce-inhibitAnyPolicy                  OBJECT IDENTIFIER  ::=  {id-ce 54}
id-ce-toBeRevoked                       OBJECT IDENTIFIER  ::=  {id-ce 58}
id-ce-RevokedGroups                     OBJECT IDENTIFIER  ::=  {id-ce 59}
id-ce-expiredCertsOnCRL                 OBJECT IDENTIFIER  ::=  {id-ce 60}
id-ce-aAissuingDistributionPoint        OBJECT IDENTIFIER  ::=  {id-ce 63}

-- matching rule OIDs --

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}
id-mr-policyMatch                       OBJECT IDENTIFIER   ::=  {id-mr 60}
id-mr-pkiPathMatch                      OBJECT IDENTIFIER   ::=  {id-mr 62}
id-mr-enhancedCertificateMatch          OBJECT IDENTIFIER   ::=  {id-mr 65}

-- 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}, {id-ce 30}

END
```

-- A.3 Attribute Certificate Framework module

```
AttributeCertificateDefinitions {joint-iso-itu-t ds(5) module(1) attributeCertificateDefinitions(32) 5}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- EXPORTS ALL --

IMPORTS
    id-at, id-ce, id-mr, informationFramework, authenticationFramework,
        selectedAttributeTypes, upperBounds, id-oc, certificateExtensions
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1)
        usefulDefinitions(0) 5}


    Name, RelativeDistinguishedName, ATTRIBUTE, Attribute,
        MATCHING-RULE, AttributeType, OBJECT-CLASS, top
        FROM InformationFramework informationFramework

    CertificateSerialNumber, CertificateList, AlgorithmIdentifier,
        EXTENSION, SIGNED{}, InfoSyntax, PolicySyntax, Extensions, Certificate
        FROM AuthenticationFramework authenticationFramework

    DirectoryString{}, TimeSpecification, UniqueIdentifier
        FROM SelectedAttributeTypes selectedAttributeTypes

    GeneralName, GeneralNames, NameConstraintsSyntax, certificateListExactMatch
       FROM CertificateExtensions certificateExtensions

    ub-name
        FROM UpperBounds upperBounds

    UserNotice
        FROM PKIX1Implicit93 {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5)
pkix(7) id-mod(0) id-pkix1-implicit-93(4)}

    ORAddress
        FROM MTSAbstractService {joint-iso-itu-t mhs(6) mts(3)
        modules(0) mts-abstract-service(1) version-1999 (1) } ;
-- Unless explicitly noted otherwise, there is no significance to the ordering
-- of components of a SEQUENCE OF construct in this Specification.

-- attribute certificate constructs --


AttributeCertificate ::= SIGNED {AttributeCertificateInfo}

AttributeCertificateInfo ::= SEQUENCE
    {
    version                 AttCertVersion, -- version is v2
    holder                  Holder,
    issuer                  AttCertIssuer,
    signature               AlgorithmIdentifier,
    serialNumber            CertificateSerialNumber,
    attrCertValidityPeriod  AttCertValidityPeriod,
    attributes              SEQUENCE OF Attribute,
    issuerUniqueID          UniqueIdentifier    OPTIONAL,
    extensions              Extensions          OPTIONAL
    }
```

```
AttCertVersion     ::= INTEGER { v2(1) }

Holder    ::=   SEQUENCE
          {
          baseCertificateID       [0] IssuerSerial              OPTIONAL,
               -- the issuer and serial number of  the holder's Public Key Certificate
          entityName              [1] GeneralNames              OPTIONAL,
               -- the name of the entity or role
          objectDigestInfo        [2] ObjectDigestInfo          OPTIONAL
               -- used to directly authenticate the holder, e.g., an executable
-- at least one of baseCertificateID, entityName or objectDigestInfo shall be present --}


ObjectDigestInfo    ::= SEQUENCE {
    digestedObjectType  ENUMERATED {
        publicKey            (0),
        publicKeyCert        (1),
        otherObjectTypes     (2) },
    otherObjectTypeID       OBJECT IDENTIFIER  OPTIONAL,
    digestAlgorithm         AlgorithmIdentifier,
    objectDigest            BIT STRING }

AttCertIssuer ::=   [0]     SEQUENCE {
    issuerName                    GeneralNames  OPTIONAL,
    baseCertificateID       [0]   IssuerSerial  OPTIONAL,
    objectDigestInfo        [1]   ObjectDigestInfo  OPTIONAL }


-- At least one component shall be present
    ( WITH COMPONENTS { ..., issuerName  PRESENT } |
      WITH COMPONENTS { ..., baseCertificateID  PRESENT } |
      WITH COMPONENTS { ..., objectDigestInfo PRESENT } )


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

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

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

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


PrivilegePolicy     ::=   OBJECT IDENTIFIER

-- privilege attributes --


role    ATTRIBUTE::= {
        WITH SYNTAX         RoleSyntax
        ID                  id-at-role }


xmlPrivilegeInfo    ATTRIBUTE  ::= {
    WITH SYNTAX UTF8String --contains XML-encoded privilege information
    ID              id-at-xMLPrivilegeInfo }


RoleSyntax    ::=   SEQUENCE {
    roleAuthority       [0]       GeneralNames OPTIONAL,
    roleName            [1]       GeneralName }
```

```
-- PMI object classes --

pmiUser OBJECT-CLASS ::= {
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {attributeCertificateAttribute}
    ID              id-oc-pmiUser
    }

pmiAA OBJECT-CLASS ::= {
-- a PMI AA
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {aACertificate |
                    attributeCertificateRevocationList |
                    attributeAuthorityRevocationList}
    ID              id-oc-pmiAA
    }


pmiSOA OBJECT-CLASS ::= {  -- a PMI Source of Authority
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {attributeCertificateRevocationList |
                    attributeAuthorityRevocationList |
                    attributeDescriptorCertificate}
    ID              id-oc-pmiSOA
    }

attCertCRLDistributionPt    OBJECT-CLASS ::= {
    SUBCLASS OF             {top}
    KIND                    auxiliary
    MAY CONTAIN             { attributeCertificateRevocationList |
                            attributeAuthorityRevocationList }
    ID                      id-oc-attCertCRLDistributionPts
    }

pmiDelegationPath           OBJECT-CLASS   ::= {
    SUBCLASS OF             {top}
    KIND                    auxiliary
    MAY CONTAIN             { delegationPath }
    ID                      id-oc-pmiDelegationPath }

privilegePolicy             OBJECT-CLASS   ::= {
    SUBCLASS OF             {top}
    KIND                    auxiliary
    MAY CONTAIN             {privPolicy }
    ID                      id-oc-privilegePolicy }

protectedPrivilegePolicy    OBJECT-CLASS   ::= {
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {protPrivPolicy }
    ID              id-oc-protectedPrivilegePolicy }

-- PMI directory attributes --

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

aACertificate                       ATTRIBUTE   ::=  {
    WITH SYNTAX                     AttributeCertificate
    EQUALITY MATCHING RULE          attributeCertificateExactMatch
    ID                              id-at-aACertificate }
```

```
attributeDescriptorCertificate      ATTRIBUTE  ::= {
    WITH SYNTAX                     AttributeCertificate
    EQUALITY MATCHING RULE          attributeCertificateExactMatch
    ID                              id-at-attributeDescriptorCertificate }

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

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

delegationPath          ATTRIBUTE  ::= {
    WITH SYNTAX         AttCertPath
    ID                  id-at-delegationPath }

AttCertPath  ::=  SEQUENCE OF AttributeCertificate

privPolicy              ATTRIBUTE  ::= {
    WITH SYNTAX         PolicySyntax
    ID                  id-at-privPolicy }

protPrivPolicy     ATTRIBUTE  ::= {
    WITH SYNTAX                     AttributeCertificate
    EQUALITY MATCHING RULE          attributeCertificateExactMatch
    ID                              id-at-protPrivPolicy }

xmlPrivPolicyATTRIBUTE  ::= {
    WITH SYNTAX UTF8String --contains XML-encoded privilege policy information
    ID          id-at-xMLPprotPrivPolicy }

-- Attribute certificate extensions and matching rules --

attributeCertificateExactMatch MATCHING-RULE  ::= {
    SYNTAX          AttributeCertificateExactAssertion
    ID              id-mr-attributeCertificateExactMatch }

AttributeCertificateExactAssertion   ::=  SEQUENCE {
    serialNumber        CertificateSerialNumber,
    issuer              AttCertIssuer
    }

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

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

holderIssuerMatch MATCHING-RULE  ::= {
    SYNTAX          HolderIssuerAssertion
    ID              id-mr-holderIssuerMatch }

HolderIssuerAssertion ::=  SEQUENCE {
    holder          [0]       Holder    OPTIONAL,
```

```
    issuer          [1]     AttCertIssuer OPTIONAL
    }

delegationPathMatch MATCHING-RULE  ::= {
    SYNTAX          DelMatchSyntax
    ID              id-mr-delegationPathMatch }

DelMatchSyntax  ::=  SEQUENCE {
    firstIssuer     AttCertIssuer,
    lastHolder      Holder }

sOAIdentifier EXTENSION  ::= {
    SYNTAX              NULL
    IDENTIFIED BY       id-ce-sOAIdentifier }

sOAIdentifierMatch MATCHING-RULE ::= {
    SYNTAX              NULL
    ID                  id-mr-sOAIdentifierMatch }

authorityAttributeIdentifier EXTENSION  ::=
    {
    SYNTAX              AuthorityAttributeIdentifierSyntax
    IDENTIFIED BY       { id-ce-authorityAttributeIdentifier } }

AuthorityAttributeIdentifierSyntax    ::=  SEQUENCE SIZE (1..MAX) OF AuthAttId

AuthAttId     ::=  IssuerSerial

authAttIdMatch MATCHING-RULE  ::= {
    SYNTAX          AuthorityAttributeIdentifierSyntax
    ID              id-mr-authAttIdMatch }

roleSpecCertIdentifier EXTENSION ::=
    {
    SYNTAX              RoleSpecCertIdentifierSyntax
    IDENTIFIED BY       { id-ce-roleSpecCertIdentifier } }

RoleSpecCertIdentifierSyntax  ::= SEQUENCE SIZE (1..MAX) OF RoleSpecCertIdentifier


RoleSpecCertIdentifier  ::= SEQUENCE {
    roleName                [0]     GeneralName,
    roleCertIssuer          [1]     GeneralName,
    roleCertSerialNumber    [2]     CertificateSerialNumber     OPTIONAL,
    roleCertLocator         [3]     GeneralNames                OPTIONAL }

roleSpecCertIdMatch MATCHING-RULE  ::= {
    SYNTAX          RoleSpecCertIdentifierSyntax
    ID              id-mr-roleSpecCertIdMatch }

basicAttConstraints EXTENSION ::=
    {
        SYNTAX              BasicAttConstraintsSyntax
        IDENTIFIED BY       { id-ce-basicAttConstraints }
    }

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

basicAttConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX          BasicAttConstraintsSyntax
    ID              id-mr-basicAttConstraintsMatch }
```

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

delegatedNameConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX          NameConstraintsSyntax
    ID              id-mr-delegatedNameConstraintsMatch}

timeSpecification EXTENSION  ::=  {
    SYNTAX              TimeSpecification
    IDENTIFIED BY       id-ce-timeSpecification }

timeSpecificationMatch MATCHING-RULE  ::= {
    SYNTAX          TimeSpecification
    ID              id-mr-timeSpecMatch }

acceptableCertPolicies EXTENSION ::= {
    SYNTAX              AcceptableCertPoliciesSyntax
    IDENTIFIED BY       id-ce-acceptableCertPolicies }

AcceptableCertPoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId

CertPolicyId ::= OBJECT IDENTIFIER

acceptableCertPoliciesMatch MATCHING-RULE  ::= {
    SYNTAX          AcceptableCertPoliciesSyntax
    ID              id-mr-acceptableCertPoliciesMatch }

attributeDescriptor EXTENSION  ::= {
    SYNTAX              AttributeDescriptorSyntax
    IDENTIFIED BY       {id-ce-attributeDescriptor } }

AttributeDescriptorSyntax  ::= SEQUENCE {
    identifier                  AttributeIdentifier,
    attributeSyntax             OCTET STRING (SIZE(1..MAX)),
    name                [0]     AttributeName   OPTIONAL,
    description         [1]     AttributeDescription    OPTIONAL,
    dominationRule              PrivilegePolicyIdentifier}

AttributeIdentifier  ::= ATTRIBUTE.&id({AttributeIDs})

AttributeIDs ATTRIBUTE  ::= {...}

AttributeName  ::= UTF8String(SIZE(1..MAX))

AttributeDescription  ::= UTF8String(SIZE(1..MAX))

PrivilegePolicyIdentifier     ::=   SEQUENCE {
    privilegePolicy             PrivilegePolicy,
    privPolSyntax               InfoSyntax }

attDescriptor MATCHING-RULE  ::= {
    SYNTAX          AttributeDescriptorSyntax
    ID              id-mr-attDescriptorMatch }

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

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

```
Targets  ::=  SEQUENCE SIZE (1..MAX) OF Target

Target::=CHOICE {
    targetName          [0]     GeneralName,
    targetGroup         [1]     GeneralName,
    targetCert          [2]     TargetCert }

TargetCert    ::=  SEQUENCE {
    targetCertificate   IssuerSerial,
    targetName          GeneralName OPTIONAL,
    certDigestInfo      ObjectDigestInfo OPTIONAL }

noRevAvail EXTENSION ::= {
    SYNTAX              NULL
    IDENTIFIED BY       id-ce-noRevAvail }

acceptablePrivilegePolicies EXTENSION ::= {
    SYNTAX              AcceptablePrivilegePoliciesSyntax
    IDENTIFIED BY       id-ce-acceptablePrivilegePolicies }

AcceptablePrivilegePoliciesSyntax  ::=  SEQUENCE SIZE (1..MAX) OF PrivilegePolicy

indirectIssuer EXTENSION ::= {
    SYNTAX          BOOLEAN
    IDENTIFIED BY       id-ce-indirectIssuer }

indirectIssuerMatch MATCHING-RULE  ::= {
    SYNTAX  BOOLEAN
    ID  id-mr-indirectIssuerMatch }

noAssertion EXTENSION ::= {
    SYNTAX NULL
    IDENTIFIED BY id-ce-noAssertion }

issuedOnBehalfOf EXTENSION ::= {
    SYNTAX GeneralName
    IDENTIFIED BY id-ce-issuedOnBehalfOf }

-- object identifier assignments --

-- object classes --

id-oc-pmiUser                       OBJECT IDENTIFIER ::=   {id-oc 24}
id-oc-pmiAA                         OBJECT IDENTIFIER ::=   {id-oc 25}
id-oc-pmiSOA                        OBJECT IDENTIFIER ::=   {id-oc 26}
id-oc-attCertCRLDistributionPts     OBJECT IDENTIFIER ::=   {id-oc 27}
id-oc-privilegePolicy               OBJECT IDENTIFIER ::=   {id-oc 32}
id-oc-pmiDelegationPath             OBJECT IDENTIFIER ::=   {id-oc 33}
id-oc-protectedPrivilegePolicy      OBJECT IDENTIFIER ::=   {id-oc 34}

-- directory attributes --

id-at-attributeCertificate              OBJECT IDENTIFIER ::=   {id-at 58}
id-at-attributeCertificateRevocationList OBJECT IDENTIFIER ::=  {id-at 59}
id-at-aACertificate                     OBJECT IDENTIFIER ::=   {id-at 61}
id-at-attributeDescriptorCertificate    OBJECT IDENTIFIER ::=   {id-at 62}
id-at-attributeAuthorityRevocationList  OBJECT IDENTIFIER ::=   {id-at 63}
id-at-privPolicy                        OBJECT IDENTIFIER ::=   {id-at 71}
id-at-role                              OBJECT IDENTIFIER ::=   {id-at 72}
id-at-delegationPath                    OBJECT IDENTIFIER ::=   {id-at 73}
id-at-protPrivPolicy                    OBJECT IDENTIFIER ::=   {id-at 74}
```

```
id-at-xMLPrivilegeInfo                OBJECT IDENTIFIER ::=  {id-at 75}
id-at-xMLPprotPrivPolicy              OBJECT IDENTIFIER ::=  {id-at 76}

-- attribute certificate extensions --

id-ce-authorityAttributeIdentifier    OBJECT IDENTIFIER ::=  {id-ce 38}
id-ce-roleSpecCertIdentifier          OBJECT IDENTIFIER ::=  {id-ce 39}
id-ce-basicAttConstraints             OBJECT IDENTIFIER ::=  {id-ce 41}
id-ce-delegatedNameConstraints        OBJECT IDENTIFIER ::=  {id-ce 42}
id-ce-timeSpecification               OBJECT IDENTIFIER ::=  {id-ce 43}
id-ce-attributeDescriptor             OBJECT IDENTIFIER ::=  {id-ce 48}
id-ce-userNotice                      OBJECT IDENTIFIER ::=  {id-ce 49}
id-ce-sOAIdentifier                   OBJECT IDENTIFIER ::=  {id-ce 50}
id-ce-acceptableCertPolicies          OBJECT IDENTIFIER ::=  {id-ce 52}
id-ce-targetInformation               OBJECT IDENTIFIER ::=  {id-ce 55}
id-ce-noRevAvail                      OBJECT IDENTIFIER ::=  {id-ce 56}
id-ce-acceptablePrivilegePolicies     OBJECT IDENTIFIER ::=  {id-ce 57}
id-ce-indirectIssuer                  OBJECT IDENTIFIER ::=  {id-ce 61}
id-ce-noAssertion                     OBJECT IDENTIFIER ::=  {id-ce 62}
id-ce-issuedOnBehalfOf                OBJECT IDENTIFIER ::=  {id-ce 64}

-- PMI matching rules --

id-mr-attributeCertificateMatch       OBJECT IDENTIFIER ::=  {id-mr 42}
id-mr-attributeCertificateExactMatch  OBJECT IDENTIFIER ::=  {id-mr 45}
id-mr-holderIssuerMatch               OBJECT IDENTIFIER ::=  {id-mr 46}
id-mr-authAttIdMatch                  OBJECT IDENTIFIER ::=  {id-mr 53}
id-mr-roleSpecCertIdMatch             OBJECT IDENTIFIER ::=  {id-mr 54}
id-mr-basicAttConstraintsMatch        OBJECT IDENTIFIER ::=  {id-mr 55}
id-mr-delegatedNameConstraintsMatch   OBJECT IDENTIFIER ::=  {id-mr 56}
id-mr-timeSpecMatch                   OBJECT IDENTIFIER ::=  {id-mr 57}
id-mr-attDescriptorMatch              OBJECT IDENTIFIER ::=  {id-mr 58}
id-mr-acceptableCertPoliciesMatch     OBJECT IDENTIFIER ::=  {id-mr 59}
id-mr-delegationPathMatch             OBJECT IDENTIFIER ::=  {id-mr 61}
id-mr-sOAIdentifierMatch              OBJECT IDENTIFIER ::=  {id-mr 66}
id-mr-indirectIssuerMatch             OBJECT IDENTIFIER ::=  {id-mr 67}

END
```

Anexo B

Reglas de procesamiento y generación de CRL

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

B.1 Introducción

Una parte confiante (usuario de certificado) precisa poder comprobar el estado de revocación de un certificado para determinar si confía o no en dicho certificado. Las listas de revocación de certificados (CRL) son un mecanismo para que las partes confiantes obtengan información de revocación. También se pueden utilizar otros mecanismos, pero están fuera del ámbito de esta Especificación.

Este anexo trata de la utilización de las CRL para que las partes confiantes comprueben el estado de revocación de certificados. Diversas autoridades pueden tener políticas diferentes en relación con la expedición de listas de revocación. Por ejemplo, en algunos casos la autoridad que expide el certificado puede autorizar a una autoridad diferente para que expida una lista de revocación de los certificados que expide. Algunas autoridades pueden combinar la revocación de certificados de entidad final y de CA en una única lista, mientras que otras autoridades las pueden separar en listas diferentes. Algunas autoridades pueden dividir sus certificados en fragmentos de CRL y otras autoridades pueden expedir actualizaciones delta de una lista de revocación entre intervalos de CRL regulares. Por tanto, las partes confiantes necesitan ser capaces de determinar el ámbito de las CRL que recogen para que puedan asegurar que tiene un conjunto completo de la información de revocación incluida en el ámbito del certificado en cuestión para los motivos de revocación de interés, dada la política con la que están trabajando. Este anexo proporciona un mecanismo para que las partes confiantes determinen el ámbito de las CRL recuperadas.

Este anexo se ha escrito para la comprobación del estado de revocación de certificados de clave pública que utilizan las CRL, las CRL de entidad final llena y completa (EPRL, *full and complete end-entity*) y las listas de revocación de autoridades de certificación (CARL, *certification authority revocation lists*). Sin embargo, esta descripción también se puede aplicar a la comprobación del estado de revocación de certificados de atributo que utilizan listas de revocación de certificados de atributo (ACRL) y listas de revocación de autoridades de atributo (AARL, *attribute authority revocation lists*). Para los fines de este anexo se puede considerar una ACRL en lugar de una CRL, una EPRL puede ser una ACRL de entidad final llena y completa y se puede considerar una AARL en lugar de una CARL. De modo similar, los atributos de directorio identificados en B.4 se harán corresponder con los de las AARL y ACRL y los campos que identifican los tipos de certificados en la extensión de punto de distribución expedidor pueden hacerse corresponder con los que son aplicables a la PMI.

B.1.1 Tipos de CRL

Una parte confiante puede disponer de CRL de uno o más de los tipos siguientes, basadas en los aspectos de revocación de la política de la autoridad que expide el certificado:

- CRL llena y completa.
- CRL de entidad final (EPRL) llena y completa.
- Lista de revocación de autoridad de certificación (CARL) llena y completa.
- CRL, EPRL, o CARL de punto de distribución.
- CRL, EPRL o CARL indirecta (ICRL).
- CRL, EPRL o CARL delta.
- dCRL, EPRL o CARL indirecta.

Una CRL llena y completa es una lista de todos los certificados de CA y de entidad final revocados, expedidos por una autoridad por uno o todos los motivos.

Una EPRL llena y completa es una lista de todos los certificados de entidad final revocados, expedidos por una autoridad por uno o todos los motivos.

Una CARL llena y completa es una lista de certificados de CA revocados, expedidos para una autoridad por uno o todos los motivos.

Una CRL, EPRL o CARL de punto de distribución es la que cubre todos o un subconjunto de certificados expedidos por una autoridad. El subconjunto se podría basar en diversos criterios.

Una CRL, EPRL o CARL indirecta (ICRL) es una CRL que contiene una lista de certificados revocados, en la que alguno o todos los certificados fueron expedidos por la autoridad que firma y expide la CRL.

Una CRL, EPRL o CARL delta es una CRL que sólo contiene modificaciones a una CRL completa para un ámbito determinado al mismo tiempo que la CRL referenciada en la dCRL. Hay que destacar que la CRL referenciada podría ser una completa para un determinado ámbito o podría ser una dCRL que se utiliza para construir localmente una CRL completa para un determinado ámbito.

Todos los tipos de CRL anteriores (salvo la dCRL) son tipos de CRL completas para su ámbito. Una dCRL se tendrá que utilizar junto con una CRL asociada que esté completa para el mismo ámbito con el fin de formar una visión competa del estado de revocación de los certificados.

Una CRL, EPRL o CARL indirecta es una CRL que sólo contiene modificaciones a un conjunto de una o más CRL, completas para sus propios ámbitos, y en la que alguno o todos los certificados pueden haber sido expedidos por la autoridad que firma y que expide esa CRL.

En este anexo, así como en esta Especificación, "Ámbito de una CRL" se define mediante dos dimensiones independientes. Una dimensión es el conjunto de certificados cubiertos por la CRL. Otra dimensión es el conjunto de códigos de motivo cubiertos por la CRL. El ámbito de una CRL se puede determinar de una o más de las maneras siguientes:

- extensión de punto de distribución expedidor (IDP, *issuing distribution point*) en la CRL; u
- otros medios, fuera del ámbito de esta Especificación.

B.1.2 Procesamiento de CRL

Si una parte confiante está utilizando las CRL como el mecanismo para determinar si un certificado está revocado, tendrá que utilizar la CRL o las CRL adecuadas para dicho certificado. Este anexo describe un procedimiento para obtener y procesar las CRL adecuadas siguiendo algunos pasos específicos. Una implementación funcionalmente equivalente al comportamiento externo que resulta de este procedimiento también se considerará conforme con este anexo y con la Especificación asociada. El algoritmo utilizado por una determinada implementación para derivar la salida correcta (es decir, estado de revocación de certificado) a partir de entradas dadas (el propio certificado y la entrada de la política local) no está normalizado. Por ejemplo, aunque este procedimiento se describe como una secuencia de pasos a procesar en orden, una implementación puede utilizar las CRL que están en su memoria local en lugar de recuperar las CRL cada vez que procesa un certificado, siempre que esas CRL estén completas para el ámbito del certificado y no violen ninguno de los parámetros del certificado o de la política.

En B.2 a B.5 se describen las siguientes etapas generales:

1) Determinación de los parámetros para las CRL.
2) Determinación de las CRL requeridas.
3) Obtención de las CRL.
4) Procesamiento de las CRL.

El paso 1) identifica los parámetros provenientes del certificado y de otros lugares, que se utilizarán para determinar qué tipos de CRL se precisan.

El paso 2) aplica los valores de los parámetros para tomar la determinación.

El paso 3) identifica los atributos de directorio de los que se pueden extraer los tipos de CRL.

El paso 4) describe el procesamiento de las CRL pertinentes.

B.2 Determinación de los parámetros para las CRL

La información ubicada en el propio certificado, así como la información proveniente de la política con la que está operando la parte confiante, proporcionan los parámetros para determinar si las CRL candidatas son adecuadas. Se necesita la siguiente operación para determinar qué tipo de CRL son adecuadas:

- Certificado (es decir, entidad final o CA).
- Punto de distribución de CRL crítica.
- CRL crítica más reciente.
- Códigos de motivo de interés.

El tipo de certificado se puede determinar a partir de la extensión de constricciones básicas en el certificado. Si la extensión está presente, indica si el certificado es un certificado de CA o un certificado de entidad final. Si la extensión

está ausente, se considera que el tipo de certificado es de entidad final. Esta información se necesita para determinar si se puede utilizar una CRL, EPRL o CARL para comprobar la revocación del certificado.

Si el certificado contiene una extensión de punto de distribución de CRL crítica, el sistema de procesamiento de certificados de la parte confiante tendrá que comprender esta extensión y obtener y utilizar las CRL señaladas por la extensión de punto de distribución de CRL para los códigos de motivo de interés a fin de determinar el estado de revocación del certificado. Confiar en una CRL llena, por ejemplo, no sería suficiente.

Si el certificado contiene una extensión de CRL crítica más reciente, la parte confiante no puede utilizar el certificado sin extraer y comprobar en primer lugar la CRL más reciente.

Los códigos de motivo de interés los determina la política y normalmente los suministra la aplicación. Se recomienda que éstos incluyan todos los códigos de motivo. Esta información se necesita para determinar cuales CRL son suficientes en términos de códigos de motivo.

Hay que destacar que la política puede también determinar si se espera o no que una parte confiante compruebe las dCRL para su estado de revocación, incluso cuando la extensión **freshestCRL** está indicada como no crítica mediante banderas o está ausente del certificado. Aunque no figura en este paso, el procesamiento de estas dCRL facultativas se describe en el paso 4).

B.3 Determinación de las CRL requeridas

Los valores de los parámetros descritos en B.2 determinan el criterio sobre qué tipos de CRL son necesarios para comprobar el estado de revocación de un determinado certificado. La determinación de los tipos de CRL se puede realizar tomando como base los conjuntos siguientes de criterios, como se describe en B.3.1 a B.3.4.

- Certificado de entidad final con DP de CRL crítica afirmado.
- Certificado de entidad final con DP de CRL no crítica afirmado.
- Certificado de CA con DP de CRL crítica afirmado.
- Certificado de CA con DP de CRL no crítica afirmado.

El tratamiento de los parámetros restantes (extensión de CRL crítica más reciente y conjunto de códigos de motivo de interés) se realiza con cada una de las subcláusulas.

Hay que destacar que en cada caso, más de un tipo de CRL puede satisfacer los requisitos. Cuando se plantee la elección del tipo de CRL, la parte confiante puede seleccionar cualquiera de los tipos adecuados de utilización.

B.3.1 Entidad final con DP de CRL crítica

Si el certificado es un certificado de entidad final y está presente la extensión **cRLDistributionPoints** y se indica como crítica mediante banderas, se obtendrán las CRL siguientes:

- una CRL de las CRL de punto de distribución denominada que incluye uno o más de los códigos de motivo de interés;
- si todos los códigos de motivo de interés no están incluidos en dicha CRL, se puede satisfacer el estado de revocación para los código de motivo restantes mediante una combinación de las CRL siguientes:
  - CRL de punto de distribución adicional.
  - CRL completas adicionales.
  - EPRL completas adicionales.

Si la extensión de CRL más reciente también está presente en el certificado y se indica como crítica mediante banderas, se obtendrán también una o más CRL de uno o más de los puntos de distribución denominados en dicha extensión, asegurando que se ha comprobado la información de revocación más reciente para todos los códigos de motivo.

B.3.2 Entidad final con DP de CRL no crítica

Si el certificado es un certificado de entidad final y la extensión **cRLDistributionPoints** está ausente del certificado o presente pero indicada como no crítica mediante banderas, se puede satisfacer el estado de revocación de los códigos de motivo de interés mediante una combinación de las CRL siguientes:

- CRL de punto de distribución (si están presentes).
- CRL completas.
- EPRL completas.

Si la extensión de CRL más reciente también está presente en el certificado y se indica como crítica mediante banderas, se obtendrá también una o más CRL a partir de uno o más puntos de distribución denominados en dicha extensión, asegurando que se comprueba la información de revocación más reciente para todos los códigos de motivo.

B.3.3 CA con DP de CRL crítica

Si el certificado es una CA y la extensión **cRLDistributionPoints** está presente en el certificado y se indica como crítica mediante banderas, se obtendrán las siguientes CRL/CARL:

- Una CRL o una CARL a partir de uno de los puntos de distribución denominados que incluya uno o más códigos de motivo de interés.
- Si todos los códigos de motivo de interés no están cubiertos por dicha CRL/CARL, se puede satisfacer el estado de revocación para los códigos de motivo restantes mediante una combinación de las siguientes CRL/CARL:
  - CRL/CARL de punto de distribución adicionales.
  - CRL completas adicionales.
  - CARL completas adicionales.

Si la extensión de CRL más reciente también está presente en el certificado y se indica como crítica mediante banderas, se obtendrán también una o más CRL/CARL a partir de uno o más puntos de distribución denominados en dicha extensión, asegurando que se comprueba la información de revocación más reciente para todos los códigos de motivo.

B.3.4 CA con DP de CRL no crítica

Si el certificado es un certificado de CA y la extensión **cRLDistributionPoints** está ausente del certificado o está presente pero indicada como crítica mediante banderas, se puede satisfacer el estado de revocación de los códigos de motivo de interés mediante una combinación de las CRL siguientes:

- CRL/CARL de punto de distribución (si están presentes).
- CRL completas.
- CARL completas.

Si la extensión de CRL más reciente también está presente en el certificado y se indica como crítica mediante banderas, se obtendrán también una o más CRL/CARL a partir de uno o más puntos de los distribución denominados en dicha extensión, asegurando que se comprueba la información de revocación más reciente para todos los códigos de motivo.

B.4 Obtención de las CRL

Si la parte confiante está extrayendo las CRL adecuadas del directorio, estas CRL se obtiene a partir del DP de CRL o del asiento de directorio de expedidor de certificado mediante al extracción de los atributos adecuados, es decir, uno o más de los atributos siguientes:

- Lista de revocación de certificados.
- Lista de revocación de autoridades.
- Lista de revocación delta.

B.5 Procesamiento de las CRL

Después de considerar los parámetros tratados en B.2 , de identificar los tipos de CRL adecuados como se describe en B.3 y de extraer un conjunto adecuado de CRL como se describe en B.4, una parte confiante está dispuesta para procesar las CRL. El conjunto de CRL contendrá por lo menos una CRL básica y puede también contener una o más dCRL. Para cada CRL que se está procesando, la parte confiante tendrá que asegurar que la CRL es precisa en relación con su ámbito. La parte confiante ya ha determinado que la CRL es adecuada para el ámbito del certificado de interés, mediante los procesos de B.2 y B.3 anteriores. Además, se tendrán que realizar comprobaciones de la validez de las CRL y se tendrán que comprobar para determinar si se ha revocado o no el certificado. Estas comprobaciones se describen en B.5.1 a B.5.4.

B.5.1 Validación del ámbito de CRL básica

Como se ha descrito en B.3, puede haber más de un tipo de CRL que puede ser utilizada como CRL básica para comprobar el estado de revocación de un certificado. En función de la política de la autoridad expedidora en relación con la expedición de la CRL, la parte confiante puede disponer de uno o más de los tipos siguientes de CRL básica.

- CRL completa para todas las entidades.
- EPRL completa.

- CARL completa.
- CRL/EPRL/CARL basadas en punto de distribución.

Las subcláusulas B.5.1.1 a B.5.1.4 proporcionan un conjunto de condiciones que tendrán que ser verdaderas para que una parte confiante utilice una CRL de cada tipo como la CRL básica para la comprobación del estado de revocación de certificados para los códigos de motivo de interés.

Las CRL básicas indirectas se tratan en cada una de las subcláusulas.

B.5.1.1 CRL completa

Para determinar que una CRL es una CRL completa para certificados de entidad final y de CA de los cuales es responsable el expedidor de CRL, para todos los códigos de motivo de interés, tendrá que ser cierto lo siguiente:

- la extensión de indicador de CRL delta tendrá que estar ausente; y
- la extensión de punto de distribución expedidor puede estar presente; y
- o la extensión de punto de distribución expedidor no debe contener el campo punto de distribución o bien uno de los nombres en el campo punto de distribución debe corresponder con el campo **issuer** en la CRL; y
- la extensión de punto de distribución expedidor no debe contener ninguno de los siguientes campos o en caso de que contenga cualquiera de ellos, ninguno de los campos presentes deberá fijarse a VERDADERO: containsUserPublicKeyCerts, containsCACerts, containsUserAttributeCerts, containsAACerts, y/o containsSOAPublicKeyCerts; y
- si está presente el campo **reasonCodes** en la extensión de punto de distribución expedidor, el campo de código de motivo tendrá que incluir todos los motivos de interés para la aplicación; y
- la extensión de punto de distribución expedidor puede o no contener el campo **indirectCR** (por tanto, no es necesario comprobar este campo).

B.5.1.2 EPRL completa

Para determinar que una CRL es una EPRL completa para los códigos de motivo de interés, tendrán que ser ciertas todas las condiciones siguientes:

- la extensión de indicador de CRL delta tendrá que estar ausente; y
- la extensión de punto de distribución de expedidor tendrá que estar presente; y
- o la extensión de punto de distribución expedidor no debe contener el campo de punto de distribución o bien uno de los nombres en el campo punto de distribución debe corresponder con el campo **issuer** en la CRL; y
- la extensión de punto de distribución expedidor tendrá que contener el campo **containsUserPublicKeyCerts**. Este campo tendrá que estar fijado a **VERDADERO**; y
- si está presente el campo **reasonCodes** en la extensión de punto de distribución expedidor, el campo de código de motivo tendrá que incluir todos los motivos de interés para la aplicación; y
- la extensión de punto de distribución expedidor puede o no contener el campo **indirectCRL** (por lo que no es necesario comprobar este campo); y

esta CRL sólo se puede utilizar si la parte confiante ha decidido que el certificado de sujeto sea aun certificado de entidad final. Por tanto, si el certificado de sujeto contiene la extensión **basicConstraints**, su valor tendrá que ser **cA=FALSO**.

B.5.1.3 CARL completa

Para determinar que una CRL es una CARL completa para los códigos de motivo de interés, tendrán que ser ciertas todas las condiciones siguientes:

- la extensión de indicador de CRL delta tendrá que estar ausente; y
- la extensión de distribución de punto expedidor tendrá que estar presente; y
- o la extensión de punto de distribución expedidor no podrá contener el campo punto de distribución o bien uno de los nombres en el campo punto de distribución debe corresponder con el campo **issuer** en la CRL; y
- la extensión de punto de distribución expedidor tendrá que contener el campo **containsCACerts.** Este campo tendrá que fijarse a **VERDADERO**; y

- si está presente el campo **reasonCodes** en la extensión de punto de distribución de expedidor, el campo de código de motivo tendrá que incluir todos los motivos de interés para la aplicación; y
- la extensión de punto de distribución expedidor puede o no contener el campo **indirectCRL** (por lo que no es necesario comprobar este campo); y

Esta CARL sólo se puede utilizar si el certificado de sujeto es un certificado de CA. Por tanto, el certificado de sujeto tendrá que contener la extensión **basicConstraints** con el valor **cA=VERDADERO**.

B.5.1.4 CRL/EPRL/CARL basada en el punto de distribución

Para determinar si una CRL es una de las CRL indicadas mediante una extensión de punto de distribución de CRL o la extensión CRL más reciente en el certificado, todas las condiciones siguientes tendrán que ser ciertas:

- o el campo de punto de distribución en la extensión de punto de distribución expedidor de la CRL tendrá que estar ausente (sólo si no se busca un DP de CRL crítica), o uno de los nombres en el campo de punto de distribución de la extensión de punto de distribución de CRL o de la extensión de la CLR más reciente del certificado tendrá que concordar con uno de los nombres en el campo de punto de distribución en la extensión de punto de distribución expedidor de la CRL. Como alternativa, uno de los nombres en el campo **cRLIssuer** del DP de CRL o la extensión de la CRL más reciente del certificado puede concordar con uno de los nombres en el punto de distribución del IDP; y
- la extensión de punto de distribución expedidor no debe contener ninguno de los siguientes campos, o en caso de que contenga cualquiera de ellos, ninguno de los campos presentes deberá fijarse a VERDADERO: containsUserPublicKeyCerts, containsCACerts, containsUserAttributeCerts, containsAACerts, y/o containsSOAPublicKeyCerts, o bien el campo apropiado para el tipo de certificado debe fijarse a VERDADERO (véase en el cuadro B.1 el tipo de campo para cada tipo de certificado); y
- si el campo de código de motivo está presente en la extensión de punto de distribución de CRL o en la extensión de la CRL más reciente del certificado, este campo tendrá que estar ausente de la extensión de punto de distribución expedidor de la CRL o contener por lo menos uno de los códigos de motivo afirmados en la extensión de punto de distribución de CRL del certificado; y
- si el campo **cRLIssuer** está ausente de la extensión de punto de distribución de CRL del certificado, tendrá que estar firmada por la misma CA que firmó el certificado; y
- si el campo **cRLIssuer** está presente en la extensión relativa (punto de distribución de CRL o extensión de la CRL más reciente) del certificado, la CRL tendrá que estar firmada por el expedidor de CRL identificado en la extensión de punto de distribución de CRL o en la extensión de la CRL más reciente del certificado y la CRL tendrá que contener el campo **indirectCRL** en la extensión de punto de distribución expedidor.

  NOTA – Cuando se comprueba la presencia de los motivos y del campo cRLIssuer, la prueba sólo puede ser satisfactoria si el campo está presente en el mismo DistributionPoint del DP de CRL o en la extensión de la CRL más reciente para la cual existe una concordancia de nombre en el campo punto de distribución de la extensión IDP en la CRL correspondiente.

Cuadro B.1 – Tipo de certificado y campo punto de distribución expedidor

Tipo de certificado	Campo punto de distribución expedidor
Entidad final (clave pública)	containsUserPublicKeyCerts
CA	containsCACerts
Entidad final (atributo)	containsUserAttributeCerts
AA	containsAACerts
SOA	containsSOAPublicKeyCerts

B.5.2 Validación del ámbito de CRL delta

La parte confiante también puede estar comprobando las dCRL, ya sea porque se requiere mediante una extensión **freshestCRL** crítica en el certificado o CRL, o porque la política según la cual está funcionado la parte confiante exige la comprobación de dCRL.

La parte confiante puede estar siempre segura de que tiene la información de CRL adecuada para un certificado si se cumplen todas las condiciones siguientes:

- la CRL básica que está utilizando la parte confiante es adecuada para el certificado (en términos del ámbito); y

- la CRL delta que la parte confiante está utilizando es adecuada para el certificado (en términos del ámbito); y
- la CRL básica se expidió en el mismo momento o después que la CRL básica referenciada por la dCRL.

Para determinar que la dCRL es adecuada para el certificado, todas las condiciones siguientes tendrán que ser ciertas:

- la extensión de indicador de CRL delta tendrá que estar presente; y
- la dCRL tendrá que haberse expedido después de la CRL básica. Una manera de asegurar esto consiste en comprobar que el número de CRL en la extensión **crlNumber** de la dCRL es superior al número de CRL en la extensión **crlNumber** de la CRL básica que está utilizando la parte confiante y que los campos **cRLStreamIdentifier** en la CRL básica y en la dCRL concuerdan. Este planteamiento puede necesitar lógica adicional para tener en cuenta el número de envolturas. Otra manera consiste en comparar los campos **thisUpdate** que tiene la parte confiante en la CRL básica y en la dCRL; y
- la CRL básica que está utilizando la parte confiante tendrá que ser la utilizada para expedir la dCRL u otra posterior. Una manera de asegurar esto consiste en comprobar que el número de CRL en la extensión **deltaCRLIndicator** de la dCRL es menor o igual que el número de CRL en la extensión **crlNumber** de la CRL básica que está utilizando la parte confiante y que los campos **cRLStreamIdentifier** en la CRL básica y en la dCRL concuerdan. Este planteamiento puede necesitar lógica adicional para tener en cuenta el número de envolturas. Otra manera consiste en comparar los campos **thisUpdate** de la CRL básica que tiene la parte confiante y de la CRL básica a la que indica la dCRL. Otra manera consiste en comparar el campo **thisUpdate** en la CRL básica que tiene la parte confiante y la extensión **baseUpdateTime** en la dCRL que tiene la parte confiante; y

  NOTA – Una parte confiante siempre puede construir una CRL básica aplicando una dCRL a una CRL básica siempre que se satisfagan las dos reglas anteriores utilizando las comprobaciones de **crlNumber** y **cRLStreamIdentifier**. En este caso, la extensión **crlNumber** y el campo **thisUpdate** de la CRL básica nueva son los de la dCRL. La parte confiante no conoce el campo **nextUpdate** de la nueva CRL básica y no necesita conocerlo para asociarlo con otra dCRL.
- si la dCRL contiene una extensión de punto de distribución expedidor, el ámbito del punto de distribución expedidor será coherente con el certificado según se describe en B.5.1.4 anterior; y
- si la dCRL no contiene ninguna de las extensiones siguientes: **streamIdentifier** e **issuingDistributionPoint**, se utilizará únicamente junto con una CRL básica llena y completa.

B.5.3 Validez y comprobaciones de actualidad en la CRL básica

Para verificar que una CRL básica es precisa y que no se ha modificado desde su expedición, se tendrán que satisfacer todas las condiciones siguientes:

- la parte confiante tendrá que ser capaz de obtener la clave pública del expedidor identificado en la CRL utilizando medios autenticados; y
- se tendrá que verificar la firma de la CRL básica utilizando esta clave pública autenticada; y
- si está presente el campo **nextUpdate**, el tiempo actual debe ser anterior al campo **nextUpdate**; y
- el nombre de expedidor en la CRL tendrá que concordar con el nombre de expedidor en el certificado que se está comprobando para su revocación, a menos que se extraiga la CRL del DP de CRL en el certificado y que la extensión DP de CRL contenga el componente expedidor de CRL. En este caso, uno de los nombres en el componente expedidor de CRL en la extensión DP de CRL tendrá que concordar con el nombre de expedidor en la CRL.

B.5.4 Validez y comprobaciones en la CRL delta

Para verificar que una dCRL es precisa y que no se ha modificado desde su expedición, se tendrán que satisfacer todas las condiciones siguientes:

- la parte confiante tendrá que ser capaz de obtener la clave pública del expedidor identificado en la CRL utilizando medios autenticados; y
- se tendrá que verificar la firma de la dCRL utilizando esta clave pública autenticada; y
- si está presente el campo **nextUpdate**, el tiempo actual debe ser anterior al campo **nextUpdate**; y
- el nombre de expedidor en la dCRL tendrá que concordar con el nombre de expedidor en el certificado que se está comprobando para su revocación, a menos que la CRL delta se extraiga del DP de CRL en el certificado y la extensión DP de CRL contenga el componente expedidor de CRL. En este caso, uno de los nombres en el componente expedidor de CRL en la extensión DP de CRL tendrá que concordar con el nombre de expedidor de la CRL.

Anexo C

Ejemplos de expedición de CRL delta

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

Existen dos modelos para expedir CRL que implican la utilización de dCRL para un determinado conjunto de certificados.

En el primer modelo, cada dCRL hace referencia a la CRL más reciente que esté completa para un determinado ámbito. Se pueden expedir varias dCRL para el mismo ámbito antes de que se expida una nueva CRL completa para dicho ámbito. La nueva CRL se utiliza como base para la siguiente secuencia de dCRL y es la CRL de referencia en la extensión pertinente en la dCRL. Cuando se expide la CRL que está completa para el ámbito, también se expide una dCRL final para la CRL previa completa para el ámbito.

El segundo modelo, aunque muy similar, difiere en que la CRL referenciada por una dCRL no es necesariamente una que esté completa para un ámbito dado (es decir, la CRL referenciada puede sólo haberse expedido como una dCRL). Si la CRL referenciada es una completa para el ámbito determinado, puede no ser necesariamente la más reciente que esté completa para dicho ámbito.

Un certificado que utiliza un sistema que está procesando una dCRL tendrá que tener también una CRL completa para dicho ámbito y esto es por lo menos tan habitual como la CRL referenciada en la dCRL. Esta CRL completa para dicho ámbito puede ser una expedida como tal por la autoridad responsable o puede ser una construida localmente por el sistema que utiliza el certificado. Hay que destacar que en algunas situaciones puede haber información duplicada en la dCRL y en la CRL completa para dicho ámbito, si , por ejemplo, el sistema que utiliza el certificado tiene una CRL que se expidió después de la CRL referenciada en la dCRL.

El cuadro siguiente ilustra tres ejemplos de la utilización de las dCRL. El ejemplo 1 es el esquema tradicional descrito como el primer modelo anterior. Los ejemplos 2 y 3 son variantes del segundo modelo descrito anteriormente.

En el ejemplo 2, la autoridad expide las CRL, completas para dicho ámbito, cada dos días y las dCRL hacen referencia a la antepenúltima CRL completa para el ámbito. Este esquema puede ser útil en entornos en los que existe la necesidad de reducir el número de usuarios que acceden al depósito al mismo tiempo para extraer una CRL completa para un determinado ámbito. En el ejemplo 2, los usuarios que tienen la CRL más reciente que esté completa para el ámbito, así como los usuarios que tienen la CRL anterior más reciente completa para el ámbito, pueden utilizar la misma dCRL. Ambos conjuntos de usuarios tiene información de revocación completa sobre los certificados para dicho ámbito en el instante de expedición de la dCRL que se está utilizando.

En el ejemplo 3, las CRL completas para el ámbito dado se expiden una vez a la semana como en el ejemplo 1, pero cada dCRL hace referencia a una base de información de revocación siete días antes de dicha dCRL.

Aquí no se proporciona un ejemplo de la utilización de las CRL indirectas, pero constituye en superconjunto de estos ejemplos.

Éstos son sólo ejemplos y son también posibles otras variantes, dependiendo de la política local. Algunos factores que podrían considerarse cuando se establece dicha política incluyen; número de usuarios y frecuencia de acceso de las CRL, réplica de las CRL, equilibrado de las cargas para sistemas de directorio que tienen CRL, características de funcionamiento, requisitos de estado latente, etc.

Día	Ejemplo 1 – Delta hace referencia a la CRL más reciente completa para un ámbito dado		Ejemplo 2 – Delta hace referencia a la siguiente CRL más reciente que está completa para un ámbito dado		Ejemplo 3 – Delta hace referencia a la información de revocación de hace 7 días	
	CRL completa para el ámbito dado	**CRL delta**	**CRL completa para el ámbito dado**	**CRL delta**	**CRL completa para el ámbito dado**	**CRL delta**
8	thisUpdate=día 8 nextUpdate=día 15 crlNumber=8	thisUpdate=día 8 nextUpdate=día 9 crlNumber=8 BaseCRLNumber=1	thisUpdate=día 8 nextUpdate=día 10 crlNumber=8	thisUpdate=día 8 nextUpdate=día 9 crlNumber=8 BaseCRLNumber=6	thisUpdate=día 8 nextUpdate=día 15 cRLNumber=8	thisUpdate=día 8 nextUpdate=día 9 cRLNumber=8 BaseCRLNumber= 1
9	no expedida	thisUpdate=día 9 nextUpdate=día 10 crlNumber=9 BaseCRLNumber=8	no expedida	thisUpdate=día 9 nextUpdate=día 10 crlNumber=9 BaseCRLNumber=6	no expedida	thisUpdate=día 9 nextUpdate=día 10 cRLNumber=9 BaseCRLNumber= 2
10	no expedida	thisUpdate=día 10 nextUpdate=día 11 crlNumber=10 BaseCRLNumber=8	thisUpdate=día 10 nextUpdate=día 12 crlNumber=10	thisUpdate=día 10 nextUpdate=día 11 crlNumber=10 BaseCRLNumber=8	no expedida	thisUpdate=día 10 nextUpdate=día 11 cRLNumber=10 BaseCRLNumber= 3
11-14	El esquema continúa como en los días precedentes					
15	thisUpdate=día 15 nextUpdate=día 22 crlNumber=15	thisUpdate=día 15 nextUpdate=día 16 crlNumber=15 BaseCRLNumber=8	no expedida	thisUpdate=día 15 nextUpdate=día 16 crlNumber=15 BaseCRLNumber=12	thisUpdate=día 15 nextUpdate=día 22 cRLNumber=15	thisUpdate=día 15 nextUpdate=día 16 cRLNumber=15 BaseCRLNumber= 8
16	no expedida	thisUpdate=día 16 nextUpdate=día 17 crlNumber=16 BaseCRLNumber=15	thisUpdate=día 16 nextUpdate=día 18 crlNumber=16	thisUpdate=día 16 nextUpdate=día 17 crlNumber=16 BaseCRLNumber=14	no expedida	thisUpdate=día 16 nextUpdate=día 17 cRLNumber=16 BaseCRLNumber= 9

Anexo D

Ejemplos de definición de política de privilegios y de atributo de privilegios

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

D.1 Introducción

La política de privilegios define, para la gestión de privilegios, exactamente cuándo un verificador de privilegios debe concluir que un conjunto presentado de privilegios es suficiente para garantizar el acceso (al objeto, recurso, aplicación, etc., requeridos) al asertor de privilegios. La especificación formal de la política de privilegios puede ayudar a un verificador de privilegios asesorando automáticamente sobre un privilegio de asertor de privilegios contra la sensibilidad del recurso solicitado, puesto que incluye las reglas para determinar el éxito/fracaso de una petición de asertor de privilegios, considerando su privilegio y la sensibilidad del recurso.

Puesto que existen requisitos para asegurar la integridad de la política de privilegios que se está utilizando en estas determinaciones, se pueden transmitir un identificador de la política de privilegios, en la forma de un identificador de objetos y un TROCEO de la totalidad de la política de privilegios en objetos firmados, almacenados en asientos de directorio, etc. Sin embargo, en esta Especificación no está normalizada ninguna sintaxis específica para definir un ejemplar de la política de privilegios.

D.2 Sintaxis de muestra

La política de privilegios se puede definir utilizando cualquier sintaxis, incluido el texto claro. Para ayudar a los que definen las políticas de privilegios en el entendimiento de diversas opciones de las definiciones, se incluyen en este anexo dos sintaxis de muestra, que podrían utilizarse para estos fines. Hay que insistir en que estos son sólo ejemplo y que para soportar éstas y otras sintaxis específicas NO se requiere la implementación de la gestión de privilegios mediante la utilización de certificados de atributo o de la extensión **subjectDirectoryAttributes** de certificados de clave pública.

D.2.1 Primer ejemplo

La siguiente sintaxis ASN.1 es un ejemplo de un útil sencillo y flexible para la definición de la política de privilegios.

```
PrivilegePolicySyntax       ::= SEQUENCE {
    version         Version,
    ppe             PrivPolicyExpression }

PrivPolicyExpression        ::= CHOICE {
    ppPredicate     [0] PrivPolicyPredicate,
    and             [1] SET SIZE (2..MAX) OF PrivPolicyExpression,
    or              [2] SET SIZE (2..MAX) OF PrivPolicyExpression,
    not             [3] PrivPolicyExpression,
    orderedPPE      [4] SEQUENCE OF PrivPolicyExpression }
    -- Note: "sequence" defines the temporal order in which the
    -- privilege shall be examined

PrivPolicyPredicate         ::= CHOICE {
    present                 [0] PrivilegeIdentifier,
    equality                [1] PrivilegeComparison, -- single/set-valued priv.
    greaterOrEqual          [2] PrivilegeComparison, -- single-valued priv.
    lessOrEqual             [3] PrivilegeComparison, -- single-valued priv.
    subordinate             [4] PrivilegeComparison, -- single-valued priv.
    substrings              [5] SEQUENCE {      -- single-valued priv.
          type                      PrivilegeType,
          initial               [0] PrivilegeValue OPTIONAL,
          any                   [1] SEQUENCE OF PrivilegeValue,
          final                 [2] PrivilegeValue OPTIONAL },
    subsetOf                [6] PrivilegeComparison, -- set-valued priv.
    supersetOf              [7] PrivilegeComparison, -- set-valued priv.
    nonNullSetInter         [8] PrivilegeComparison, -- set-valued priv.
    approxMatch             [9] PrivilegeComparison,
    -- single/set-valued priv. (approximation defined by application)
```

```
    extensibleMatch         [10] SEQUENCE {
        matchingRule            OBJECT IDENTIFIER,
        inputs                  PrivilegeComparison } }


PrivilegeComparison   ::= CHOICE {
    explicit        [0] Privilege,
   -- the value(s) of an external privilege identified by
   -- Privilege.privilegeId is(are) compared with the value(s)
   -- explicitly provided in Privilege.privilegeValueSet
   byReference      [1] PrivilegeIdPair }
   -- the value(s) of an external privilege identified by
   -- PrivilegeIdPair.firstPrivilege is(are) compared with
   -- the value(s) of a second external privilege identified by
   -- PrivilegeIdPair.secondPrivilege

Privilege           ::= SEQUENCE {
    type            PRIVILEGE.&id ({SupportedPrivileges}),
    values          SET SIZE (0..MAX) OF
                    PRIVILEGE.&Type ({SupportedPrivileges} {@type})
    }

SupportedPrivileges  PRIVILEGE ::=     { ... }
PRIVILEGE ::=       ATTRIBUTE
-- Privilege is analogous to Attribute

PrivilegeIdPair     ::=  SEQUENCE {
    firstPrivilege      PrivilegeIdentifier,
    secondPrivilege     PrivilegeIdentifier }

PrivilegeIdentifier ::=  CHOICE {
    privilegeType       [0] PRIVILEGE.&id ({SupportedPrivileges}),
    xmlTag              [1] OCTET STRING,
    edifactField        [2] OCTET STRING }
-- PrivilegeIdentifier extends the concept of AttributeType to other
-- (e.g., tagged) environments, such as XML and EDIFACT

Version             ::= INTEGER { v1(0) }
```

Un ejemplo concreto puede ayudar a aclarar la creación y utilización del constructivo **PrivilegePolicy** anterior.

Considérese el privilegio aprobar un incremento salarial. Por sencillez, considérese que la política a seguir establece que sólo gestores senior y superiores pueden aprobar los aumentos y que su aprobación sólo puede darse para una posición inferior a la suya propia (por ejemplo un director puede aprobar un aumento para un gestor senior, pero no para un vicepresidente). Para este ejemplo, supóngase que existen seis posibles niveles ("Personal Técnico" = 0, "Gestor" = 1, "Gestor Senior" = 2, "Director" = 3, "Vicepresidente" = 4, "Presidente" = 5).

Supóngase además que el tipo de atributo (el "privilegio") que identifica el nivel en un certificado de atributo es OBJECT ID *OID-C* y que el tipo de atributo (la "sensibilidad") que identifica el nivel en el registro de la base de datos cuyo campo de salario debe modificarse es OBJECT ID *OID-D* (que se substituiría por supuesto por identificadores de objeto reales en una implementación real). La siguiente expresión booleana describe la política "aprobación de salario" deseada (la codificación de esto en una expresión **PrivilegePolicy** es una tarea relativamente directa):

AND (NOT (lessOrEqual (valor correspondiente a OID-C, valor correspondiente a OID-D))

subsetOf (valor correspondiente a OID-C, { 2, 3, 4, 5 }))

Esta codificación de política dice que la categoría del que aprueba deberá ser superior a la (expresado como "NOT less-than-or-equal-to") categoría del aprobado Y que la categoría del que aprobará tiene que ser uno de {Gestor Senior, ..., Presidente} de forma que esta expresión booleana se evalúe como VERDADERA. La primera comparación de privilegio es "por referencia", comparando los valores correspondientes al tipo de atributo "categoría" para ambas entidades implicadas. La segunda comparación de privilegios es "explícita"; aquí se compara el valor correspondiente al privilegio "categoría" para el que aprueba con una lista de valores incluida explícitamente. El verificador de privilegios en este caso, por lo tanto, necesita un constructivo que codifique esta política junto con dos atributos, uno asociado con el que aprueba y otro asociado con el aprobado. El atributo del que aprueba (que estaría contenido en un certificado de atributo) puede tener el valor {*OID-C* 3}, y el atributo del aprobado (que puede estar contenido en un registro de base de datos) puede tener el valor {*OID-D* 3}. La comparación del valor de atributo correspondiente al tipo de atributo del que aprueba (en este ejemplo, 3) con el valor de atributo correspondiente al tipo de atributo aprobado (en este ejemplo también 3) da como resultado FALSO para la expresión "NOT lessOrEqual", y así se niega al primer director la

capacidad de aprobar un aumento salarial para el segundo director. Por otra parte, si el atributo del aprobado fuera {*OID-D* 1}, se atribuiría al director la capacidad de aprobar el aumento al gestor.

No resulta difícil concebir adiciones útiles a la expresión anterior. Por ejemplo, se podría añadir un tercer componente al **'and'** que indique que el entorno variable "currentTime", leído a partir del reloj local y codificado después como un atributo del tipo OBJECT ID *OID-E*, tendrá que estar dentro de un periodo de tiempo determinado especificado explícitamente en la expresión como un atributo del tipo OBJECT ID *OID-F*. Así, por ejemplo, se pueden permitir actualizaciones salariales, únicamente si se satisfacen las condiciones anteriores y si la petición tiene lugar durante horas laborables.

D.2.2 Segundo ejemplo

Una política de seguridad en su forma más simple es un conjunto de criterios para la prestación de servicios de seguridad. En lo que respecta al control de acceso la política de seguridad es un subconjunto de una política de seguridad de sistema de nivel más alto, que define los medios para asegurar políticas de control de acceso entre iniciadores y objetivos. Los mecanismos de control de acceso necesitan permitir la comunicación cuando lo permita una política específica y denegar la comunicación cuando una política específica no lo permita explícitamente.

Una política de seguridad constituye la base para que los mecanismos de control de acceso tomen decisiones. La información de política de seguridad específica de dominio se transmite a través del fichero de información de política de seguridad (SPIF, *security policy information file*).

El SPIF es un objeto firmado para protegerlo de modificaciones no autorizadas. El SPIF contiene información utilizada para interpretar los parámetros de control de acceso contenidos en una etiqueta de seguridad y en el atributo de acreditación. El identificador de política de seguridad que aparece en el atributo de acreditación necesita estar asociado con una sintaxis y una semántica de implementación específica, como se define en la política de seguridad. Esta sintaxis de implementación asociada con una política de seguridad específica se mantiene en un SPIF.

El SPIF transmite equivalencias entre autorizaciones y sensibilidades a través de los dominios de política de seguridad como se determina en las políticas de seguridad, proporciona una representación de las etiquetas de seguridad que se pueden imprimir y hace corresponder ristras que se pueden presentar con niveles y categorías de seguridad para su presentación a usuarios finales al seleccionar un atributo de seguridad de objeto de datos. Las correspondencias de equivalencia se expresan de tal forma que una etiqueta generada en un dominio de política de seguridad lo puede interpretar correctamente una aplicación que funcione en otro domino de política de seguridad. El SPIF también hace corresponder el atributo de acreditación con los campos de etiqueta de seguridad de mensaje y las etiquetas de presentación que deberán presentarse al usuario. Esta correspondencia, si tiene éxito, verifica que el receptor pretendido tiene las autorizaciones adecuadas para aceptar el objeto de datos.

El SPIF contiene una secuencia de lo siguiente:

- **versionInformation** – Indica la versión de la sintaxis ASN.1.
- **updateInformation** – Indica la versión de la sintaxis y de la semántica de la especificación de SPIF.
- **securityPolicyIdData** – Identifica la política de seguridad a la que aplica el SPIF.
- **privilegeId** – Indica el OID que identifica la sintaxis que está incluida en la categoría de seguridad de atributo de acreditación.
- **rbacId** – Identificador de objeto que identifica la sintaxis de la categoría de seguridad que se utiliza junto con el SPIF.
- **securityClassifications** – Hace corresponder la clasificación de la etiqueta de seguridad con una clasificación en el atributo de acreditación y también proporciona correspondencias de equivalencias.
- **securityCategoryTagSets** – Hace corresponder las categorías de seguridad de la etiqueta de seguridad con las categorías de seguridad en el atributo de acreditación y también proporciona correspondencias de equivalencias.
- **equivalentPolicies** – Consolida todas las políticas equivalentes en el SPIF.
- **defaultSecurityPolicyIdData** – Identifica la política de seguridad que se puede aplicar si los datos se reciben sin una etiqueta de seguridad.
- **extensions** – Proporciona un mecanismo para incluir capacidades adicionales cuando se identifiquen requisitos futuros.

El archivo de información de política de seguridad se define en la sintaxis siguiente:

```
SecurityPolicyInformationFile ::= SIGNED {SPIF}
SPIF ::=  SEQUENCE    {
    versionInformation              VersionInformationData DEFAULT v1,
    updateInformation               UpdateInformationData,
```

```
    securityPolicyIdData                ObjectIdData,
    privilegeId                         OBJECT IDENTIFIER,
    rbacId                              OBJECT IDENTIFIER,
    securityClassifications     [0]     SEQUENCE OF SecurityClassification OPTIONAL,
    securityCategories          [1]     SEQUENCE OF SecurityCategory  OPTIONAL,
    equivalentPolicies          [2]     SEQUENCE OF EquivalentPolicy  OPTIONAL,
    defaultSecurityPolicyIdData [3]     ObjectIdData OPTIONAL,
    extensions                  [4]     Extensions OPTIONAL }

VersionInformationData ::= INTEGER { v1(0) }


UpdateInformationData  ::=  SEQUENCE    {
    sPIFVersionNumber               INTEGER,
    creationDate                    GeneralizedTime,
    originatorDistinguishedName     Name,
    keyIdentifier                   OCTET STRING OPTIONAL }

ObjectIdData ::=   SEQUENCE  {
    objectId            OBJECT IDENTIFIER,
    objectIdName        DirectoryString {ubObjectIdNameLength} }

SecurityClassification  ::=  SEQUENCE  {
    labelAndCertValue                   INTEGER,
    classificationName                  DirectoryString {ubClassificationNameLength},
    equivalentClassifications   [0]     SEQUENCE OF EquivalentClassification OPTIONAL,
    hierarchyValue                      INTEGER,
    markingData                 [1]     SEQUENCE OF MarkingData OPTIONAL,
    requiredCategory            [2]     SEQUENCE OF OptionalCategoryGroup OPTIONAL,
    obsolete                            BOOLEAN DEFAULT FALSE           }

EquivalentClassification ::= SEQUENCE {
    securityPolicyId            OBJECT IDENTIFIER,
    labelAndCertValue           INTEGER,
    applied                     INTEGER {
                                    encrypt  (0),
                                    decrypt  (1),
                                    both     (2) } }

MarkingData ::= SEQUENCE {
    markingPhrase       DirectoryString {ubMarkingPhraseLength} OPTIONAL,
    markingCodes        SEQUENCE OF MarkingCode OPTIONAL }

MarkingCode ::= INTEGER {
    pageTop                 (1),
    pageBottom              (2),
    pageTopBottom           (3),
    documentEnd             (4),
    noNameDisplay           (5),
    noMarkingDisplay        (6),
    unused                  (7),
    documentStart           (8),
    suppressClassName       (9)}

OptionalCategoryGroup ::= SEQUENCE {
    operation               INTEGER {
                                onlyOne         (1),
                                oneOrMore       (2),
                                all             (3)},
    categoryGroup           SEQUENCE OF OptionalCategoryData }

OptionalCategoryData ::= SEQUENCE {
    optCatDataId            OC-DATA.&id({CatData}),
    categorydata            OC-DATA.&Type({CatData}{@optCatDataId }) }

OC-DATA ::= TYPE-IDENTIFIER

CatData OC-DATA ::= {  ... }

EquivalentPolicy   ::= SEQUENCE {
    securityPolicyId            OBJECT IDENTIFIER,
    securityPolicyName          DirectoryString {ubObjectIDNameLength}
    OPTIONAL}
```

```
Extensions ::= SEQUENCE OF Extension

Extension ::= SEQUENCE {
    extensionId             EXTENSION.&objId ({ExtensionSet}),
    critical                BOOLEAN DEFAULT FALSE,
    extensionValue              OCTET STRING  }
```

Hay que destacar que el ejemplo de SPIF es una sintaxis que evoluciona y que una definición y descripción de cada elemento se encuentra en los objetos de información de seguridad de Rec. UIT-T X.841 | ISO/CEI 15816.

D.3 Ejemplo de atributo de privilegios

El ejemplo siguiente de un atributo que transmite un determinado privilegio se proporciona únicamente para ilustración. La especificación real de esta sintaxis y del atributo asociado está incluida en 19.5 de la Rec. UIT-T X.501 | ISO/CEI 9594-2. Este atributo en particular transfiere la acreditación que se puede asociar con una entidad denominada, incluido un DUA para fines de comunicación de DSA.

Un atributo de acreditación asocia una acreditación con una entidad denominada, incluidos DUA.

```
clearance  ATTRIBUTE ::=  {
    WITH SYNTAX             Clearance
    ID                      id-at-clearance }

Clearance  ::=  SEQUENCE {
    policyId                OBJECT IDENTIFIER,
    classList               ClassList DEFAULT {unclassified},
    securityCategories      SET SIZE (1MAX) OF SecurityCategory  OPTIONAL}

ClassList  ::=  BIT STRING {
    unmarked            (0),
    unclassified        (1),
    restricted          (2),
    confidential        (3),
    secret              (4),
    topSecret           (5) }
```

Los componentes individuales se describen con la especificación real de este privilegio en el documento referenciado.

Anexo E

Introducción a la criptografía de claves públicas[3]

(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 (Xp) se conoce públicamente, y puede ser usada por cualquier usuario para cifrar datos. Solamente X, quien posee la clave privada complementaria (Xs), puede descifrar los datos. (Esto se representa por la notación D = Xs[Xp[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 Xp. 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 E.1.

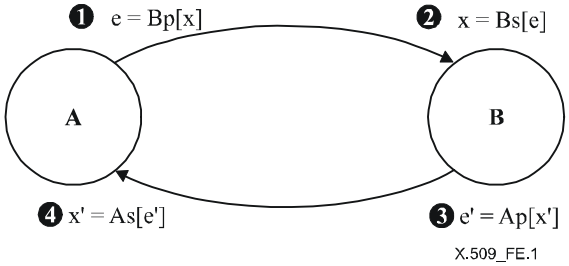


Figura E.1 – Uso de un PKCS para intercambiar información secreta

El usuario A tiene la clave pública Ap y la clave privada As, y el usuario B tiene otro conjunto de claves, Bp y Bs. 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 E.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:

$$e = Bp[x]$$

2) B puede ahora descifrar este cifrado e para obtener la información x usando la clave secreta de descifrado Bs. Obsérvese que B es el único posesor de Bs, 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 Bs determina la identidad de B. La operación de descifrado se representa por:

$$x = Bs[e], \text{ o } x = Bs[Bp[x]]$$

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

$$e' = Ap[x']$$

4) A obtiene x' descifrando e':

---

[3] Para más información, véase:

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

$$x' = As[e'], \text{ o } x' = As[Ap[x']]$$

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.

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, As y Bs respectivamente. A puede determinar si B está en posesión de la clave secreta de descifrado, Bs, 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 Bs. 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 = Xp[Xs[D]]. Esto permite que una información que pudiera haber sido originada solamente por X sea legible por cualquier usuario (que esté en posesión de Xp). 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 F

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-itu-t ds(5) module(1) algorithmObjectIdentifiers(8) 5}
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-itu-t ds(5) module(1) usefulDefinitions(0) 5}
    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 --

rsaALGORITHM    ::=  {
    KeySize
    IDENTIFIED BYid-ea-rsa }

KeySize         ::=  INTEGER

-- the following object identifier assignments reserve values assigned to deprecated functions

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

END
```

Anexo G

Ejemplos de la utilización de constricciones del trayecto de certificación

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

G.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 }

G.2 Ejemplo 2: 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)}}

G.3 Utilización de la extensión de constricciones de nombre

G.3.1 Ejemplos de un formato de certificado con la extensión de constricciones de nombre

Las CA pueden imponer diversas restricciones a los nombres de los sujetos (en el campo **subject** o en la extensión **subjectAltName**) de los certificados que expiden y de los certificados subsiguientes del trayecto de certificación, por inclusión de la extensión de constricciones de nombre a sus certificados de CA. En esta cláusula se describen ejemplos del formato del certificado con la extensión de constricciones de nombre.

Para simplificar los ejemplos, las formas de nombre requeridas (**requiredNameForms**) por la extensión de constricciones de nombre indican sólo el nombre rfc822 (**rfc822Name**) y el DN (**directoryName**).

G.3.1.1 Ejemplos de *permittedSubtrees* (*subárboles permitidos*)

(1-1) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN, si existe, será igual o subordinado al de Acme Inc. de Estados Unidos (es decir, {C=US, O=Acme Inc}).

extensión **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{base(directoryName) {C=US, O=Acme Inc}}}	(vacío)	(vacío)

(1-2) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN, si existe, debe ser igual o inmediatamente subordinado al de Acme Inc. de Estados Unidos (es decir, {C=US, O=Acme Inc}).

extensión **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{base(directoryName) {C=US, O=Acme Inc}, **maximum 1**}}	(vacío)	(vacío)

(1-3) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN, si existe, debe ser subordinado al de Acme Inc. de Estados Unidos (es decir, {C=US, O=Acme Inc}).

extensión **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base(directoryName)** {C=US, O=Acme Inc}, **minimum 1**}}	(vacío)	(vacío)

(1-4) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN, si existe, debe ser igual o subordinado al de Acme Inc. de Estados Unidos (es decir, {C=US, O=Acme Inc}), o igual o subordinado al de Acme Ltd. del Reino Unido (es decir, {C=UK, O=Acme Ltd}).

extensión **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base(directoryName)** {C=US, O=Acme Inc}}, {**base(directoryName)** {C=UK, O=Acme Ltd}}}	(vacío)	(vacío)

G.3.1.2 Ejemplos de *excludedSubtrees* (*subárboles excluidos*)

(2-1) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN, si existe, no debe ser igual ni subordinado al de Acme Corp. de Canadá (es decir, {C=CA, O=Acme Corp}).

extensión **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
(vacío)	{{**base(directoryName)** {C=CA, O=Acme Corp}}}	(vacío)

(2-2) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN, si existe, no debe ser subordinado a cada subordinado inmediato de Acme Corp. de Canadá (es decir, {C=CA, O=Acme Corp}).

extensión **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
(vacío)	{{**base(directoryName)** {C=CA, O=Acme Corp}, **minimum 2**}}	(vacío)

(2-3) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN, si existe, no debe ser igual al de Acme Corp. de Canadá (es decir, {C=CA, O=Acme Corp}).

extensión **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
(vacío)	{{**base(directoryName)** {C=CA, O=Acme Corp}, **maximum 0**}}	(vacío)

(2-4) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN, si existe, no debe ser igual ni subordinado al de Acme Corp. de Canadá (es decir, {C=CA, O=Acme Corp}), ni igual ni subordinado al de Acme Asia de Japón (es decir, {C=JP, O=Asia Acme}).

extensión **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
(vacío)	{{**base(directoryName)** {C=CA, O=Acme Corp}}, {**base(directoryName)** {C=JP, O=Asia Acme}}}	(vacío)

G.3.1.3 Ejemplos de permittedSubtrees y excludedSubtrees

(3-1) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN, si existe, debe ser igual o subordinado al de Acme Inc. de Estados Unidos (es decir, {C=US, O=Acme Inc}) excepto la unidad de organización de investigación y desarrollo (R&D) de Acme Inc. y los subordinados de dicha organización.

extensión **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base(directoryName)** {C=US, O=Acme Inc}}}	{{**base(directoryName)** {C=US, O=Acme Inc, OU=R&D}}}	(vacío)

(3-2) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN, si existe, debe ser igual al de uno de los subordinados inmediatos de Acme Inc. de Estados Unidos (es decir, {C=US, O=Acme Inc}) excepto la unidad de organización de adquisiciones (es decir, {C=US, O=Acme Inc, OU=Purchasing}).

extensión **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base(directoryName)** {C=US, O=Acme Inc}, **minimum 1, maximum 1**}}	{{base(directoryName) {C=US, O=Acme Inc, OU=Purchasing}}}	(vacío)

G.3.1.4 Ejemplos de permittedSubtrees y excludedSubtrees con requiredNameForms

(4-1) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, al menos uno de los nombres de sujeto (en el campo **subject** o en la extensión **subjectAltName**) del certificado estará en la forma de nombre DN. No obstante, cada nombre de sujeto no está constreñido por espacios de nombre.

extensión **nameConstraints**			
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**	
		nombre rfc822	DN
(no válido)	(no válido)	DESACTIVADO	ACTIVADO

(4-2) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, al menos uno de los nombres de sujeto (en el campo **subject** o en la extensión **subjectAltName**) debe tener la forma de nombre DN. Además, cada nombre de sujeto en la forma de nombre DN debe satisfacer los espacios de nombres constreñidos por **permittedSubtrees** y **excludedSubtrees**.

extensión **nameConstraints**			
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**	
		nombre rfc822	DN
{{**base(directoryName)** {C=JP, O=Asia Acme}}}	{{**base(directoryName)** {C=JP, O=Asia Acme, OU=Marketing}}}	DESACTIVADO	ACTIVADO

(4-3) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN, si existe, debe satisfacer los espacios de nombre constreñidos por **permittedSubtrees** y **excludedSubtrees**.

extensión **nameConstraints**			
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**	
		nombre rfc822	DN
{{**base(directoryName)** {C=JP, O=Asia Acme}}}	{{**base(directoryName)** {C=JP, O=Asia Acme, OU=Marketing}}}	DESACTIVADO	DESACTI-VADO

NOTA – El ejemplo anterior de un certificado de CA es compatible con el siguiente certificado de CA con la extensión de constricciones de nombre sin el elemento **requiredNameForms**.

extensión **nameConstraints**		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base(directoryName)** {C=JP, O=Asia Acme}}}	{{**base(directoryName)** {C=JP, O=Asia Acme, OU=Marketing}}}	(vacío)

(4-4) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN, si existe, debe satisfacer los espacios de nombre constreñidos por **permittedSubtrees** y **excludedSubtrees**. Además, debe estar presente al menos un **subjectAltName** en la forma de nombre **rfc822Name**, aunque su nombre no esté constreñido por espacios de nombre.

extensión**nameConstraints**			
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**	
		nombre rfc822	DN
{{**base(directoryName)** {C=JP, O=Asia Acme}}}	{{**base(directoryName)** {C=JP, O=Asia Acme, OU=Marketing}}}	ACTIVADO	DESACTI-VADO

(4-5) Si el certificado de la CA contiene la siguiente extensión de constricciones de nombre, en todos los certificados subsiguientes del trayecto de certificación, al menos uno de los nombres de sujeto (en el campo **subject** o en la extensión **subjectAltName**) del certificado debe tener la forma de nombre DN o rfc822. Cada nombre de sujeto en la forma de nombre DN, si existe, debe satisfacer los espacios de nombre constreñidos por **permittedSubtrees** y **excludedSubtrees**. Cada nombre de sujeto en la forma de nombre **rfc822Name** no está constreñido por espacios de nombre.

extensión **nameConstraints**			
permittedSubtrees	**ExcludedSubtrees**	**requiredNameForms**	
		nombre rfc822	DN
{{**base(directoryName**) {C=JP, O=Asia Acme}}}	{{**base(directoryName**) {C=JP, O=Asia Acme, OU=Marketing}}}	ACTIVADO	ACTIVADO

G.3.2 Ejemplo de tratamiento de los certificados con la extensión de constricciones de nombre

En esta cláusula se describen algunos ejemplos de la forma en que se valida el nombre del sujeto (en el campo **subject** o en la extensión **subjectAltName**) durante el procesamiento del certificado con las variables de estado de procesamiento de trayecto, concretamente *subárboles permitidos, subárboles excluidos y formas de nombre requeridas.*

Para simplificar los ejemplos, la variable de estado de procesamiento de trayecto *formas de nombre requeridas* indica sólo el nombre rfc822 (**rfc822Name**), el DN (**directoryName**) y el URI (**uniformResourceIdentifier**).

G.3.2.1 Constricciones de espacios de nombre mediante *subárboles permitidos* en la forma de nombre DN

En este caso, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN que aparece en el certificado en cuestión debe satisfacer la constricción mediante la variable de estado de procesamiento de trayecto *subárboles permitidos*.

(1-1) Un subárbol permitido para DN está presente y se requiere DN en *formas de nombre requeridas.*

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
{{**base**(**directoryName**) {C=US, O=Acme Inc}}}	NINGUNO	DESAC-TIVADO	ACTI-VADO	DESAC-TIVADO

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing}
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName** (**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {} **subjectAltName** (**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName** (**rfc822Name**) = manager@purchasing.acme.com
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName** (**directoryName**) = {C=US, O=Acme Inc, OU=Accounting}

Ejemplos de certificados inaceptables

1	**subject** = {C=US, O=*Acme Ltd*,OU=Purchasing}
2	**subject** = {} **subjectAltName** (**rfc822Name**) = manager@purchasing.acme.com NOTA – *Falta el DN*
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName** (**directoryName**) = {C=US, O=*Acme Ltd*, OU=Purchasing}
4	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName** (**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing}
5	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName** (**directoryName**) = {C=US, O=*Acme Ltd*, OU=Accounting}

(1-2) Dos subárboles permitidos para DN están presentes y se requiere DN en *formas de nombre requeridas*.

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
{{**base(directoryName)** {C=US, O=Acme Inc}}, {**base(directoryName)** {C=US, O=Acme Ltd}}}	NINGUNO	DESAC-TIVADO	ACTI-VADO	DESAC-TIVADO

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Purchasing}
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
5	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU= Accounting}

Ejemplos de certificados inaceptables

1	**subject** = {C=US, O=*Acme International*, OU=Accounting}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com NOTE – *Falta el DN*
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme International*, OU=Accounting}
4	**subject** = {C=US, O=*Acme International*, OU=Accounting} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing}
5	**subject** = {C=US, O=*Acme International*, OU=Accounting} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Corp*, OU=Accounting}
6	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme International*, OU=Accounting} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com

(1-3) Un subárbol permitido para DN está presente y *formas de nombre requeridas* está vacío.

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
{{**base(directoryName)** {C=US, O=Acme Inc}}}	NINGUNO	vacío		

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing}
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Accounting}
6	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com

Ejemplos de certificados inaceptables

1	**subject** = {C=US, O=*Acme Ltd*,OU=Purchasing}
2	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Ltd*, OU=Accounting}
3	**subject** = {C=US, O=*Acme Ltd*, OU=Accounting} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing}
4	**subject** = {C=US, O=*Acme Ltd*, OU=Accounting} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Ltd*, OU=Purchasing}

G.3.2.2 Constricciones de espacios de nombre mediante *subárboles excluidos* en la forma de nombre DN

En este caso, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN que aparece en el certificado en cuestión debe satisfacer la constricción mediante la variable de estado de procesamiento de trayecto *subárboles excluidos*.

(2-1) Un subárbol excluido para DN está presente y se requiere DN en *formas de nombre requeridas*.

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
NINGUNO	{{**base**(**directoryName**) {C=US, O=Acme Ltd}}}	DESAC-TIVADO	ACTI-VADO	DESAC-TIVADO

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing}
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Accounting}

Ejemplos de certificados inaceptables

1	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com NOTA – *Falta el DN*
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Ltd*, OU=Accounting}

(2-2) Dos subárboles excluidos para DN están presentes y se requiere DN en *formas de nombre requeridas*.

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
NINGUNO	{{**base(directoryName)** {C=US, O=Acme Inc}}, {**base(directoryName)** {C=US, O=Acme Ltd}}}	DESAC-TIVADO	ACTI-VADO	DESAC-TIVADO

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme International, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme International, OU=Purchasing}
3	**subject** = {C=US, O=Acme International, OU=Purchasing} **subjectAltName**(**rfc822Name**) = purchasing@acme-international.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme International, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme N.Y, OU=Purchasing} **subjectAltName**(**rfc822Name**) = purchasing@acme-international.com

Ejemplos de certificados inaceptables

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
3	**subject** = {} **subjectAltName**(**rfc822Name**) = purchasing@acme-international.com NOTA – *Falta el DN*
4	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme International, OU=Accounting}
5	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme International, OU=Purchasing} **subjectAltName**(**rfc822Name**) = purchasing@acme-international.com

(2-3) Un subárbol excluido para DN está presente y *formas de nombre requeridas* está vacío.

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
NINGUNO	{{**base(directoryName)** {C=US, O=Acme Inc}}}	vacío		

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Purchasing}
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
5	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Accounting}
6	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com

Ejemplos de certificados inaceptables

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Accounting}

G.3.2.3 Constricciones de espacios de nombre sólo mediante formas de nombre requeridas

(3-1) Se requiere DN en *formas de nombre requeridas*.

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
NINGUNO	NINGUNO	DESAC-TIVADO	ACTI-VADO	DESAC-TIVADO

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=JP, O=Acme Inc, OU=Purchasing}
3	**subject** = {C=JP, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
5	**subject** = {C=JP, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Accounting}

Ejemplos de certificados inaceptables

1	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com NOTE – *Falta el DN*
2	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-ltd.com NOTA – *Falta el DN*

(3-2) Se requiere DN o **rfc822Name** en *formas de nombre requeridas*.

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
NINGUNO	NINGUNO	ACTI-VADO	ACTI-VADO	DESAC-TIVADO

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**directoryName**) = {C=JP, O=Acme Inc, OU=Purchasing}
3	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
5	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com **subjectAltName**(**rfc822Name**) = purchasing@acme-ltd.com

Ejemplos de certificados inaceptables

1	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-ltd.com NOTA – *Falta el DN y el rfc822*
2	**subject** = {} **subjectAltName**(**dNSName**) = www.acme-ltd.com NOTE – *Falta el DN y el rfc822*

G.3.2.4 Constricciones de espacios de nombre mediante *subárboles permitidos* en formas de nombre múltiples

En este caso, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN o rfc822 que aparece en el certificado en cuestión debe satisfacer la constricción mediante la variable de estado de procesamiento de trayecto *subárboles permitidos*.

(4-1) Un subárbol permitido para DN y otro permitido para **rfc822Name** están presentes. Además, se requiere DN en *formas de nombre requeridas*.

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**rfc822Name**) .acme.com}}	NINGUNO	DESACTIVADO	ACTIVADO	DESAC-TIVADO

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Inc, OU=Accounting}
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com

Ejemplos de certificados inaceptables

1	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com NOTA – *Falta el DN*
3	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-inc.com*
5	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-ltd.com*
6	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com NOTA – *Falta el DN*

(4-2) Un subárbol permitido para DN y otro subárbol permitido para **rfc822Name** están presentes. Además, se requiere al menos uno de DN o de **rfc822Name** en *formas de nombre requeridas*.

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**rfc822Name**) .acme.com}}	NINGUNO	ACTI-VADO	ACTI-VADO	DESAC-TIVADO

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**directoryName**) = { C=US, O=Acme Inc, OU=Accounting}
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com

Ejemplos de certificados inaceptables

1	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-inc.com*
3	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-inc.com*
5	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-ltd.com*
6	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com NOTA – *Falta el DN y rfc822*

(4-3) Un subárbol permitido para DN y otro permitido para **rfc822Name** están presentes. No se requieren formas de nombre en *formas de nombre requeridas*.

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*Subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**rfc822Name**) .acme.com}}	NINGUNO	vacío		

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme.com
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme.com

Ejemplos de certificados inaceptables

1	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-inc.com*
3	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-inc.com*
5	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme-inc.com*

G.3.2.5 Constricciones de espacios de nombre mediante *subárboles excluidos* en formas de nombres múltiples

En este caso, cada nombre de sujeto (en el campo **subject** o en la extensión **subjectAltName**) en la forma de nombre DN o rfc822 que aparece en el certificado en cuestión debe satisfacer la constricción mediante la variable de estado de procesamiento de trayecto *subárboles excluidos*.

(5-1) Un subárbol excluido para DN y otro excluido para **rfc822Name** están presentes. Además, se requiere DN en *formas de nombre requeridas*.

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
NINGUNO	{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**rfc822Name**) .acme.com}}	DESAC-TIVADO	ACTI-VADO	DESAC-TIVADO

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Accounting}
4	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-ltd.com

Ejemplos de certificados inaceptables

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.com NOTA – *Falta el DN*
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
4	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-inc.com
5	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
6	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Inc*, OU=Accounting}
7	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com NOTE – *Falta el DN*

(5-2) Un subárbol excluido para DN y otro excluido para **rfc822Name** están presentes. Además, se requiere al menos un DN o **rfc822Name** en *formas de nombre requeridas*.

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
NINGUNO	{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**rfc822Name**) .acme.com}}	ACTI-VADO	ACTI-VADO	DESAC-TIVADO

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme.org
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
4	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=Acme Ltd, OU=Accounting}
5	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-ltd.com

Ejemplos de certificados inaceptables

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
4	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-inc.com
5	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
6	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Inc*, OU=Accounting}
7	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com NOTA – *Falta el DN y rfc822*

(5-3) Un subárbol excluido para DN y otro excluido para **rfc822Name** están presentes. No se requieren formas de nombre en *formas de nombre requeridas*.

Variables de estado de procesamiento de trayecto				
subárboles permitidos	*subárboles excluidos*	*formas de nombre requeridas*		
		rfc822	DN	URI
NINGUNO	{{**base**(**directoryName**) {C=US, O=Acme Inc}}, {**base**(**rfc822Name**) .acme.com}}	vacío		

Ejemplos de certificados aceptables

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-inc.com
5	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**uniformResourceIdentifier**) = http://purchasing.www.acme-ltd.com

Ejemplos de certificados inaceptables

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
4	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = manager@purchasing.acme-inc.com
5	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName**(**rfc822Name**) = *manager@purchasing.acme.com*
6	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName**(**directoryName**) = {C=US, O=*Acme Inc*, OU=Accounting}

Anexo H

Orientación para determinar para qué políticas es válido un trayecto de certificación

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

La finalidad de este anexo es ofrecer orientación sobre las aplicaciones habilitadas en la PKI relativas al control de la política de certificados en el procesamiento de la validación del trayecto de certificados. El control de la política de certificados en el procesamiento del contenido de los certificados que realiza la PKI se describe en la parte de la Especificación que trata el procedimiento de procesamiento del trayecto del certificado.

El presente anexo trata la inicialización de dos de las entradas relativas a política del procedimiento de procesamiento del trayecto: *conjunto de políticas inicial* y *política explícita inicial*. Además de éstas, las entradas *inhibición de correspondencia de políticas inicial* y *cualquier política de inhibición inicial* al procedimiento, que también pueden ser inicializadas por el usuario, repercuten en el procesamiento de la información relativa a política durante el procesamiento del trayecto, aunque éstas quedan fuera del alcance de este anexo. La fijación de *inhibición de correspondencia de políticas inicial* a **VERDADERO** impide que se utilicen correspondencias de políticas en validaciones de trayecto satisfactorias. La fijación de *cualquier política de inhibición inicial* a **VERDADERO** impide que el OID especial para **anyPolicy**, si está presente en un certificado, constituya una correspondencia aceptable para un OID de política específica.

Los términos de "usuario" podrán designar a un "usuario humano" o una "aplicación" habilitada para PKI.

Se prevén los siguientes escenarios:

1) El usuario exige que el trayecto de certificación sea válido para una de las políticas que le interesan.
2) El usuario exige que el trayecto de certificación sea válido al menos para una política, pero sin preocuparse de qué política se trata. Este escenario podría ser útil cuando el usuario pretende realizar un procesamiento de política adicional empleando otra información contextual y otro contenido de información, a fin de determinar si una de las políticas para la que es válido el trayecto de certificación puede ser aceptable para el usuario en la transacción considerada.
3) El usuario no tiene requisitos relativos a la política en el trayecto de certificación. En otras palabras, el usuario está dispuesto a aceptar un trayecto de certificación que no es válido para ninguna política, pero que por lo demás es válido.
4) El usuario desea que el trayecto de certificación sea válido para una de las políticas que le interesan, pero si así no fuera, desea tener la oportunidad de reconsiderar trayectos que no son válidos para las políticas que le interesan. Este escenario podría ser útil cuando el usuario necesita, en general, que el trayecto de certificación sea válido para una política que le sea aceptable, pero basándose en otra información contextual y en otro contenido de información, el usuario puede desear pasar por alto el fallo de política.

En las cláusulas que siguen se describe como debe proceder el usuario para obtener la información deseada de una máquina de validación de trayecto conforme.

H.1 Trayecto de certificación válido para una política requerida especificada por el usuario

En este escenario, el usuario exige que el trayecto de certificación sea válido para una de las políticas que le interesan. Para obtener la información deseada, el usuario debe fijar las entradas de validación del trayecto de certificación relativas al procesamiento de política de la siguiente manera:

conjunto de políticas inicial = {conjunto de políticas que le interesan al usuario}

política explícita inicial = **VERDADERO**

Si la validación del trayecto resulta satisfactoria, el trayecto de certificación es válido al menos para una de las políticas que le interesan al usuario. El trayecto de certificación es válido para las políticas enumeradas en la variable de salida *conjunto de políticas constreñidas por el usuario*.

En este escenario, las aplicaciones no deben utilizar un trayecto de certificación que sea rechazado por una máquina de validación de trayecto por fallos relativos a la política de certificados[4].

H.2 Trayecto de certificación válido para cualquier política requerida

En este escenario, el usuario exige que el trayecto de certificación sea válido al menos para una política, sin preocuparse de qué política se trate. Para obtener la información deseada, el usuario debe fijar las entradas de validación del trayecto de certificación relativas al procesamiento de política de la siguiente manera:

conjunto de políticas inicial = **{anyPolicy}**

política explícita inicial = **VERDADERO**

Si la validación del trayecto resulta satisfactoria, el trayecto de certificación es válido al menos para una política. El trayecto de certificación es válido para las políticas enumeradas en la variable de salida *conjunto de políticas constreñidas por el usuario*.

En este escenario, las aplicaciones no deben utilizar un trayecto de certificación que sea rechazado por una máquina de validación de trayecto por fallos relativos a la política de certificados.

H.3 Trayecto de certificación válido independientemente de la política

En este escenario, el usuario no tiene requisitos relativos a la política en el trayecto de certificación. Para obtener la información deseada, el usuario debe fijar las entradas de validación del trayecto de certificación relativas al procesamiento de política de la siguiente manera:

conjunto de políticas inicial = {**anyPolicy**}

política explícita inicial = **FALSO**

Si la validación del trayecto resulta satisfactoria, el trayecto de certificación es válido para las políticas enumeradas en la variable de salida *conjunto de políticas constreñidas por el usuario*.

En este escenario, las aplicaciones no deben utilizar un trayecto de certificación que sea rechazado por una máquina de validación de trayecto por fallos relativos a la política de certificados.

Cabe señalar que en este escenario, el trayecto de certificación puede tener un fallo relativo a la política. Un ejemplo sería cuando la infraestructura (es decir, un certificado de CA en el trayecto de certificación) provoca la fijación del *indicador política explícita*. En este caso, si el trayecto no es válido para ninguna política, es decir, el *conjunto de políticas constreñidas por las autoridades* está vacío, una máquina de validación de trayecto conforme devolverá un fallo. Cuando se presenta este tipo de fallo las aplicaciones deben rechazar el trayecto de certificación.

H.4 Trayecto de certificación válido para una política específica deseada por el usuario, aunque no se requiere

En este escenario, el usuario desea que el trayecto de certificación sea válido para una de las políticas que le interesan, pero no desea rechazar trayectos que no son válidos para alguna de las políticas que le interesan. Para obtener la información deseada, el usuario debe fijar las entradas de validación del trayecto de certificación relativas al procesamiento de política de la siguiente manera:

conjunto de políticas inicial = {conjunto de políticas que le interesan al usuario}

política explícita inicial = **FALSO**

Si la validación del trayecto resulta satisfactoria, el trayecto de certificación es válido para las políticas enumeradas en la variable de salida *conjunto de políticas constreñidas por el usuario*. Este conjunto es un subconjunto del *conjunto de políticas inicial*. Cabe señalar que en este caso el *conjunto de políticas constreñidas por el usuario* podría ser **NULO**

---

[4] Un fallo de validación de trayecto es un fallo relativo a la política de certificados cuando éste es provocado por las extensiones o por las variables de estado relativas a la política de certificados. Las extensiones relativas a la política de certificados son: **certificatePolicies**, **policyMappings**, **policyConstraints** y **inhibitAnyPolicy**. Las variables de estado relacionadas con la política de los certificados son: *conjunto de políticas constreñidas por las autoridades, indicador política explícita, indicador inhibición de correspondencia de políticas e indicador inhibición de cualquier política.*

cuando no está fijado el *indicador política explícita*. La aplicación debería examinar el *conjunto de políticas constreñidas por el usuario* para determinar si el trayecto es aceptable para el usuario.

En este caso, las aplicaciones deben rechazar el trayecto de certificación cuando la infraestructura provoca un fallo relativo a la política, (es decir, cuando el *conjunto de políticas constreñidas por las autoridades* está vacío y está fijado el *indicador política explícita*).

Cabe señalar que en este escenario, el trayecto de certificación puede tener un fallo relativo a la política. Un ejemplo sería cuando la infraestructura (es decir, un certificado de CA en el trayecto de certificación) provoca la fijación del *indicador política explícita*. En este caso, si el trayecto no es válido para ninguna política, es decir, el *conjunto de políticas constreñidas por las autoridades* está vacío, una máquina de validación de trayecto conforme devolverá un fallo. Las aplicaciones deben rechazar el trayecto de certificación si se produce un fallo de este tipo.

Otro ejemplo es cuando la combinación de la entrada del usuario y la infraestructura producen un fallo relativo a la política. Esto ocurre cuando un certificado de CA en el trayecto de certificación provoca la fijación del *indicador política explícita*, que el *conjunto de políticas constreñidas por las autoridades* no esté vacío y que el *conjunto de políticas constreñidas por el usuario* esté vacío. La máquina de validación de trayecto conforme devolverá un fallo. En estas condiciones, si el único motivo por el que la máquina de validación de trayecto devuelve un fallo es que el *conjunto de políticas constreñidas por el usuario* está vacío, las aplicaciones pueden tomar la decisión de pasar por alto dicho fallo y aceptar el trayecto de certificación. Las limitaciones impuestas por la autoridad se siguen respetando, en virtud de que el *conjunto de políticas constreñidas por las autoridades* no está vacío. La aceptación de este trayecto por una aplicación equivale a que la aplicación envíe nuevamente el trayecto a la máquina de validación con el *conjunto de políticas inicial* igual a **anyPolicy** y la *política explícita inicial* igual a **FALSO**, y a examinar el *conjunto de políticas constreñidas por el usuario* para determinar si el trayecto es aceptable.

Anexo I

Cuestiones relativas a la extensión del certificado de utilización de claves

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

Combinar el bit contentCommitment en la extensión del certificado keyUsage con otros bits keyUsage puede tener implicaciones de seguridad dependiendo del entorno de seguridad en el que vaya a utilizarse el certificado. Si el entorno del sujeto puede controlarse plenamente y se puede confiar en él, no hay riesgo de implicaciones de seguridad específicas. Por ejemplo, en los casos en que el sujeto sabe exactamente qué datos están firmados o en los casos en que está completamente seguro de las características de seguridad del protocolo de autenticación que se está utilizando. Si el entorno del sujeto no está plenamente controlado o no hay plena confianza en él, es posible la firma de compromisos no intencionada. Ejemplos son la utilización de intercambios de autenticación mal formados y el empleo de un componente de software ilegal. Si un sujeto utiliza entornos que no son de confianza, las repercusiones de seguridad correspondientes podrán limitarse mediante la adopción de las siguientes medidas:

- no combinar la fijación de la utilización de la clave contentCommitment en certificados con cualquier otra fijación de utilización de clave y aplicar la clave privada correspondiente sólo con este certificado;
- limitar el empleo de claves privadas asociadas con certificados que tengan fijado el bit de utilización de clave contentCommitment, a entornos que se consideren adecuadamente controlados y que sean de confianza.

Anexo J

Lista alfabética de las definiciones de elementos de información

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

Este anexo proporciona un índice de las definiciones de formatos de certificado y de CRL, extensiones de certificado, clases de objeto, formas de nombre, tipos de atributo y reglas de concordancia definidas en esta Especificación de directorio.

Elemento	Cláusula
Formatos de certificado y de CRL	
Formato de certificado de atributo	12.1
Lista de revocación de certificados	7.3
Formato de certificado de clave pública	7
Extensiones de certificado, de CRL y de asiento de CRL	
Extensión de política de certificados aceptable	15.5.2.3
Extensión de política de privilegios aceptable	15.1.2.4
Extensión de descriptor de atributos	15.3.2.2
Extensión de identificador de atributos de autoridad	15.5.2.4
Extensión de identificador de claves de autoridad	8.2.2.1
Extensión de actualización básica	8.6.2.5
Extensión de constricciones de atributo básico	15.5.2.1
Extensión de constricciones básicas	8.4.2.1
Extensión de expedidor de certificado	8.6.2.3
Extensión de políticas de certificado	8.2.2.6
Extensión de puntos de distribución de CRL	8.6.2.1
Extensión de número de CRL	8.5.2.1
Extensión de ámbito de CRL	8.5.2.5
Extensión de identificador de flujo de CRL	8.5.2.7
Extensión de constricciones de nombre delegado	15.5.2.2
Extensión de indicador de CRL delta	8.6.2.4
Extensión de información delta	8.5.2.9
Extensión de certificados expirados en la CRL	8.5.2.12
Extensión de utilización de clave extendida	8.2.2.4
Extensión de CRL más reciente	8.6.2.6
Extensión de código de instrucción, retención	8.5.2.3
Extensión de expedidor indirecto	15.1.2.5
Extensión inhibición de cualquier política	8.4.2.4
Extensión de fecha de no validez	8.5.2.4
Extensión expedido en nombre de	15.5.2.6
Extensión de nombre alternativo de expedidor	8.3.2.2
Extensión de punto de distribución expedidor	8.6.2.2
Extensión de utilización de clave	8.2.2.3
Extensión de constricciones de nombre	8.4.2.2
Extensión sin aserción	15.1.2.6
Extensión de información de no revocación	15.2.2.2
Extensión de lista ordenada	8.5.2.8
Extensión de constricciones de políticas	8.4.2.3

Elemento	Cláusula
Extensión de correspondencias de políticas	8.2.2.7
Extensión de periodo de utilización de clave privada	8.2.2.5
Extensión de código de motivo	8.5.2.2
Extensión de grupo de certificados revocados	8.5.2.11
Extensión de identificador de certificado de especificación de cometido	15.4.2.1
Extensión de identificador SOA	15.3.2.1
Extensión de referencia de estado	8.5.2.6
Extensión de nombre alternativo de sujeto	8.3.2.1
Extensión de identificador de clave de sujeto	8.2.2.2
Extensión de atributos de directorio de sujeto	8.3.2.3
Extensión de información de objetivos	15.1.2.2
Extensión de especificación de tiempos	15.1.2.1
Extensión por revocar	8.5.2.10
Extensión de notificación de usuario	15.1.2.3
Clases de objeto y formas de nombre	
Clases de objeto de punto de distribución de CRL de certificado de atributo	17.1.4
Clase de objeto de política de certificado y de CPS	11.1.5
Clase de objeto y forma de nombre de punto de distribución de CRL	11.1.3
Clase de objeto de CRL delta	11.1.4
Clase de objeto de CA de PKI	11.1.2
Clase de objeto de trayecto de certificados de PKI	11.1.6
Clase de objeto de usuario de PKI	11.1.1
Clase de objeto de AA de PMI	17.1.2
Trayecto de delegación de PMI	17.1.5
Clase de objeto de SOA de PMI	17.1.3
Clase de objeto de usuario de PMI	17.1.1
Clase de objeto política de privilegios	17.1.6
Clase de objetos política de privilegios protegidos	17.1.7
Atributos de directorio	
Atributo de certificado de AA	17.2.2
Atributo de lista de revocación de certificados de AA	17.2.5
Atributo de certificado de atributo	17.2.1
Atributo de lista de revocación de certificados de atributo	17.2.4
Atributo de certificado de descriptor de atributo	17.2.3
Atributo de lista de revocación de autoridades	11.2.5
Atributo de certificado de CA	11.2.2
Atributo de declaración de práctica de certificación	11.2.8
Atributo de política de certificado	11.2.9
Atributo de lista de revocación de certificados	11.2.4
Atributo de pares de certificados cruzados	11.2.3
Atributo de trayecto de delegación	17.2.6
Atributo de lista de revocación delta	11.2.6
Atributo de trayecto de PKI	11.2.10
Atributo de política de privilegios	17.2.7
Atributo de política de privilegios protegidos	17.2.8
Atributo de algoritmos soportados	11.2.7
Atributo de certificado de usuario	11.2.1
Atributo de información de privilegios XML	14.5
Atributo de política de privilegios protegidos XML	17.2.9

Elemento	Cláusula
Reglas de concordancia	
Concordancia de identificadores de AA	15.5.2.4.1
Concordancia de políticas de certificado aceptables	15.5.2.3.1
Concordancia de identificadores de algoritmos	11.3.7
Concordancia exacta de certificados de atributos	17.3.1
Concordancia de certificados de atributos	17.3.2
Concordancia de descriptores de atributos	15.3.2.2.1
Concordancia de constricciones de atributos básicos	15.5.2.1.1
Concordancia exacta de certificados	11.3.1
Concordancia exacta de listas de certificados	11.3.5
Concordancia de listas de certificados	11.3.6
Concordancia de certificados	11.3.2
Concordancia exacta de pares de certificados	11.3.3
Concordancia de pares de certificados	11.3.4
Concordancia de constricciones de nombre delegado	15.5.2.2.1
Concordancia de trayectos de delegación	17.3.4
Concordancia de certificados mejorada	11.3.10
Concordancia de expedidores titulares	17.3.3
Extensión de expedidor indirecto	15.1.2.5
Concordancia de trayectos PKI	11.3.9
Concordancia de políticas	11.3.8
Concordancia de ID de certificados de especificación de cometido	15.4.2.1.1
Concordancia de identificador de SOA	15.3.2.1.1
Concordancia de especificaciones de tiempos	15.1.2.1.1

Anexo K

Enmiendas y corrigenda

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

La edición de esta Especificación de directorio incluye los siguientes proyectos de enmienda que fueron votados y aprobados por ISO/CEI:

– Enmienda 4 relativa a extensiones de certificados de claves públicas y de atributos.

La presente versión de esta Especificación de directorio incluye los siguientes corrigenda técnicos para rectificar los defectos que figuran en los siguientes informes de defectos con referencia a la 4ª versión de esta especificación:

– Corrigendum técnico 1 (que abarca los informes de defectos 272, 273, 274, 275, 276, 277, 278 y 279);

– Corrigendum técnico 2 (que abarca los informes de defectos 284, 285 y 286); y

– Corrigendum técnico 3 (que abarca los informes de defectos 281, 282, 289, 291, 296, 298, 299, 300, 301, 304 y 305).

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	Gestión de las telecomunicaciones, incluida la RGT y el mantenimiento de redes
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, comunicaciones de sistemas abiertos y seguridad**
Serie Y	Infraestructura mundial de la información, aspectos del protocolo Internet y Redes de la próxima generación
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación

Impreso en Suiza
Ginebra, 2006