Reemplazada por una versión más reciente

UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T X.509

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

(11/93)

REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS

DIRECTORIO

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

Recomendación UIT-T X.509
Reemplazada por una versión más reciente

(Anteriormente «Recomendación del CCITT»)

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. En el UIT-T, que es la entidad que establece normas mundiales (Recomendaciones) sobre las telecomunicaciones, participan unos 179 países miembros, 84 empresas de explotación de telecomunicaciones, 145 organizaciones científicas e industriales y 38 organizaciones internacionales.

Las Recomendaciones las aprueban los Miembros del UIT-T de acuerdo con el procedimiento establecido en la Resolución N.° 1 de la CMNT (Helsinki, 1993). Adicionalmente, la Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, aprueba las Recomendaciones que para ello se le sometan y establece el programa de estudios para el periodo siguiente.

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. El texto de la Recomendación UIT-T X.509 se aprobó el 16 de noviembre de 1993. Su texto se publica también, en forma idéntica, como Norma Internacional ISO/CEI 9594-8.

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.

RECOMENDACIONES DE LA SERIE UIT-T X

REDES DE DATOS
Y COMUNICACIÓN DE SISTEMAS ABIERTOS

(Febrero 1994)

ORGANIZACIÓN DE LAS RECOMENDACIONES DE LA SERIE X

Dominio	Recomendaciones
REDES PÚBLICAS DE COMUNICACIÓN 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 con conexión	X.220-X.229
Especificación de los protocolos en modo sin conexión	X.230-X.239
Formularios PICS	X.240-X.259
Identificación de protocolos	X.260-X.269
Protocolos de seguridad	X.270-X.279
Objetos gestionados de red	X.280-X.289
Pruebas de conformidad	X.290-X.299
INTERFUNCIONAMIENTO ENTRE REDES	
Consideraciones generales	X.300-X.349
Sistemas móviles de transmisión de datos	X.350-X.369
Gestión	X.370-X.399
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400-X.499
DIRECTORIO	X.500-X.599
GESTIÓN DE REDES OSI Y ASPECTOS DE SISTEMAS	
Gestión de redes	X.600-X.649
Denominación, direccionamiento y registro	X.650-X.679
Notación de sintaxis abstracta N.° 1 (ASN.1)	X.680-X.699
GESTIÓN OSI	X.700-X.799
SEGURIDAD	X.800-X.849
APLICACIONES OSI	
Cometimiento, concurrencia y recuperación	X.850-X.859
Procesamiento de transacción	X.860-X.879
Operaciones a distancia	X.880-X.899
TRATAMIENTO ABIERTO DISTRIBUIDO	X.900-X.999

Reemplazada por una versión más reciente

ÍNDICE

Sumario

Esta Recomendación | Norma Internacional define un marco para el suministro de servicios de autenticación por el directorio a sus usuarios. Describe dos niveles de autenticación: autenticación simple, mediante el uso de una contraseña como verificación de una identidad pretendida, y autenticación fuerte, que implica credenciales formadas usando técnicas criptográficas. Si bien la autenticación simple ofrece cierta protección limitada contra el acceso no autorizado, sólo la autenticación fuerte debe servir de base para ofrecer servicios seguros.

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), se utiliza típicamente para facilitar la comunicación entre, con o sobre objetos tales como entidades de aplicación de OSI, personas, terminales y listas de distribución.

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

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

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

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

Esta segunda edición, revisa y mejora técnicamente, sin sustituirla, a la primera edición de esta Recomendación | Norma Internacional. Las implementaciones pueden seguir alejando la conformidad con la primera edición.

Esta segunda edición especifica la vesión 1 de los protocolos y del serivico de directorio. La primera edición especifica también la versión 1. Las diferencias entre los servicios y entre los protocolos definidas en las dos ediciones quedan abarcadas mediante la utilización de las reglas de extensibilidad definidas en la presente edición de la Rec. X. 519 | ISO/CEI 9594-5

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

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

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

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

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

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

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

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

El Anexo J, que no es parte integrante de esta Recomendación | Norma Internacional, enumera las enmiendas e informes de defectos que han sido incorporados para formar esta edición de la presente Recomendación | Norma Internacional.

NORMA INTERNACIONAL

RECOMENDACIÓN UIT-T

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

SECCIÓN 1 – GENERALIDADES

1 Alcance

Esta Recomendación | Norma Internacional:

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

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

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

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

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

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

En general, el marco de autenticación no depende del uso de un determinado algoritmo criptográfico, siempre que tenga las propiedades descritas en 7.1. Es probable, en la práctica, que se use cierto número de algoritmos diferentes. Sin embargo, dos usuarios que quieran autenticar tienen que soportar el mismo algoritmo criptográfico para que la autenticación se realice correctamente. Así, dentro del contexto de un conjunto de aplicaciones conexas, la elección de un algoritmo único servirá para maximizar la comunidad de usuarios capaces de autenticar y comunicar de manera segura. En el Anexo D se presenta un ejemplo de un algoritmo criptográfico de claves públicas.

Análogamente, dos usuarios que deseen autenticar tienen que soportar la misma función hash [véase 3.3 f)] [usada en la formación de credenciales y testigos (tokens) de autenticación)]. Aquí también, en principio, un número de funciones hash alternativas pudieran ser usadas, a expensas de reducir las comunidades de usuarios capaces de autenticar. En el Anexo E se presenta una breve introducción a las funciones hash, así como un ejemplo de función hash.

2 Referencias normativas

Las siguientes Recomendaciones y las Normas Internacionales siguientes contienen disposiciones, que mediante su 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 las Normas son objeto de revisiones, con 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 las 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.500 (1993) | ISO/CEI 9594-1:1994, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Visión de conjunto de conceptos, modelos y servicios*.
- Recomendación UIT-T X.501 (1993) | ISO/CEI 9594-2:1994, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Modelos*.
- Recomendación UIT-T X.511 (1993) | ISO/CEI 9594-3:1994, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Definición del servicio abstracto*.
- Recomendación UIT-T X.518 (1993) | ISO/CEI 9594-4:1994, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Procedimientos para operación distribuida*.
- Recomendación UIT-T X.519 (1993) | ISO/CEI 9594-5:1994, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Especificaciones de protocolos*.
- Recomendación UIT-T X.520 (1993) | ISO/CEI 9594-6:1994, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Tipos de atributos seleccionados*.
- Recomendación UIT-T X.521 (1993) | ISO/CEI 9594-7:1994, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Clases de objetos seleccionadas*.
- Recomendación UIT-T X.525 (1993) | ISO/CEI 9594-9:1994, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Replicación*.
- Recomendación UIT-T X.680 (1994) | ISO/CEI 8824-1:1994, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*
- Recomendación UIT-T X.681 (1994) | ISO/CEI 8824-2:1994, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de objetos de información.*
- Recomendación UIT-T X.682 (1994) | ISO/CEI 8824-3:1994, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de constricciones.*
- Recomendación UIT-T X.683 (1994) | ISO/CEI 8824-4:1994, *Tecnología de la información – Notación de sintaxis abstracta uno: Parametrización de especificaciones ASN.1.*
- Recomendación UIT-T X.690 (1994) | ISO/CEI 8825-1:1994, *Especificación de las reglas de codificación básica, de las reglas de especificación canónica y de las reglas de codificación distinguida.*
- Recomendación UIT-T X.880 (1994) | ISO/CEI 13712-1:1994, *Tecnología de la información – Operaciones a distancia: Conceptos, modelo y notación.*
- Recomendación UIT-T X.881 (1994) | ISO/CEI 13712-2:1994, *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 UIT-T X.800 del CCITT (1991), *Arquitectura de seguridad de interconexión de sistemas abiertos para aplicaciones del CCITT.*

  ISO/CEI 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. X.800 del CCITT | ISO 7498-2:

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

3.2 Definiciones relativas al modelo de directorio

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

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

3.3 Definiciones relativas al marco de autenticación

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

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

3.3.2 certificado de usuario; certificado: Claves públicas de un usuario, junto con alguna otra información, hechas infalsificables por cifrado con la clave secreta de la autoridad de certificación que la emitió.

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

3.3.4 trayecto de certificación: Secuencia ordenada de certificados de objetos en el DIT la cual, junto con la clave pública del objeto inicial en el trayecto, puede ser procesada para obtener la del objeto final en el trayecto.

3.3.5 sistema criptográfico, criptosistema: Colección de transformaciones de texto ordinario en texto cifrado y viceversa, seleccionándose por claves la transformación o transformaciones a ser usadas. Las transformaciones se definen normalmente por un algoritmo matemático.

3.3.6 función hash: Función (matemática) que hace corresponder valores de un dominio vasto (que puede ser muy vasto) con una gama menor. Una función hash 'buena' es aquélla que cuando se aplica a un conjunto (grande) de valores en el dominio, los resultados se distribuyen uniformemente (y aparentemente al azar) en toda la gama.

3.3.7 función unidireccional: Función (matemática) «f» fácil de computar, pero que, para un valor general «y» en la gama, es computacionalmente difícil hallar, en el dominio, un valor x tal que f(x)=y. Puede haber unos pocos valores «y» para los cuales hallar x no sea computacionalmente difícil.

3.3.8 clave pública: (En un criptosistema de claves públicas) la clave, de un par de claves de un usuario, que se conoce públicamente.

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

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

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

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

3.3.13 fiduciario: Generalmente, se puede decir que una entidad acepta como «fiduciaria» a una segunda entidad cuando aquélla (la primera entidad) confía en que la segunda entidad se comportará exactamente como ella lo espera. Esta relación fiduciaria puede que sea aplicable solamente para alguna función específica. El papel principal de la confianza en el marco de la autenticación es el de describir la relación entre la entidad autenticadora y una autoridad de certificación; una entidad autenticadora tiene que estar segura de que puede confiar en que la autoridad de certificación creará solamente certificados válidos y fiables.

3.3.14 número secuencial de certificado: Valor entero, único en la CA expedidora, que va asociado inequívocamente a un certificado expedido por dicha CA.

4 Abreviaturas

A los efectos de esta Recomendación | Norma Internacional se utilizan las siguientes abreviaturas:

CA	Autoridad de certificación (*certification authority*)
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*)
PKCS	Criptosistema de claves públicas (*public key cryptosystem*)

5 Convenios

Con pequeñas excepciones, esta especificación de directorio se ha preparado con arreglo a las directrices de «Presentación de textos comunes UIT-T/ISO/CEI» que figuran en la Guía para la cooperación entre el UIT-T y el JTC1 ISO/CEI, de marzo de 1993.

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

Esta especificación de directorio utiliza el término «sistemas edición 1988» para hacer referencia a los sistemas conformes a la anterior edición (1988) de las especificaciones de directorio. Los sistemas conformes a las actuales especificaciones de directorio se designan por «sistemas edición 1993», es decir, la edición de 1988 de las Recomendaciones de la serie X.500 del CCITT.

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

La notación usada en esta especificación de directorio se define en el Cuadro 1.

Cuadro 1 – Notación

Notación	Significado
Xp	Clave pública de un usuario X.
Xs	Clave secreta de X.
Xp[I]	Cifrado de alguna información, I, mediante la clave pública de X.
Xs[I]	Cifrado de I mediante la clave secreta 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 $X_1<<X_{n+1}>>$. Por ejemplo, la posesión de A<<B>>B<<C>> confiere la misma capacidad que A<<C>>, a saber, la aptitud para hallar Cp se da Ap.
$X_1p \bullet X_1<<X_2>>$	La operación de desenvolver (<<unwrap>>) 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: $Ap \bullet A<<B>> \quad B<<C>>$ 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, X_1, X_2, etc., aparecen en lugar de los nombres de los usuarios, mientras que el símbolo I aparece en lugar de una información arbitraria.	

SECCIÓN 2 – AUTENTICACIÓN SIMPLE

6 Procedimiento de autenticación simple

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

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.

NOTAS

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

2 Los procedimientos de señalización para proteger las contraseñas pueden ser una cuestión de interés para la ampliación de la Recomendación.

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

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

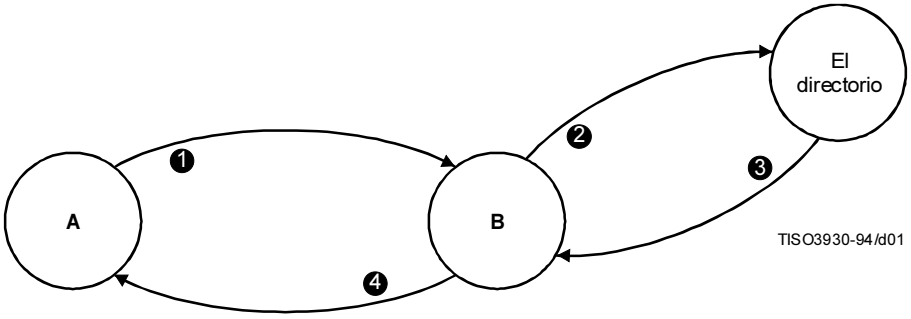


Figura 1 – Procedimiento de autenticación simple no protegida

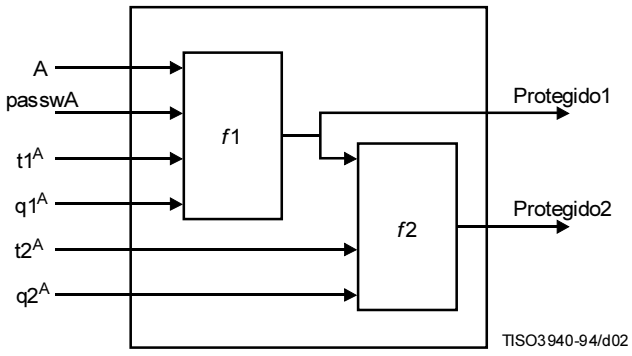
Comprende los siguientes pasos:

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.

6.1 Generación de información de identificación protegida

La Figura 2 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
Ctña^A	Contraseña de A *(password of A)*
q^A	Números aleatorios, y opcionalmente con un contador incluido

Figura 2 – Autenticación simple protegida

6.2 Procedimiento de autenticación simple protegida

La Figura 3 ilustra el procedimiento de autenticación simple protegida.

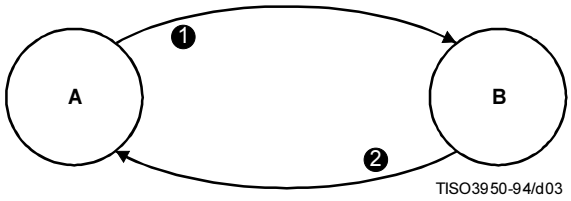


Figura 3 – El procedimiento de autenticación simple protegida

Comprende los siguientes pasos (inicialmente sólo se utiliza $f1$):

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 2, donde la indicación de tiempo y/o el número aleatorio (si se utiliza) tienen por finalidad minimizar la reproducción y ocultar la contraseña.

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

   Protegido1 = $f1$ ($t1^A$; $q1^A$; A; passw A)

   La información transportada a B tiene la forma siguiente:

   Autenticador1 = $t1^A$; $q1^A$; A; protegido1

   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.

2) 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 su información de identificación protegida (adicionalmente) (autenticación2) a B. Una protección adicional se obtiene aplicando una segunda función unidireccional, $f2$, como se ilustra en la Figura 2. 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 (esto es similar en principio al paso 1 de 6.4.1);

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 las Recomendaciones UIT-T X.511 | ISO/CEI 9594-3 y Rec. 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.

6.3 Tipo de atributo contraseña de usuario

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

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

SECCIÓN 3 – AUTENTICACIÓN FUERTE

7 Base de la autenticación fuerte

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

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

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

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

Cada usuario queda identificado por el hecho de estar en posesión de la clave secreta. Un segundo usuario puede determinar si su 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 cláusula 8 describe un proceso por el cual una clave pública de usuario puede ser verificada por referencia al directorio. Este proceso sólo puede operar si hay una cadena ininterrumpida de puntos de confianza, en el directorio, entre los usuarios que solicitan autenticación. Esta cadena puede construirse identificando un punto común de confianza. Este punto común de confianza deberá estar enlazado con cada usuario por una cadena ininterrumpida de puntos de confianza.

8 Obtención de una clave pública de usuario

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

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

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

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

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

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

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

```
Certificate ::= SIGNED { SEQUENCE {
    version                    [0]  Version DEFAULT v1,
    serialNumber                    CertificateSerialNumber,
    signature                            AlgorithmIdentifier,
    issuer                     Name,
    validity                        Validity,
    subject                         Name,
    subjectPublicKeyInfo            SubjectPublicKeyInfo,
    issuerUniqueIdentifier     [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                                    -- si está presente, la versión tiene que ser v2
    subjectUniqueIdentifier    [2]  IMPLICIT UniqueIdentifier OPTIONAL
                                    -- si está presente, la versión tiene que ser v2 -- }}

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

CertificateSerialNumber ::= INTEGER

AlgorithmIdentifier ::= SEQUENCE {
    algorithm  ALGORITHM.&id ({SupportedAlgorithms}),
    parameters ALGORITHM.&Type ({SupportedAlgorithms}{ @algorithm}) OPTIONAL }
-- Se demora la definición del objeto de información siguiente, quizás hasta que se disponga de perfiles
-- normalizados o de enunciados de conformidad de implementación de protocolo. El conjunto necesitará
-- especificar una constricción tabular impuesta al componente parameters de AlgorithmIdentifier.
-- SupportedAlgorithms  ALGORITHM ::= { ... | ...}

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

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

NOTA 2 – En los casos en que la autoridad de denominación reasigne un nombre distinguido a un usuario diferente, los CA pueden utilizar el identificador único para distinguir entre los casos de reutilización. Sin embargo, si el mismo usuario recibe certificados de múltiples CA, se recomienda que, como parte de su procedimiento de registración de usuarios, los CA coordinen la asignación de identificadores únicos.

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

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

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

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

Un trayecto de certificación forma lógicamente una cadena ininterrumpida de puntos de confianza («trusted points») 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 AB y BA puede variar. Una manera de facilitar esto consiste en organizar una jerarquía de CA, que puede o no coincidir con la totalidad o una parte de la jerarquía del DIT. La ventaja de esto es que los usuarios que tienen CA en la jerarquía pueden establecer entre sí un trayecto de certificación utilizando el directorio sin ninguna información previa; para que esto sea posible, cada CA puede almacenar un certificado (directo) y un certificado inverso designado como correspondiente a su CA superior.

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

```
userCertificate                 ATTRIBUTE   ::=         {
        WITH SYNTAX             Certificate
        ID                      id-at-userCertificate }

cACertificate                   ATTRIBUTE   ::=         {
        WITH SYNTAX             Certificate
        ID                      id-at-cAcertificate }

crossCertificatePair            ATTRIBUTE   ::=         {
        WITH SYNTAX             CertificatePair
        ID                      id-at-crossCertificatePair }

CertificatePair                 ::=     SEQUENCE {
        forward         [0]     Certificate OPTIONAL,
        reverse         [1]     Certificate OPTIONAL
                                -- por lo menos uno del par debe estar presente -- }
```

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

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

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

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

```
ForwardCertificationPath  ::=   SEQUENCE OF CrossCertificates

CrossCertificates               ::=     SET OF Certificate
```

8.1 Optimización del volumen de información obtenido del directorio

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

a) si los usuarios que quieren autenticar son servidos por la misma autoridad de certificación, el trayecto de certificación resulta trivial y los usuarios desenvuelven («unwrap») 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.

8.2 Ejemplo

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

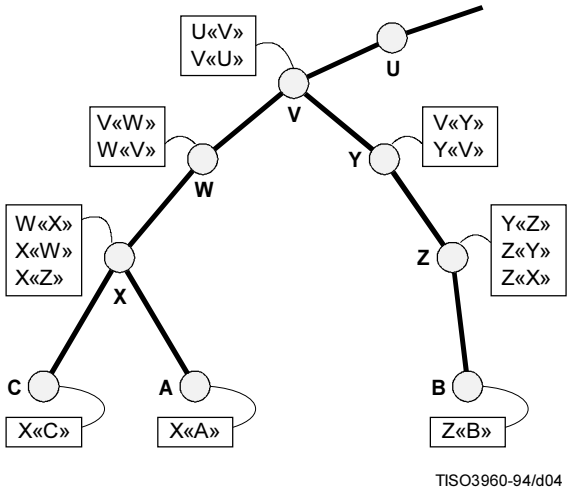


Figura 4 – Jerarquía de CA – Ejemplo teórico

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

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

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

$$Bp = Xp \bullet X\langle\langle W\rangle\rangle\ W\langle\langle V\rangle\rangle\ V\langle\langle Y\rangle\rangle\ Y\langle\langle Z\rangle\rangle\ Z\langle\langle B\rangle\rangle$$

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

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

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

$$Ap = Zp \bullet Z\langle\langle Y\rangle\rangle\ Y\langle\langle V\rangle\rangle\ V\langle\langle W\rangle\rangle\ W\langle\langle X\rangle\rangle\ X\langle\langle A\rangle\rangle$$

Aplicando las optimizaciones de 8.1:

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

$$Cp = Xp \bullet X\langle\langle C\rangle\rangle$$

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

$$Ap = Xp \bullet X\langle\langle A\rangle\rangle$$

b) suponiendo que A conociera así W<<X>>, Wp, V<<W>>, Vp, U<<V>>, hacia arriba, etc., la información que A tiene que obtener del directorio para formar el trayecto de autenticación se reduce a:

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

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

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

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 del directorio para X, y viceversa (esto se muestra en la Figura 4). Si A quiere autenticar hacia B, sólo necesita obtener:

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

para formar el trayecto de certificación, y:

$$Z\langle\langle X\rangle\rangle$$

para formar el trayecto de certificación de retorno;

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

$$Cp = Xp \bullet X\langle\langle C\rangle\rangle$$

y

$$Ap = Xp \bullet X\langle\langle A\rangle\rangle$$

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

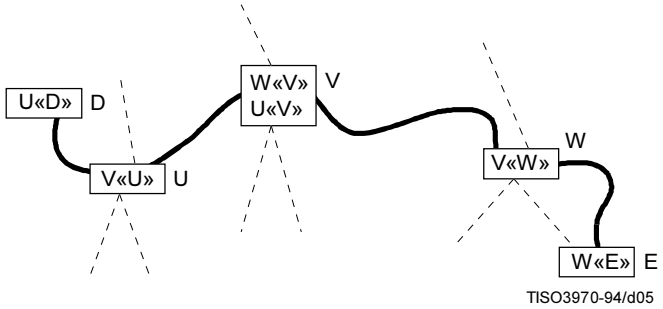


Figura 5 – Ejemplo de trayecto de certificación no jerárquico

Sea V una CA con la cual las CA, U y W han efectuado anteriormente cierto intercambio de redes públicas en una situación de confianza. Como resultado de esto se han generado y almacenado en el directorio certificados U<<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 debe encontrar un trayecto de certificación E. Este usuario podría utilizar diversos métodos. Uno de ellos consistiría en considerar los usuarios y CA como nodos, y los certificados como arcos en un gráfico dirigido. En estos términos, D debe efectuar una búsqueda en el gráfico para encontrar un trayecto de U a E, siendo uno de ellos U<<V>>, V<<W>>, W<<E>>. Una vez descubierto este trayecto, se puede construir también el trayecto inverso W<<V>>, V«U», U<<D>>.

9 Firmas digitales

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

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

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

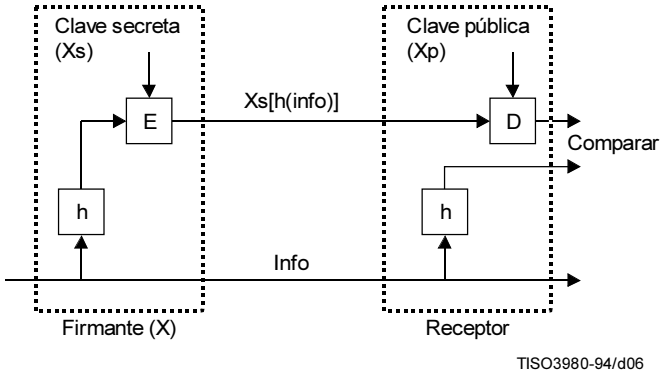


Figura 6 – Firmas digitales

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

El receptor de información firmada verifica la firma:

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

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

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

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

```
ENCRYPTED { ToBeEnciphered }  ::=      BIT STRING ( CONSTRAINED BY {
        -- debe ser el resultado de aplicar un procedimiento de cifrado a los octetos--
        -- con codificación BER de un valor de  -- 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. X.690 UIT-T | ISO/CEI 8825-1) del valor del tipo **ToBeEnciphered** y aplicando un procedimiento de cifrado a esos octetos.

NOTA 2 – El procedimiento de encripción 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. Es en los procedimientos de encripción donde se especificarán los medios para obtener la sintonización de los datos de emisor y del receptor, lo que puede incluir información en los bits que deban transmitirse.

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

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

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

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

Cuando sólo se requiera la firma, puede utilizarse la siguiente macro ASN.1 para definir el tipo de datos resultante de la aplicación de una firma al tipo de datos dado.

```
SIGNATURE { OfSignature }      ::=    SEQUENCE {
    algorithmIdentifier                AlgorithmIdentifier,
    encrypted                          ENCRYPTED { HASHED { OfSignature }}}
```

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

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 («Tag value»);

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 («scaling factor») binario será cero.

10 Procedimientos de autenticación fuerte

10.1 Visión de conjunto

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

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

     Las últimas propiedades pueden ser determinadas también para todo otro dato arbitrario adicional en la transferencia;

b) la autenticación bidireccional, descrita en 10.3, implica, además, una respuesta de B a A. Determina además lo siguiente:
   - que el testigo de autenticación generado en la respuesta fue generado realmente por B y estaba previsto para ser enviado a A;
   - la integridad y originalidad del testigo de autenticación enviado en la respuesta;
   - (opcionalmente), el secreto mutuo de una parte de los testigos;

c) la autenticación tridireccional, descrita en 10.4, implica, además, una transferencia ulterior de A a B. Determina las mismas propiedades que la autenticación bidireccional, pero lo hace sin necesidad de comprobación de la indicación de la hora de la asociación.

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

La comprobación de las indicaciones de hora mencionadas en las siguientes 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.

10.2 Autenticación unidireccional

Se siguen los pasos indicados en la Figura 7:

1) A genera r^A, un número no repetitivo, que se usa para detectar ataques de reactuación y para prevenir la falsificación.

2) A envía el siguiente mensaje a B:

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

donde t^A es una indicación de tiempo. t^A consta de una o dos fechas: la hora de generación del testigo (que es opcional) y la fecha de expiración. Como otra posibilidad, si se debe proporcionar autenticación del origen de datos de 'sgnData' por firma digital:

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

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

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

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 $B \rightarrow A$, 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 hora t^A en lenguaje ordinario junto con la parte del testigo a que se aplicó la función hash, durante el rango de tiempo t^A.

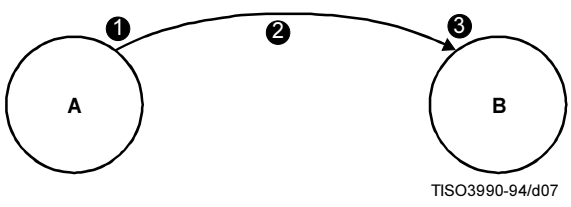


Figura 7 – Autenticación unidireccional

10.3 Autenticación bidireccional

Se siguen los pasos indicados en la Figura 8:

1) Como en 10.2.
2) Como en 10.2.
3) Como en 10.2.
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 hora t^B es «corriente»;
   d) opcionalmente, comprueba que r^B no ha sido reactuado [véase 10.2, apartado 3 d)].

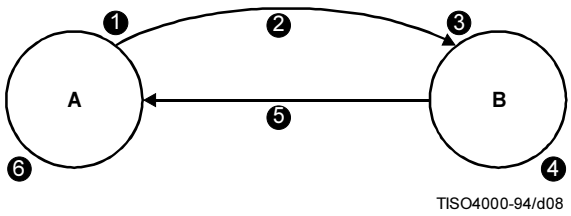


Figura 8 – Autenticación bidireccional

10.4 Autenticación tridireccional

Se siguen los pasos indicados en la Figura 9:

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

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

9) B efectúa las siguientes acciones:
   a) descifra el testigo de autenticación, después comprueba la firma y por consiguiente la integridad de la información firmada;
   b) comprueba que el r^B recibido es idéntico al r^B que fue enviado por B.

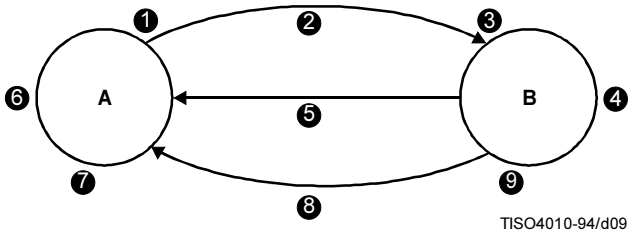


Figura 9 – Autenticación tridireccional

11 Gestión de claves y certificados

11.1 Generación de pares de claves

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

Los datos de la clave no son fáciles de recordar por un usuario humano, por lo que hay que emplear un método apropiado para almacenarla en un modo transportable conveniente. Un mecanismo posible sería el uso de una «tarjeta inteligente» («smart card») que podría contener las claves secreta y (opcionalmente) pública del usuario, el certificado del usuario, y una copia de la clave pública de la autoridad de certificación. El uso de esta tarjeta podría también asegurarse por medio de un PIN (número de identificación personal), 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 secreta del usuario nunca es pasada a otra entidad, pero requiere un cierto nivel de competencia por el usuario, como se describe en el Anexo D.

b) El par de claves es generado por una tercera entidad. La tercera entidad tiene que pasar la clave secreta 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. Este es un caso especial de b) y las consideraciones hechas allí son aplicables.

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

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

11.2 Gestión de certificados

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

a) una autoridad de certificación tiene 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.

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

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

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 fraudelentas.

b) Si se emplea el medio de generación de los pares de claves de 11.1 c), no se necesita una transferencia securizada.

c) Si se emplea el medio de generación de pares claves descrito en 11.1 a) ó 11.1 b), el usuario puede utilizar diferentes métodos (en línea o fuera de línea) para comunicar su clave pública a la CA de una manera segura. Los métodos en-línea pueden proporcionar una mayor flexibilidad para las operaciones a distancia efectuadas entre el usuario y la CA.

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

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

– La validez de los certificados deberá organizarse de tal modo que la validez de uno entrañe la caducidad del precedente, o se puede permitir que sus periodos de validez se superpongan. Esto último evita que las CA tengan que instalar y distribuir un gran número de certificados que pudieran agotarse en la misma fecha de caducidad (expiración).

– Los certificados caducados normalmente serán sacados del directorio. Es cuestión de política de seguridad y de responsabilidad de la CA mantener los certificados antiguos durante cierto periodo de tiempo, si se presta el servicio de «no repudio de los datos».

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

– La revocación de un certificado de usuario o de un certificado de CA debe ponerse en conocimiento de la CA, y deberá expedirse un nuevo certificado si fuese procedente. La CA podrá entonces informar al propietario del certificado, sobre su revocación, por un procedimiento fuera de línea.

– La CA mantendrá:

  a) una lista, con indicación de tiempo, de los certificados expedidos que han sido revocados;

  b) una lista, con indicación de tiempo, de los certificados revocados de todas las CA, conocidos por la CA, expedidos por la CA.

  Ambas listas deberán existir, aunque pueden estar vacías.

– El mantenimiento de inserciones del directorio afectadas por las listas de revocaciones, por la CA, es responsabilidad del directorio y sus usuarios, quienes actúan de acuerdo con la política de seguridad. Por ejemplo, el usuario puede modificar su inserción de objeto reemplazando el antiguo certificado por uno nuevo. Este último se utilizará entonces para autenticar el usuario ante el directorio.

– Las listas de revocaciones («listas negras») se mantienen dentro de inserciones como atributos de tipos «CertificateRevocationList» y «AuthorityRevocationList». Esos atributos pueden ser operados utilizando los mismos procedimientos empleados para otros atributos. Estos tipos de atributo se definen como sigue:

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

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

CertificateList             ::=   SIGNED { SEQUENCE {
    signature                         AlgorithmIdentifier,
    issuer                            Name,
    thisUpdate                        UTCTime,
    nextUpdate                        UTCTime OPTIONAL,
    revokedCertificates               SEQUENCE OF SEQUENCE {
        userCertificate               CertificateSerialNumber,
        revocationDate                UTCTime } OPTIONAL }}
```

NOTAS

1 La verificación de la totalidad de los certificados es un asunto local.

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

Anexo A

Marco de autenticación en ASN.1

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

Este anexo incluye todas las definiciones de tipo, macro y valor ASN.1, contenidas en esta Recomendación, en la forma del módulo ASN.1 **«AuthenticationFramework»**.

```
AuthenticationFramework {joint-iso-ccitt ds(5) module(1) authenticationFramework(7) 2}
DEFINITIONS ::=
BEGIN

-- EXPORTS All --
-- Los tipos y valores definidos en este módulo son exportados para su utilización en otros módulos ASN.1 contenidos
-- en las especificaciones de directorio, y para uso de otras aplicaciones que los utilizarán para acceder a servicios
-- de directorio. Otras aplicaciones pueden utilizarlos para sus propios fines, pero esto no constreñirá las
-- extensiones y modificaciones necesarias para mantener o mejorar el servicio de directorio.

IMPORTS
        id-at, informationFramework, upperBounds, selectedAttributeTypes, basicAccessControl
                FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 2}

        Name, ATTRIBUTE
                FROM InformationFramework informationFramework

        ub-user-password
                FROM UpperBounds upperBounds

        AuthenticationLevel
                FROM BasicAccessControl basicAccessControl

        UniqueIdentifier, octetStringMatch
                FROM SelectedAttributeTypes selectedAttributeTypes      ;

-- tipos --

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,
                                                        -- si está presente, la versión tiene que ser v2
        subjectUniqueIdentifier         [2]     IMPLICIT UniqueIdentifier OPTIONAL
                                                        -- si está presente, la versión tiene que ser v2  -- }}

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

CertificateSerialNumber     ::=     INTEGER

AlgorithmIdentifier         ::=     SEQUENCE {
        algorithm               ALGORITHM.&id ({SupportedAlgorithms}),
        parameters              ALGORITHM.&Type ({SupportedAlgorithms}{ @algorithm}) OPTIONAL }

-- Se demora la definición del objeto de información siguiente, quizás hasta que se disponga de perfiles
-- normalizados o de enunciados de conformidad de implementación de protocolo. El conjunto necesitará
-- especificar una constricción tabular impuesta al componente parameters de AlgorithmIdentifier.
```

```
SupportedAlgorithms     ALGORITHM ::=       { ... }

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

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

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

ForwardCertificationPath ::=    SEQUENCE OF CrossCertificates

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

CrossCertificates       ::=     SET OF Certificate

CertificateList         ::=     SIGNED { SEQUENCE {
        signature               AlgorithmIdentifier,
        issuer                  Name,
        thisUpdate              UTCTime,
        nextUpdate              UTCTime OPTIONAL,
        revokedCertificates     SEQUENCE OF SEQUENCE {
                userCertificate         CertificateSerialNumber,
                revocationDate          UTCTime } OPTIONAL}}

CertificatePair         ::=     SEQUENCE {
        forward         [0]     Certificate OPTIONAL,
        reverse         [1]     Certificate OPTIONAL
                                -- por lo menos uno del par debe estar presente -- }

-- tipos de atributos --

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

userCertificate         ATTRIBUTE   ::=     {
        WITH SYNTAX     Certificate
        ID              id-at-userCertificate }

cACertificate           ATTRIBUTE   ::=     {
        WITH SYNTAX     Certificate
        ID              id-at-cAcertificate }

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

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

crossCertificatePair ATTRIBUTE  ::=     {
        WITH SYNTAX     CertificatePair
        ID              id-at-crossCertificatePair }

-- clases de objeto de información --

ALGORITHM ::= TYPE-IDENTIFIER

-- tipos parametrizados --

HASHED { ToBeHashed }           ::=     OCTET STRING ( CONSTRAINED BY {
        -- debe ser el resultado de aplicar un procedimiento hashing a los octetos --
        -- con codificación DER (véase 8.7) --
        -- de un valor de -- ToBeHashed })
```

```
ENCRYPTED { ToBeEnciphered }  ::=     BIT STRING ( CONSTRAINED BY {
        -- debe ser el resultado de aplicar un procedimiento de cifrado a los octetos --
        -- con codificación BER de un valor -- ToBeEnciphered })

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

SIGNATURE { OfSignature }         ::=     SEQUENCE {
        algorithmIdentifier               AlgorithmIdentifier,
        encrypted                         ENCRYPTED { HASHED { OfSignature }}}

-- asignaciones de indentificadores de objeto --

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}

END
```

Anexo B

Requisitos de seguridad[1)]

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

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

B.1 Peligros

Algunos peligros comúnmente conocidos son:

a) *Intercepción de identidad* – La identidad de uno o más de los usuarios que participan en una comunicación se observa con el fin de detectar todo uso incorrecto.

b) *Usurpación de identidad, suplantación, impostura o mascarada (masquerade)* – Pretensión de un usuario de ser otro diferente para ganar acceso a información u obtener privilegios adicionales.

c) *Reactuación o reproducción (Replay)* – Registración y posterior reactuación de una comunicación en alguna fecha posterior.

d) *Intercepción de datos* – Observación de datos de un usuario durante una comunicación, por un usuario no autorizado.

e) *Manipulación* – Reemplazo, inserción, supresión u ordenación incorrecta de datos de usuario durante una comunicación, por un usuario no autorizado.

f) *Repudio* – Negación por un usuario de haber participado en una comunicación, o parte de ella.

g) *Denegación de servicio* – Prevención o interrupción de una comunicación, o demora de operaciones críticas en el tiempo.

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

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

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

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

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

B.2 Servicios de seguridad

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

a) *Autenticación de entidad par* – Este servicio proporciona una corroboración de que un usuario en una determinada instancia de comunicación es el que se anuncia como tal. Pueden solicitarse dos servicios diferentes de autenticación de identidad par:

- *autenticación de entidad simple* (ya sea autenticación de entidad de *origen de datos* o autenticación de entidad de *receptor de datos*);
- *autenticación mutua*, donde ambos usuarios comunicantes se autentican el uno al otro.

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

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

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

b) *Control de acceso* – Este servicio puede usarse para proteger contra el uso no autorizado de recursos. El servicio de control de acceso es proporcionado por el directorio u otra aplicación y no es por consiguiente un asunto del marco de autenticación.

c) *Confidencialidad de datos* – Este servicio puede usarse para suministrar protección de los datos contra una revelación no autorizada. El servicio de confidencialidad de datos está soportado por el marco de autenticación. El mismo puede usarse para proteger contra intercepción de datos.

d) *Integridad de datos* – Este servicio suministra prueba de la integridad de los datos en una comunicación. El servicio de integridad de datos está soportado por el marco de autenticación. Puede usarse para detectar y proteger contra la manipulación.

e) *No repudio* – Este servicio suministra la prueba de la integridad y del origen de los datos – ambos en una relación infalsificable – que pueden ser verificados por un tercero en cualquier momento.

B.3 Mecanismos de seguridad

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

a) *Intercambio de autenticación* – Hay dos grados de mecanismos de autenticación suministrados por el marco de autenticación:

Autenticación simple – Se basa en que el originador suministre su nombre y contraseña, los cuales son comprobados por el receptor;

Autenticación fuerte – Se basa en el uso de técnicas criptográficas para proteger el intercambio de información de validación. En el marco de autenticación, la autenticación fuerte se basa en un esquema asimétrico.

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

b) *Cifrado* – El marco de autenticación contempla el cifrado de datos durante la transferencia. Pueden usarse esquemas simétricos o asimétricos. El intercambio necesario de claves se realiza o bien dentro de un intercambio de autenticación precedente o fuera de línea en cualquier momento antes de la comunicación que se va a hacer. Este último caso está fuera del ámbito del marco de autenticación. El mecanismo de cifrado soporta el servicio de confidencialidad de datos.

c) *Integridad de los datos* – Este mecanismo implica el cifrado de una cadena comprimida de los datos pertinentes a transmitir. Junto con los datos ordinarios, este mensaje se le envía al receptor. El receptor repite la compresión y el cifrado ulterior de los datos ordinarios y compara el resultado con el creado por el originador para probar la integridad.

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

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

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

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

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

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

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

Cuadro B.1 – Peligros y protección

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

B.5 Negociación de servicios y mecanismos de seguridad

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

Anexo C

Introducción a la criptografía de claves públicas[2)]

(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 secreta complementaria (Xs), puede descifrar los datos. (Esto se representa por la notación D = Xs[Xp[D]]). Es computacionalmente irrealizable derivar la clave secreta 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 C.1.

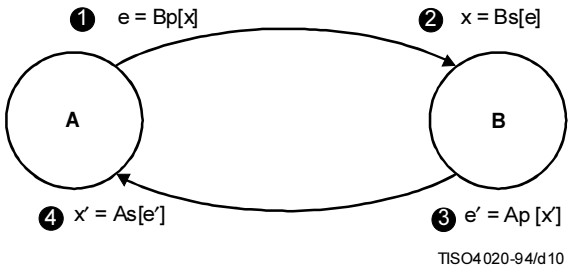


Figura C.1 – Uso de un PKCS para intercambiar información secreta

El usuario A tiene la clave pública Ap y la clave secreta 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 secreta del otro. A y B pueden por consiguiente intercambiar información secreta entre ellos siguiendo los pasos siguientes (ilustrados en la Figura C.1).

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

$$e = Bp[x]$$

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′:

$$x' - As[e'], \text{ o } x' = As[Ap[x']]$$

---

2) Para más información, véase:

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

Por este medio, A y B han intercambiado la información secreta x y x′. Esta información no puede ser obtenida por ningun otro que A y B, siempre que sus claves secretas 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 D

El criptosistema[3] de claves públicas RSA[4]

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

D.1 Alcance y campo de aplicación

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

D.2 Definiciones

Los siguientes términos se definen en este campo.

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

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

SEQUENCE {INTEGER, INTEGER}

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

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

D.3 Símbolos y abreviaturas

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

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

---

3) El criptosistema especificado en este anexo fue creado por R.L. Rivest, A. Shamir y L. Adleman, y se conoce generalmente por algoritmo RSA.

4) Para más información, véase:

Aspectos generales

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

Referencia de generación de claves

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

Referencia de descifrado

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

D.4 Descripción

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

$$Y = X^e \bmod n \quad \text{con} \quad 0 \le X < n$$

$$X = Y^d \bmod n \quad \text{con} \quad 0 \le Y < n$$

que puede ser satisfecha, por ejemplo, por

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

$$ed \bmod (p-1)(q-1) = 1$$

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

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

D.5 Requisitos de seguridad

D.5.1 Longitudes de las claves

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

D.5.2 Generación de claves

La seguridad de RSA se basa en la dificultad de factorizar n. Hay muchos algoritmos para realizar esta operación, y para obstaculizar el uso de cualquier técnica actualmente conocida, los valores p y q tienen que ser escogidos cuidadosamente, de acuerdo a las reglas siguientes (por ejemplo, véase la nota de pie de página 4, «Referencia de generación de claves»):

a) deben ser escogidos al azar;

b) deben ser grandes;

c) deben ser números primos;

d) $|p-q|$ debe ser grande;

e) $(p+1)$ tendrá un factor primo grande;

f) $(q+1)$ tendrá un factor primo grande;

g) $(p-1)$ tendrá un factor primo grande, por ejemplo, r;

h) $(q-1)$ tendrá un factor primo grande, por ejemplo, s;

i) $(r-1)$ tendrá un factor primo grande;

j) $(s-1)$ tendrá un factor primo grande.

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

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

D.6 Exponente público

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

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

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

$$= 65537 \text{ decimal, y}$$

$$= 1\ 0000\ 0000\ 0000\ 0001 \text{ binario.}$$

NOTAS

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

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

D.7 Conformidad

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

Anexo E

Funciones hash

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

La función hash cuadrado-mod descrita anteriormente en este anexo de la primera edición de la presente especificación de directorio está desaconsejada.

E.1 Requisitos de las funciones hash

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

Una función hash fuerte cumplirá los siguientes requisitos:

a) La función hash tiene que ser unidireccional, es decir, dado un resultado hash posible cualquiera, tiene que ser computacionalmente irrealizable construir un mensaje de entrada que, sometido a una función hash, dé este resultado.

b) La función hash tiene que estar libre de colisiones, es decir, tiene que ser computacionalmente irrealizable construir dos mensajes de entrada distintos que al ser sometidos a una función hash den el mismo resultado.

Anexo F

Peligros contra los que ofrece protección el método de autenticación fuerte

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

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

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

Comprometer la clave secreta del usuario – Uno de los principios básicos de autenticación fuerte es que la clave secreta del usuario permanezca segura. Un número de métodos prácticos están disponibles para que el usuario mantenga su clave secreta en una forma que ofrezca la seguridad adecuada. Las consecuencias de esta situación se limitan a un trastorno de la comunicación en que interviene ese usuario.

Comprometer la clave secreta de la CA – El hecho de que la clave secreta de una CA permanezca segura es también un principio básico de la autenticación fuerte. La seguridad física y los métodos «necesidad de conocer» se aplican. Las consecuencias de esta situación se limitan a un trastorno de la comunicación en que interviene cualquier usuario certificado por esa CA.

Inducir a error a la CA para que cree un certificado no válido – El hecho de que las CA funcionen fuera de línea da cierta protección. Recae sobre la CA el trabajo de comprobar que las credenciales fuertes contempladas son válidas, antes de crear un certificado. Las consecuencias de esta situación se limitan a un trastorno de la comunicación en que interviene el usuario para el cual se creó el certificado, y cualquiera afectado por el certificado no válido.

Colusión entre una CA deshonesta y un usuario – Un ataque de este tipo hará fracasar este método. Esto podría constituir una traición a la confianza depositada en la CA. Las consecuencias de una CA deshonesta se limitan a un trastorno de la comunicación en que interviene cualquier usuario certificado por esa CA.

Falsificación de un certificado – El método de autenticación fuerte protege contra la falsificación de un certificado consiguiendo que lo firme la CA. El método depende del mantenimiento del secreto de la clave secreta de la CA.

Falsificación de un token – El método de autenticación fuerte protege contra la falsificación de un token haciendo que lo firme el emisor. El método depende del mantenimiento del secreto de la clave secreta del emisor.

Reactuación de un token – Los métodos de autenticación unidireccionales y bidireccionales protegen contra la reactuación de un token por medio de la inclusión de una indicación de tiempo en el token. El método tridireccional lo hace por medio de la comprobación de los números aleatorios.

Ataque al sistema criptográfico – Los adelantos conseguidos en la teoría de los números, basados en las nuevas técnicas computacionales se reflejan en una mayor probabilidad de eficaces criptoanálisis de los sistemas; de ahí que sea razonable pensar en claves de mayor longitud.

Anexo G

Confidencialidad de los datos

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

G.1 Introducción

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

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

G.2 Confidencialidad de los datos por cifrado asimétrico

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

G.3 Confidencialidad de los datos por cifrado simétrico

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

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

Anexo H

Definición de referencia de los identificadores de objeto para algoritmo

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

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

```
AlgorithmObjectIdentifiers {joint-iso-ccitt ds(5) module(1) algorithmObjectIdentifiers(8) 2}
DEFINITIONS ::=
BEGIN

-- EXPORTS All --

-- Los tipos y valores definidos en este módulo son exportados para su utilización en otros módulos ASN.1 contenidos
-- en las especificaciones de directorio, y para uso de otras aplicaciones que los utilizarán para acceder a servicios
-- de directorio. Otras aplicaciones pueden utilizarlos para sus propios fines, pero esto no constreñirá las extensiones
-- y modificaciones necesarias para mantener o mejorar el servicio de directorio

IMPORTS
        algorithm, authenticationFramework
                FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 2}
        ALGORITHM
                FROM AuthenticationFramework authenticationFramework;

-- categorías de indentificador de objeto --

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

-- sinónimos --

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

-- algoritmos --

rsa     ALGORITHM                   ::=     {
        KeySize
        IDENTIFIED BY   id-ea-rsa }

KeySize         ::=     INTEGER

-- asignaciones de identificador de objeto --

id-ea-rsa               OBJECT IDENTIFIER  ::=     {id-ea 1}

-- las siguientes asignaciones de identificador de objeto reservan valores asignados a funciones desaconsejadas

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

END
```

Anexo J

Enmiendas y corrigendos

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

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

– Enmienda 1 para control de acceso

Esta edición de esta especificación de directorio incluye los siguientes corrigendos técnicos que subsanan los defectos señalados en los siguientes informes de defectos:

– Corrigendo técnico 1 (incluyendo los informes de defectos 009, 015, 016, 019, 031).