UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

# CCITT

COMITÉ CONSULTIVO
INTERNACIONAL
TELEGRÁFICO Y TELEFÓNICO

# X.509

(11/1988)

SERIE X: REDES DE COMUNICACIÓN DE DATOS:
LA GUIA

---

# LA GUIA – MARCO DE AUTENTICACION

Reedición de la Recomendación X.509 del CCITT
publicada en el Libro Azul, Fascículo VIII.8 (1988)

---

**NOTAS**

1 La Recomendación X.509 del CCITT se publicó en el fascículo VIII.8 del Libro Azul. Este fichero es un extracto del Libro Azul. Aunque la presentación y disposición del texto son ligeramente diferentes de la versión del Libro Azul, el contenido del fichero es idéntico a la citada versión y los derechos de autor siguen siendo los mismos (véase a continuación).

2 Por razones de concisión, el término «Administración» se utiliza en la presente Recomendación para designar a una administración de telecomunicaciones y a una empresa de explotación reconocida.

**Recomendación X.509**

## LA GUIA – MARCO DE AUTENTICACION [1)]

*(Melbourne, 1988)*

INDICE



---

1) La Recomendación X.509 y la norma ISO 9594-8, Information Processing Systems – Open Systems Interconnection – The Directory-Authentication Framework (Sistemas de procesamiento de información – Interconexión de sistemas abiertos – La Guía – Marco de autenticación) se redactaron en estrecha colaboración y están técnicamente alineadas.

## 0 Introducción

0.1 Esta Recomendación, junto con las otras de la serie, ha sido elaborada para facilitar la interconexión de los sistemas de procedimiento de información para suministrar servicios de *guía*. El conjunto de tales sistemas, junto con la información de guía que contienen, puede ser visto como un todo integrado, llamado la guía. La información contenida por la guía, conocida colectivamente por la base de información de la guía (BIG), se usa típicamente para facilitar la comunicación entre, con o sobre objetos tales como entidades de aplicación de ISA, personas, terminales y listas de distribución.

0.2 La guía desempeña un papel importante en la Interconexión de Sistemas Abiertos cuyo objetivo es el permitir, con un mínimo de concordancia técnica 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.

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

0.4 Esta Recomendación define un marco para el suministro de servicios de autenticación por la guía a sus usuarios. Estos usuarios incluyen la propia guía, así como otras aplicaciones y servicios. Incumbe a la guía la satisfacción de 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. La guía es el lugar natural porque ella contiene otras informaciones que se requieren para la comunicación y se obtienen con anterioridad al inicio de la comunicación. La obtención de la información de autenticación de un copartícipe potencial en la comunicación, desde la guía, es, con este enfoque, similar a la obtención de una dirección. Debido al vasto alcance de la guía para los fines de comunicación, se espera que este marco de autenticación será ampliamente usado por una gama de aplicaciones.

## 1 Alcance y campo de aplicación

1.1 Esta Recomendación:

- especifica la forma de la información de autenticación contenida por la guía;
- describe cómo puede obtenerse la información de autenticación a partir de la guía;
- enuncia los supuestos formulados en cuanto a la formación y al emplazamiento de esa información de autenticación en la guía;
- 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.

1.2 Esta Recomendación describe dos niveles de autenticación: autenticación simple, mediante el uso de una contraseña como verificación de una identidad alegada, 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.

1.3 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 limitada por los servicios proporcionados según una norma.

1.4 Incumbe a las normas definir las aplicaciones que usan el marco de autenticación para especificar los intercambios de protocolo que necesitan ser realizados para lograr la autenticación basada en la información de autenticación de la guía. El protocolo usado por las aplicaciones para obtener la información de autenticación de la guía es el protocolo de acceso a la guía (PAG), especificado en la Recomendación X.519.

1.5 El método de autenticación fuerte especificado en esta Recomendación 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 la guía como atributos, y ser comunicados libremente dentro del sistema de la guía y obtenidos por los usuarios de la guía del mismo modo que otra información de guía. Se supone que los certificados de usuario están formados por medios "fuera-de-línea", y que son introducidos en la guía por su creador. La generación certificados de usuario la efectúa cierta autoridad de certificación "fuera-de-línea" que está completamente separada de los ASG en la guía. En particular, no se imponen requisitos especiales a los suministradores de la guía para almacenar o comunicar certificados de usuario en una forma segura.

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

1.6 En general, el marco de autenticación no depende del uso de un determinado algoritmo criptográfico, siempre que tenga las propiedades descritas en el § 6.1. 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 C se presenta un ejemplo de un algoritmo criptográfico de claves públicas.

1.7 Análogamente, dos usuarios que deseen autenticar tienen que soportar la misma función hash (véase el § 3.3 f)) (usada en la formación de credenciales y testigos 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 D se presenta una breve introducción a las funciones hash, así como un ejemplo de función hash.

## 2 Referencias

2.1 ISO 7498-2 Information Processing Systems – Open Systems Interconnection – Security Architecture (Sistemas de procesamiento de información – interconexión de sistemas abiertos – arquitectura de seguridad).

## 3 Definiciones

3.1 Esta Recomendación emplea los siguientes términos generales relacionados con la seguridad definidos en la Parte 2 del Modelo de Referencia de ISA para Seguridad:

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 Los siguientes términos usados en esta Recomendación se definen en la Recomendación X.501:

a) *atributo*;

b) *base de información de la guía*;

c) *árbol de información de la guía*;

d) *nombre distinguido*;

e) *asiento*;

f) *objeto*;

g) *raíz*.

3.3 Los siguientes términos específicos se definen y usan en esta Recomendación:

a) *testigo de autenticación (testigo):* la información transportada durante un intercambio de autenticación fuerte y que puede usarse para autenticar a quien la envió;

b) *certificado de usuario (certificado):* la clave pública de un usuario, junto con alguna otra información, hecha infalsificable por cifrado con la clave secreta de la autoridad de certificación que la emitió;

c) *autoridad de certificación:* una 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;

d) *trayecto de certificación:* una secuencia ordenada de certificados de objetos en el AIG 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;

e) *sistema criptográfico, criptosistema:* una 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;

f) *función hash:* una función (matemática) que hace corresponder valores de un dominio vasto (que puede ser muy vasto) con una gama menor. Una función hash 'buena' es aquella 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;

g) *función unidireccional:* una función (matemática) que es 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;

h) *clave pública:* (en un criptosistema de claves públicas) la clave, de un par de claves de un usuario, que se conoce públicamente;

i) *clave privada (clave secreta – desaconsejada):* (en un criptosistema de claves públicas) la clave, de un par de claves de un usuario, que es conocida solamente por ese usuario;

j) *autenticación simple:* autenticación por medio de arreglos de contraseñas simples;

k) *política de seguridad:* el conjunto de reglas establecidas por la autoridad de seguridad que rigen la utilización y prestación de servicios y facilidades de seguridad;

l) *autenticación fuerte:* autenticación por medio de credenciales derivadas criptográficamente;

m) *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.

n) *número secuencial de certificado:* valor entero, único en la AC expedidora, que va asociado inequívocamente a un certificado expedido por dicha AC.

## 4 Notación y abreviaturas

4.1 La notación usada en esta Recomendación se define en el cuadro 1/X.509.

*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.

4.2 En esta Recomendación se usan las siguientes abreviaturas:

AC Autoridad de certificación

AIG Arbol de información de la guía

BIG Base de información de la guía

CSCP Criptosistema de claves públicas

CUADRO 1/X.509

**Notación**

| NOTACION | 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 encifrado añadido |
| CA(X) | una autoridad de certificación del usuario X |
| $CA^n(X)$ | (donde n > 1): AC(AC(...n veces ...(X))) |
| $X_1\langle\langle X_2\rangle\rangle$ | el certificado de usuario $X_2$ emitido por la autoridad de certificación $X_1$ |
| $X_1\langle\langle X_2\rangle\rangle X_2\langle\langle X_3\rangle\rangle$ | una cadena de certificados (puede tener una longitud arbitraria), donde cada ítem es el certificado para la autoridad de certificación que produjo el siguiente. Es funcionalmente equivalente al siguiente certificado $X_1\langle\langle X_{n+1}\rangle\rangle$. Por ejemplo la posesión de A<<B>>B<<C>> confiere la misma capacidad que A<<C>>, a saber, la aptitud para hallar Cp cuando se da Ap |
| $X_{1p} \cdot X_1\langle\langle X_2\rangle\rangle$ | la operación de desenvolver un certificado (o cadena de certificados) para extraer una clave pública. Es un operador infijo, cuyo operando izquierdo es la clave pública de una autoridad de certificación, y cuyo operando derecho es un certificado emitido por esa autoridad de certificación. El resultado es la clave pública del usuario cuyo certificado es el operando derecho. Por ejemplo: $A_p$ .A<<B>>B<<C>> denota la operación de usar la clave pública de A para obtener la clave pública de B, $B_p$, de su certificado, seguido por el uso de $B_p$ para desenvolver el certificado de C. El resultado de la operación es la clave pública de C, $C_p$ |
| A→B | un trayecto de certificación de A a B, formado por una cadena de certificados, que comienza por $AC(A)\langle\langle AC^2(A)\rangle\rangle$ y termina por AC(B)<<B>>. |

## 5 Procedimiento de autenticación simple

5.1 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 AUG y un ASG, o entre un ASG y otro ASG. 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 destinatario, para su evaluación;

b) la transferencia del nombre distinguido del usuario, la contraseña, y un número aleatorio y/o una indicación de tiempo, todo lo cual se protege mediante la aplicación de una función unidireccional;

c) la transferencia de la información protegida descrita en b) junto con un número aleatorio y/o una indicación de tiempo, todo lo cual se protege por la aplicación de una función unidireccional.

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

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

5.2 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.

5.3 El procedimiento general para la obtención de una autenticación simple se muestra en la figura 1/X.509.

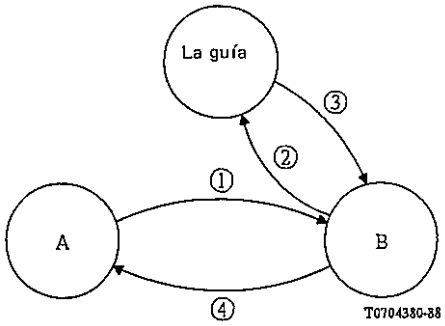



FIGURA 1/X.509

**Procedimiento de autenticación simple no protegida**

5.3.1 Comprende los siguientes pasos:

1) un usuario originador A envía su nombre distinguido y contraseña a un usuario receptor (o destinatario) B;

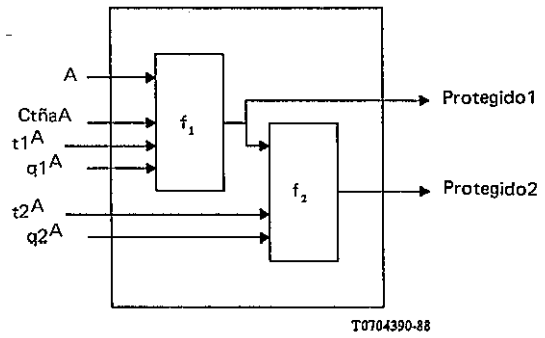
2) B envía el nombre distinguido contemplado y la contraseña de A a la guía, donde la contraseña se comprueba contra la mantenida como el atributo de **Contraseña de Usuario** dentro del asiento de la guía para A (usando la operación comparar de la guía);

3) la guía confirma (o rechaza) a B que las credenciales son válidas;

4) el éxito (o fracaso) de la autenticación puede comunicarse a A.

5.3.2 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).

5.4 La figura 2/X.509 muestra dos métodos que pueden emplearse para generar información de identificación protegida. $f_1$ y $f_2$ son funciones unidireccionales (que pueden ser idénticas o diferentes) y las indicaciones de tiempo y los números aleatorios son opcionales y están sujetos a acuerdo bilateral.




A = Nombre distinguido de usuario
$t^A$ = Indicaciones de tiempo
CtñaA = Contraseña de A
$q^A$ = Números aleatorios, y opcionalmente con un contador incluido

FIGURA 2/X.509

**Autenticación simple protegida**

5.4.1 La figura 3/X.509 ilustra el procedimiento de autenticación simple protegida.

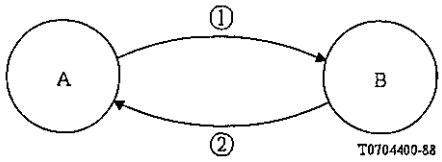



FIGURA 3/X.509

**El procedimiento de autenticación simple protegida**

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

1) el usuario de origen, usuario A, envía su información de identificación protegida (Autenticador1) al usuario B. La protección se consigue aplicando la función unidireccional ($f_1$ de la figura 2/X.509, donde la indicación de tiempo y/o el número aleatorio (si se utiliza) tienen por finalidad minimizar la reproducción y ocultar la contraseña.

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

   Protegido1 = $f_1$ ($t1^A$; $q1^A$; Ctña 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 la indicación de tiempo, el nombre distinguido y, opcionalmente, la indicación de tiempo adicional 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.

5.4.2 El procedimiento descrito en el § 5.4.1 puede modificarse para dar una mayor protección (mediante el empleo de $f_1$ y $f_2$). Las diferencias principales son las siguientes:

1) A envía su información de identificación protegida (adicionalmente) (Autenticación2) a B. Una protección adicional se obtiene aplicando una segunda función unidireccional, $f_2$, como se ilustra en la figura 2/X.509. Esta mayor protección adopta la forma siguiente:

   Protegido2 = $f_2$ ($t2^A$, $q2^A$, 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 del § 5.4.1);

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

*Nota* – Los procedimientos definidos en esta cláusula se especifican sobre la base de A y B. Atendiendo a la aplicación a la guía (especificada en las Recomendaciones X.511 y X.518), A podría ser un AUG vinculado a un ASG, B; alternativamente A, podría ser un ASG vinculado a otro ASG, B.

5.5 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 ATTRIBUTE-SYNTAX
                    OCTET STRING (SIZE (0..ub-user-password))
                    MATCHES FOR EQUALITY
```

5.6 La siguiente macro NSA.1 puede utilizarse para definir el tipo datos que se obtiene al aplicar una función unidireccional a otro tipo dado de datos.

```
PROTECTEDMACRO ::= SIGNATURE
```

SECCION 2 – *Autenticación fuerte*

## 6 Bases de autenticación fuerte

6.1 El enfoque a la autenticación fuerte adoptado en esta Recomendación utiliza las propiedades de una familia de sistemas criptográficos, conocidos como criptosistemas de claves públicas (CSCP). 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 B da una breve introducción a estos criptosistemas y sus propiedades útiles para la autenticación. Para que un CSCP sea utilizable en este marco de autenticación, actualmente, debe tener la propiedad de que ambas claves del par de claves puedan ser usadas para el cifrado, empleándose la clave secreta para descifrar si se usó la clave pública, y empleándose la clave pública para descifrar si se usó la clave secreta. Dicho sea en otras palabras, $X_p \cdot X_s = X_s \cdot X_p$ siendo $X_p/X_s$ funciones de cifrado/descifrado que utilizan las claves pública/secreta de X.

*Nota* – En una futura y posible ampliación podrán especificarse otros tipos de CSCP, es decir, tipos que no requieran la propiedad de permutabilidad y que puedan ser soportados sin grandes modificaciones de esta Recomendación.

6.2 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 C.

6.3 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.

6.4 Cada usuario se identifica por su 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.

6.5 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 del asiento del usuario en la guía es inmediata, la verificación de su corrección plantea ciertos problemas. Puede haber varias formas posibles de realizar esto: el § 7 describe un proceso por el cual una clave pública

de usuario puede ser verificada por referencia a la guía. Este proceso sólo puede operar si hay una cadena ininterrumpida de puntos de confianza, en la guía, 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.

## 7 Obtención de una clave pública de usuario

7.1 Para que un usuario confíe el procedimiento de autenticación, tiene que obtener la clave pública del otro usuario desde una fuente en la cual confía. Tal fuente, llamada autoridad de certificación (AC), usa el algoritmo de clave pública para certificar la clave pública, produciendo un certificado. El certificado, cuya forma se especifica en el § 7.2, 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 la guía, sin que ésta tenga que tomar disposiciones especiales para protegerlos.

*Nota* – Aunque las AC están definidas inequívocamente por un nombre distinguido en el AIG, esto no implica que exista una relación entre la organización de las AC y el AIG.

7.2 Una autoridad de certificación produce el certificado de un usuario firmando (véase el § 8) una colección de informaciones, incluidos el nombre distinguido y la clave pública del usuario. Específicamente, el certificado de un usuario con el nombre distinguido A, producido por la autoridad de certificación AC, tiene la forma siguiente:

$AC\langle\langle A\rangle\rangle = AC\ \{NS, IA, AC, A, Ap, T^A\}$

donde NS es el número de serie de certificado, IA es el identificador del algoritmo utilizado para firmar el certificado, 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 ACP. El siguiente tipo de datos NSA.1 puede usarse para representar certificados:

```
Certificate          ::=    SIGNED SEQUENCE{
    version                          [0]Version DEFAULT 1988
    serialNumber                     SerialNumber,
    signature                        Algorithmidentifier
    issuer                           Name
    validity                         Validity,
    subject                          Name,
    subjectPublicKeyInfo             SubjectPublicKeyInfo}

Version              ::=    INTEGER { 1988(0)}
SerialNumber         ::=    INTEGER

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

SubjectPublicKeyInfo ::=
    SEQUENCE{
        algorithm          AlgorithmIdentifier
        subjectKey         BIT STRING}

AlgorithmIdentifier  ::=
    SEQUENCE{
        algorithm          OBJECT IDENTIFIER
        parameters         ANY DEFINED BY algorithm
                               OPTIONAL}
```

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

7.4 Si el usuario A, que trata de obtener la clave pública del usuario B, ya ha obtenido la clave pública de AC(B) el proceso habrá terminado. A fin de permitir que A obtenga la clave pública de AC(B), el asiento de guía 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.

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

- comienza por el certificado inverso producido por AC(A), a saber $AC(A)<<X^1>>$ 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 en el árbol de información de la guía, entre dos usuarios que desean autenticar. El método preciso empleado por los usuarios A y B para obtener trayectos de certificación A → B y B → A puede variar. Una manera de facilitar esto consiste en organizar una jerarquía de ACs, que puede o no coincidir con la totalidad o una parte de la jerarquía del AIG. La ventaja de esto es que los usuarios que tienen ACs en la jerarquía pueden establecer entre sí un trayecto de certificación utilizando la guía sin ninguna información previa; para que esto sea posible, cada AC puede almacenar un certificado (directo) y un certificado inverso designado como correspondiente a su AC superior.

7.6 Los certificados están contenidos en asientos de la guía como atributos de tipo **certificado usuario**, **certificado AC** y **par de certificados cruzados**. Estos tipos de atributos son conocidos por la guía. Se puede actuar sobre estos atributos utilizando las mismas operaciones de protocolo empleadas para atributos. La definición de estos tipos puede encontrarse en el § 3.3 de esta Recomendación; la especificación de estos tipos de atributo es la siguiente:

```
UserCertificate          ::=    ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX Certificate

CACertificate            ::=    ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX Certificate

CrossCertificatePair     ::=    ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX CertificatePair

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

Un usuario puede obtener uno o más certificados de una o más autoridades de certificación. Cada certificado comporta el nombre de la autoridad de certificación que lo expidió.

7.7 En el caso general, antes de que los usuarios puedan autenticar mutuamente, la guía 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 de la guía 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 directamente los certificados de cada uno de los otros;

b) un usuario pudiera almacenar claves públicas, certificados y certificados inversos de todas las autoridades de certificación entre el usuario y la raíz del AIG. 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 AC en particular, este usuario pudiera aprender el trayecto de certificación a ese AC y el trayecto de certificación de retorno desde ese AC, con lo que sólo sería necesario obtener el certificado del otro usuario, desde la guía;

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 a la guía.

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.

7.8 (Ejemplo). La figura 4/X.509 ilustra un ejemplo hipotético de un fragmento del AIG, en el cual las AC forman una jerarquía. Además de la información indicada en las AC, se supone que cada usuario conoce la clave pública de su autoridad de certificación, y sus propias claves pública y secreta.

7.8.1 Si las AC de los usuarios forman una jerarquía, A puede obtener los siguientes certificados de la guía para establecer un trayecto de certificación a B:

X<<W>>, W<<V>>, V<<Y>>, Y<<Z>>, Z<<B>>.

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

Bp = Xp · X<<W>> W<<V>> V<<Y>> Y<<Z>> Z<<B>>.

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

Z<<Y>>, Y<<V>>, V<<W>>, W<<X>>, X<<A>>.

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

Ap = Zp · Z<<Y>> Y<<V>> V<<W>> W<<X>> X<<A>>.

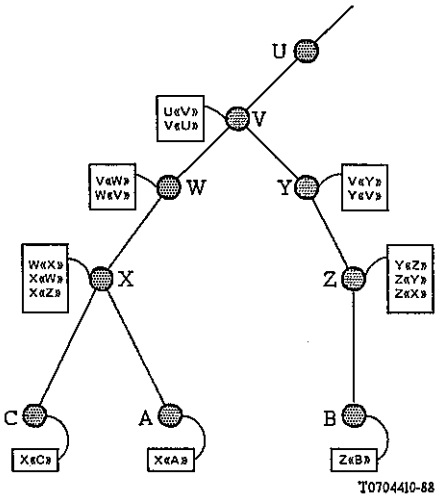



FIGURA 4/X.509

**Jerarquía de AC – Un ejemplo hipotético**

7.8.2 Aplicando las optimizaciones del § 7.7:

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 · X<<C>>

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

Ap = Xp · X<<A>>.

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

V<<Y>>, Y<<Z>>, Z<<B>>

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

Z<<Y>>, Y<<V>>.

c) suponiendo que A comunica frecuentemente con usuarios certificados por Z, él puede aprender (además de las claves públicas aprendidas en b)) V<<Y>>, Y<<V>>, Y<<Z>>, y Z<<Y>>. Para comunicar con B, sólo necesita por consiguiente obtener Z<<B>> de la guía;

d) suponiendo que los usuarios certificados por X y Z comunican frecuentemente, entonces X<<Z>> estaría contenido en el asiento de la guía para X, y viceversa (esto se muestra en la figura 4/X.509). Si A quiere autenticar hacia B, sólo necesita obtener:

X<<Z>>, Z<<B>>

para formar el trayecto de certificación, y:

Z<<X>>

para formar el trayecto de certificación de retorno;

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

Cp = Xp · X<<C>>

y

Ap = Xp · X<<A>>.

7.8.3 En el caso más general, las autoridades de certificación no guardan una relación jerárquica. En el ejemplo hipotético de la figura 5/X.509, supóngase que un usuario D, certificado por U, desea autenticar al usuario E, certificado por W. El asiento de guía del usuario D contendrá el certificado U<<D>> y el asiento del usuario E contendrá el certificado W<<E>>.

Sea V una AC con la cual las AC, 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 la guía, certificados U<<V>>, V<<U>>, W<<V>> y V<<W>>. Supóngase que U<<V>> y W<<V>> están almacenados en el asiento de V, V<<U>> está almacenado en el asiento de U, y V<<W>> está almacenado en el asiento 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 ACs 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>>.

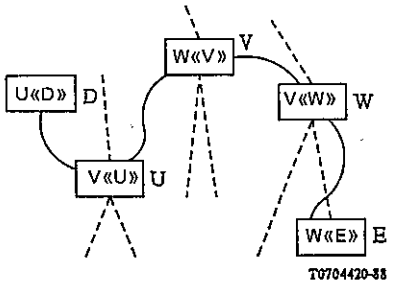



FIGURA 5/X.509

**Ejemplo de trayecto de certificación no jerárquico**

## 8 Firmas digitales

En esta sección no se pretende especificar una norma para firmas digitales en general, sino especificar los medios para firmar los testigos en la guía.

8.1 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/X.509). Así

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

*Nota* – 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.

8.2 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.

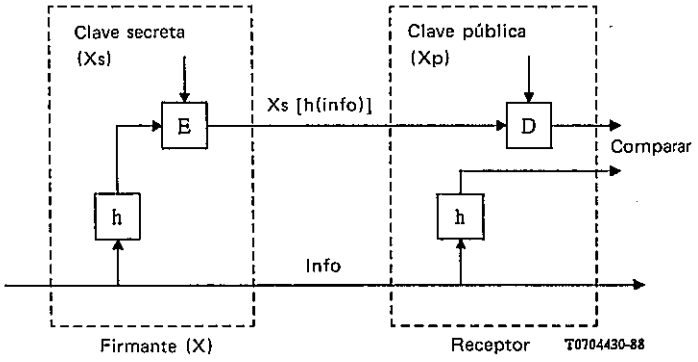



FIGURA 6/X.509

**Firmas digitales**

8.3 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. Una función hash que tiene probabilidad de ser ampliamente usada se especifica en el anexo D.

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

8.4 El cifrado de algún ítem de datos puede describirse utilizando la siguiente macro NSA.1:

```
ENCRYPTED MACRO   ::=
BEGIN

TYPE NOTATION     ::= type (ToBeEnciphered)
VALUE NOTATION    ::= value (VALUE BIT STRING)
END
```

El valor de la cadena de bits se genera tomando los octetos que forman la codificación completa (utilizando las reglas de codificación básica NSA.1) del valor del tipo A-Cifrar (**ToBeEnciphered**) y aplicando un procedimiento de cifrado a esos octetos.

*Nota 1* – 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 2* – El procedimiento de encripción deberá admitir como entrada una cadena de octetos y generar una cadena única de bits, como resultado.

*Nota 3* – 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 Recomendación.

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

```
SIGNED MACRO        ::=
BEGIN

TYPE NOTATION       ::= type (ToBeSigned)

VALUE NOTATION      ::= value (VALUE

    SEQUENCE{
        ToBeSigned,
        AlgorithmIdentifier,
        -- del algoritmo utilizado
        -- para computar la firma
        ENCRYPTED OCTET STRING
        -- donde la cadena de octetos
        -- es el resultado de aplicar
        -- la función hash del valor de
        -- 'ToBeSigned' --}

END -- of SIGNED         )
```

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

```
SIGNATURE MACRO         ::=
BEGIN

TYPE NOTATION           ::= type (OfSignature)
VALUE NOTATION          ::= value (VALUE

    SEQUENCE{
        AlgorithmIdentifier,
        -- del algoritmo utilizado
        -- para computar la firma
        ENCRYPTED OCTET STRING
        -- donde la cadena de octetos es una función
        -- (por ejemplo, una versión comprimida o tratada
        -- por la función hash) del valor'OfSignature',
        -- que puede incluir el identificador del algoritmo
        -- utilizado para computar la firma --}

END -- of SIGNATURE      )
```

8.7 A fin de permitir la validación de los tipos **SIGNED** y **SIGNATURE** en un entorno distribuido, se requiere codificación distinguida. Una codificación distinguida de un valor de datos **SIGNED** o **SIGNATURE** se obtendrá aplicando las Reglas de Codificación Básicas definidas en la Recomendación X.209 con las siguientes limitaciones:

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

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

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

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

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

f) si el valor de un tipo Booleano es verdadero, el octeto de contenido de la codificación deberá fijarse a $'FF'_{16}$ ;

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

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

## 9 Procedimiento de autenticación fuerte

### 9.1 *Visión de conjunto*

9.1.1 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.

9.1.2 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 el § 9.2 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 w'originalidadw' (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 el § 9.3, 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 el § 9.4, 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 a la guía, como se describió en el § 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 secciones 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.

### 9.2 *Autenticación unidireccional*

Se siguen los siguientes pasos que muestra la figura 7/X.509:

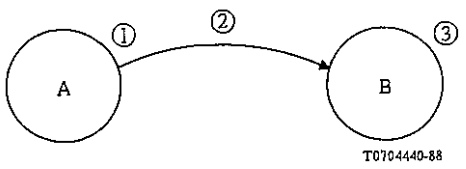



FIGURA 7/X.509

**Autenticación unidireccional**

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 facultativa) 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 w'encDataw' 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 él mismo (B) es el receptor deseado;

   d) comprueba que la indicación de tiempo está actual;

   e) opcionalmente, comprueba que $r^A$ no ha sido maniobrada. 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 a que se aplicó la función hash del testigo durante el intervalo de tiempo $t^A$.

9.3 *Autenticación bidireccional*

Se siguen los pasos indicados en la figura 8/X.509.

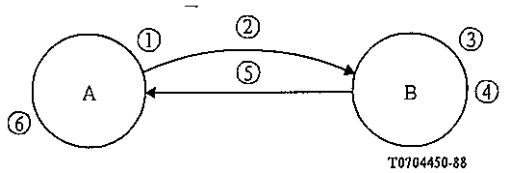



FIGURA 8/X.509

**Autenticación bidireccional**

1) Como en el § 9.2.

2) Como en el § 9.2.

3) Como en el § 9.2.

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 'enData'):

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 maniobrado [véase el § 9.2 paso 3 e)].

9.4 *Autenticación tridireccional*

Se siguen los siguientes pasos indicados en la figura 9/X.509:

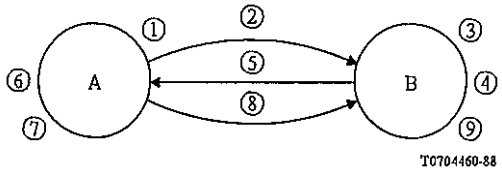



FIGURA 9/X.509

**Autenticación tridireccional**

1) Como en el § 9.3.

2) Como en el § 9.3. La indicación de tiempo $t^A$ puede ser cero.

3) Como en el § 9.3, excepto que la indicación de tiempo no necesita ser comprobada.

4) Como en el § 9.3.

5) Como en el § 9.3. La indicación de tiempo $t^B$ puede ser cero.

6) Como en el § 9.3, 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$\{r^B\}$.

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.

## 10 Gestión de claves y certificados

10.1 *Generación de pares de claves*

10.1.1 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 a que pertenecen.

Los datos de la clave no son fáciles de recordar por el usuario humano, por lo que hay que emplear un método apropiado para almacenarla en un modo transportable conveniente. Un mecanismo posible sería el uso de una "tarjeta inteligente" que podría contener las claves secreta y 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 NIP (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 Recomendación.

10.1.2 Hay tres modos en los cuales un par de claves del usuario pueden ser producidos como se describe en el § 10.1.2.1 a 10.1.2.3.

10.1.2.1 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 C.

10.1.2.2 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 más las propias claves. Hay que emplear medidas de seguridad adecuadas para garantizar que la tercera entidad y las operaciones de datos no son objeto de fraudes.

10.1.2.3 El par de claves se genera por la AC. Este es un caso especial del § 10.1.2.2 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 necesarias de seguridad física. Este método tiene la ventaja de no requerir una transferencia securizada de datos a la AC para la certificación.

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

### 10.2 *Gestión de certificados*

10.2.1 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 estar satisfecha de la identidad de un usuario antes de crear un certificado para el mismo;

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

10.2.2 La producción de un certificado ocurre fuera de línea y no deberá realizarse 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 AC, nunca se conoce excepto en la AC aislada y físicamente segura, la AC secreta sólo puede ser entonces averiguada por un ataque a la propia AC, lo que hace poco probable un compromiso.

10.2.3 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 AC expidiera un certificado para un usuario con una clave pública que haya sido objeto de un fraude;

b) si se emplea el medio de generación de los pares de claves del § 10.1.2.3, no se necesita una transferencia segura;

c) si se emplea el medio de generación de pares claves descrito en los § 10.1.2.1 ó 10.1.2.2, el usuario puede emplear diferentes métodos (en línea o fuera de línea) para comunicar su clave pública a la AC de una manera segura. Los métodos en línea pueden proporcionar una mayor flexibilidad para las operaciones a distancia efectuadas entre el usuario y la AC.

10.2.4 Un certificado es una información disponible públicamente, y no se necesita emplear medidas de seguridad específicas con respecto a su transporte a la guía. 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 asiento de la guía en un acceso ulterior a la guía. Alternativamente la AC podría custodiar el certificado para el usuario, en cuyo caso a este agente tendrían que otorgársele derechos de acceso adecuados.

10.2.5 Los certificados tendrán asociada cierta duración, al final de la cual caducan (expiran). A fin de asegurar la continuidad del servicio, la AC garantizará el suministro oportuno de certificados de sustitución que reemplazan a los caducados o próximos a caducar. Los distintos aspectos de esta cuestión se describen en los § 10.2.5.1 y 10.2.5.2.

10.2.5.1 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 AC tengan que instalar y distribuir un gran número de certificados que pudieran agotarse en la misma fecha de expiración.

10.2.5.2 Los certificados caducos normalmente serán sacados de la guía. Es cuestión de política de seguridad y de responsabilidad de la AC mantener los antiguos certificados durante cierto periodo de tiempo si no se presta el servicio de "incuestionabilidad de los datos" (denominado también "norepudio de los datos").

10.2.6 Los certificados pueden ser revocados antes de su expiración, por ejemplo si se supone que la clave secreta del usuario puede ser objeto de maniobras irregulares, o si el usuario ya no deberá ser certificado por la AC, o si se supone que el certificado de la AC ha sido objeto de maniobras irregulares. Los distintos aspectos de esta cuestión se describen en los § 10.2.6.1 a 10.2.6.4.

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

10.2.6.2 La AC 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 AC, conocidos por la AC, certificados por la AC.

Ambas listas certificadas existirán, incluso si estuviesen vacías.

10.2.6.3 El mantenimiento de asientos de la guía afectados por las listas de revocaciones, por la AC, es responsabilidad de la guía y sus usuarios, quienes actúan de acuerdo con la política de seguridad. Por ejemplo, el usuario puede modificar su asiento de objeto reemplazando el antiguo certificado por uno nuevo. Este último se utilizará entonces para autenticar el usuario ante la guía.

10.2.6.4 Las listas de revocaciones ("listas negras") se mantienen dentro de asientos como atributos de tipos "**lista de revocaciones de certificados**" y "**lista de revocaciones de autoridad**". Esos atributos pueden ser operados utilizando los mismos procedimientos empleados para otros atributos. Estos tipos de atributo se definen como sigue:

```
CertificateRevocationList       ::=     ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX CertificateList

AuthorityRevocationList         ::=     ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX CertificateList

CertificateList                 ::=     SIGNED SEQUENCE{
    signature AlgorithmIdentifier,
    issuer Name,
    lastUpdate UTCTime,
    revokedCertificates
        SIGNED SEQUENCE OF SEQUENCE{
            signature AlgorithmIdentifier,
            issuer Name, CertificateSerialNumber subject,
            revocationDate UTCTime}
                    OTPIONAL}
```

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

*Nota 2* – Si un servicio de "incuestionabilidad de los datos" depende de claves proporcionadas por la AC, dicho servicio deberá asegurar que todas las claves pertinentes de la AC (revocadas o caducas) y las listas de revocaciones con indicación de tiempo son archivadas y certificadas por una autoridad "corriente".

ANEXO A

(a la Recomendación X.509)

**Requisitos de seguridad**

Este anexo no forma parte integrante de esta Recomendación.

[Material adicional sobre este tema puede encontrarse en la Norma ISO 7498 – Information Proccessing Systems – OSI Reference Model – Part 2, Security Architecture.]

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

## A.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) *Impostura (o mascarada):* pretensión de un usuario de ser otro diferente para ganar acceso a información u obtener privilegios adicionales.

c) *Reactuación:* grabación 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, eliminación 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 la demora de operaciones críticas en cuanto al tiempo.

*Nota* – 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* – El encaminamiento incorrecto ocurrirá naturalmente en las capas 1 a 3 de ISA, 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, dirección, secuencia, tipo, cantidad, etc.).

*Nota* – Los peligros de análisis de tráfico no están naturalmente limitados a una capa ISA 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 aleatorios o cifrados.

## A.2 *Servicios de seguridad*

Para la protección contra los peligros conocidos deben prestarse diversos servicios de seguridad. Los servicios de seguridad suministrados por el marco de autenticación se realizan por medio del mecanismo de seguridad descrito en el § A.3 de este anexo.

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.

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 impostura 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 la guía 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 cualquier tercero en cualquier momento.

### A.3 *Mecanismos de seguridad*

Los mecanismos de seguridad que se describen aquí realizan los servicios de seguridad descritos en el § A.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 el 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 se procesa 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.

A.4 *Peligros contra los que protegen los servicios de seguridad*

La tabla al final de este anexo indica los peligros de seguridad contra los que cada servicio de seguridad puede proteger. La presencia de un asterisco (*) indica que un cierto servicio de seguridad ofrece protección contra cierto peligro.

A.5 *Negociación de servicios y mecanismos de seguridad*

La provisión de características 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.

SERVICIOS

| PELIGROS | Autenticación de entidad | Confidencialidad de datos | Integridad de datos | No-Repudio |
|---|---|---|---|---|
| Intercepción de Identidad | * (si se requiere) | | | |
| Intercepción de Datos | | * | | |
| Impostura | * | | | |
| Re-actuación | * (identidad) | | * (datos) | * |
| Manipulación | | | * | * |
| Repudio | | | | * |

ANEXO B

(a la Recomendación X.509)

**Una introducción a la criptografía de claves públicas**

Este anexo no forma parte integrante de esta Recomendación.

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 (CSCP), 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 B-1/X.509.

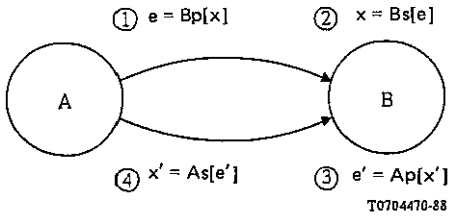



FIGURA B-1/X.509

**Uso de un CSCP 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 B-1/X.509).

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], 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'], o x' = As[Ap[x']]

Por este medio, A y B han intercambiado la información secreta x y x'. Esta información no puede ser obtenida por ninguno 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 xw' 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 CSCP 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 CSCP que tienen esta propiedad (permeabilidad) son apropiados para uso en este marco de autenticación. En el anexo C se describe uno de estos algoritmos.

Para más información, véase:

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

ANEXO C

(a la Recomendación X.509)

## El criptosistema de claves públicas RSA

Este anexo no forma parte integrante de esta Recomendación.

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

### C.1 *Alcance y campo de aplicación*

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

### C.2 *Referencias*

Para más información, véase:

1) Aspectos generales

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

2) Generación de claves

   GORDON, J. – Strong RSA Keys, *Electronics Letters*, 20, 5, 514-516.

3) Descifrado

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

### C.3 *Definiciones*

a) *Clave pública:* el par de parámetros formado por el exponente público y el módulo aritmético.

   *Nota 1* – El elemento de datos NSA.1 **subjectPublicKey**, definido como **BIT STRING** (véase el anexo G) 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 básicas de NSA.1.

b) *Clave secreta:* el par de parámetros formado por el exponente secreto y el módulo aritmético.

### C.4 *Símbolos y abreviaturas*

X,Y bloques de datos que son aritméticamente menores que el módulo

n el módulo aritmético

e el exponente público

d el exponente secreto

p,q los 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.

mod n módulo aritmético n.

### C.5 *Descripción*

Este algoritmo asimétrico usa la función de potenciación para la transformación de bloques de datos tales que:

$y = X^e \bmod n$ con $0 \leq X < n$

$X = Y^d \bmod n$ $0 \leq Y < n$

que puede ser satisfecha, por ejemplo, por

ed mod ecm (p–1,q–1) = 1, o

ed mod (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 l). El entero se forma entonces como la suma de los bits después de darle un peso de $2^{l-1}$ 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.

### C.6 *Requisitos de seguridad*

#### C.6.1 *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 soporte físico, 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.

#### C.6.2 *Generación de claves*

La seguridad del criptosistema 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 referencia 2), § C.2]:

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" constituidos 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 descripción por un factor de uno a cuatro. La decisión de mantener p y q se considera un asunto local [referencia 3)].

Se tiene que asegurar que $e > \log_2(n)$ para prevenir el ataque tomando la ésima raíz mod n para revelar el texto sencillo.

### C.7 *Exponente público*

El exponente público (e) debe 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 y la complejidad de la transformación (véase la nota 1).

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

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

= 65537 decimal, y

= 1 0000 0000 0000 0001 binario.

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

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

### C.8 *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 conformes con este anexo y que sean mutuamente compatibles.

## ANEXO D

(a la Recomendación X.509)

## Funciones hash

Este anexo no forma parte integrante de esta Recomendación.

### D.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 imposible construir un mensaje de entrada que dé como hash este resultado.

b) La función hash tiene que estar libre de colisiones, es decir, tiene que ser computacionalmente imposible construir dos mensajes de entrada distintos que den en hash este mismo resultado.

### D.2 *Descripción de una función hash*

La siguiente función hash ("cuadrado-mod-n") realiza la compresión de los datos en un bloque, operando bloque por bloque.

El hasheado se hace en tres pasos principales:

1) La cadena de datos que se va a hashear se divide en bloques B de la misma longitud. Esta longitud se determina por las características del criptosistema asimétrico que se usa para el firmado. Con el criptosistema RSA, esta longitud (en octetos) es el mayor valor entero de l, para el que en módulo n se cumple:

   $16\, l < \log_2 n$.

2) Por razones de no-invertibilidad, cada octeto del bloque se divide por la mitad. Cada una de las mitades es hasheada w'rellenadaw' por unos binarios. Por medio de esta zonificación, se introduce una rigidez o redundancia que incrementa considerablemente la propiedad de no invertibilidad de la función hash. Cada bloque generado en el paso 1 se ensancha a la longitud del módulo n.

3) Cada bloque resultante del paso 2 se suma en módulo 2 al bloque precedente, se eleva al cuadrado y se reduce en módulo n, hasta que todos los m bloques son procesados m.
   sigue:

   Así pues, el resultado es el valor $H_m$, donde:

   $H_0 = 0$
   $H_i = (H_{i-1} \oplus B_i)^2 \bmod n$, para $1 \leq i \leq m$

   Si el último bloque está incompleto, se rellena con "1"s.

ANEXO E

(a la Recomendación X.509)

## Peligros contra los que ofrece protección el método de autenticación fuerte

Este anexo no forma parte integrante de esta Recomendación.

El método de autenticación fuerte que se describe en esta Recomendación ofrece protección contra los peligros como se describe en el anexo A 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 AC* – el hecho de que la clave secreta de una AC 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 AC.

*Inducir a error a la AC para que cree un certificado no válido* – el hecho de que las AC funcionen "fuera de línea" da cierta protección. Recae sobre la AC 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 AC deshonesta y un usuario* – un ataque de este tipo hará fracasar este método. Esto podría constituir una traición a la confianza depositada en la AC. Las consecuencias de una AC deshonesta se limitan a un trastorno de la comunicación en que interviene cualquier usuario certificado por esa AC.

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

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

*Reactuación de un testigo* – los métodos de autenticación unidireccionales y bidireccionales protegen contra la reactuación de un testigo por medio de la inclusión de una indicación de tiempo en el testigo. 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 F

(a la Recomendación X.509)

## Confidencialidad de los datos

Este anexo no forma parte integrante de esta Recomendación.

### F.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 el § 9 o por algún otro proceso de intercambio de claves; esto último está fuera del alcance de este anexo.

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

### F.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 secreta.

### F.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.

Cuando un intercambio de autenticación de acuerdo al § 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 secretas depende de la transformación que se utilice. Las partes tienen que estar seguras de que son claves fuertes. Esta Recomendación no especifica cómo se hace esta selección, aunque es evidente que esto debería ser acordado por las partes interesadas, o especificado en otras Recomendaciones.

ANEXO G

(a la Recomendación X.509)

## Marco de autenticación en ASN.1

Este anexo forma parte de la Recomendación.

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

```
AuthenticationFramework {joint-iso-ccitt ds(5) modules(1)
                         authenticationFramework(7)}

DEFINITIONS ::=
BEGIN

EXPORTS AlgorithmIdentifier, AuthorityRevocationList, CACertificate, Certificate,
        Certificates, CertificationPath, CertificateRevocationList, UserCertificate,
        CrossCertificatePair, UserPassword, ALGORITHM,
        ENCRYPTED, PROTECTED, SIGNATURE, SIGNED;
```

```
IMPORTS
    informationFramework, selectedAttributeTypes, upperBounds
        FROM UsefulDefinitions  {joint-iso-ccitt ds(5)modules(1)
                                 usefulDefinitions(0)}
      Name, ATTRIBUTE,ATTRIBUTE-SYNTAX
        FROM InformationFramework informationFramework

ub-user-passwordFROM UpperBounds upperBounds;

-- tipos

Certificate       ::=  SIGNED SEQUENCE{
                           version                 [0] Version DEFAULT 1988,
                           serialNumber            SerialNumber,
                           signature               AlgorithmIdentifier,
                           issuer                  Name,
                           validity                Validity,
                           subject                 Name,
                           subjectPublicKeyInfo    SubjectPublicKeyInfo}

Version           ::=  INTEGER { 1988(0)}
SerialNumber      ::=  INTEGER

Validity          ::=  SEQUENCE{
                           notBefore               UTCTime
                           notAfter                UTCTime}

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

AlgorithmIdentifier  ::=  SEQUENCE{
                              algorithm            OBJECT IDENTIFIER,
                              parameters           ANY DEFINED BY algorithm OPTIONAL}

Certificates ::=     SEQUENCE{
                           certificate             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,
                            lastUpdate             UTCTime,
                            revokedCertificates        SIGNEDSEQUENCE OF SEQUENCE{
                                signature          AlgorithmIdentifier,
                                issuer             Name,
                                userCertificate    SerialNumber,
                                revocationDate     UTCTime}
                                                       OPTIONAL}

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

-- tipos de atributo

UserCertificate          ::=  ATTRIBUTE
                                  WITH ATTRIBUTE-SYNTAXCertificate

CACertificate            ::=  ATTRIBUTE
                                  WITH ATTRIBUTE-SYNTAXCertificate
```

```
CrossCertificatePair            ::=     ATTRIBUTE
                                            WITH ATTRIBUTE-SYNTAXCertificatePair
CertificateRevocationList       ::=     ATTRIBUTE
                                            WITH ATTRIBUTE-SYNTAXCertificateList
AuthorityRevocationList         ::=     ATTRIBUTE
                                            WITH ATTRIBUTE-SYNTAXCertificateList
UserPassword                    ::=     ATTRIBUTE
                                            WITH ATTRIBUTE-SYNTAX
                                                OCTETSTRING(SIZE(0...ub-user-password))
                                                MATCHES FOR EQUALITY

-- macros

ALGORITHM MACRO    ::=
BEGIN
TYPE NOTATION      ::= "PARAMETER"  type
VALUE NOTATION     ::= value(VALUE OBJECT IDENTIFIER)
END -- of ALGORITHM

ENCRYPTED MACRO    ::=
BEGIN

TYPE NOTATION   ::= type (ToBeEnciphered)
VALUENOTATION ::= value (VALUE BIT STRING
     -- el valor de la cadena de bits se genera
     -- tomando dos octetos que forman la codificación completa
     -- (utilizando las Reglas de Codificación Básicas NSA.1)
     -- del valor del tipo ToBeEnciphered y aplicando
     -- un procedimiento de cifrado a esos octetos --

END

SIGNED MACRO    ::=
BEGIN
TYPE NOTATION   ::= type (ToBeSigned)
VALUE NOTATION ::= value(VALUE

SEQUENCE{
     ToBeSigned,
     AlgorithmIdentifier, -- del algoritmo utilizado para generar la firma
     ENCRYPTED OCTET STRING
          -- donde la cadena de octetos es el resultado
          -- de aplicar la función al valor de
          -- "ToBeSigned" --}
                         )
END -- of SIGNED

SIGNATURE MACRO ::=
BEGIN

TYPE NOTATION      ::= type (OfSignature)
VALUE NOTATION   ::= value(VALUE

     SEQUENCE{
          AlgorithmIdentifier,
          -- del algoritmo utilizado para computar la firma
          ENCRYPTED OCTET STRING
          -- donde la cadena de octetos es una función, (por ejemplo, una versión comprimida
          o hasheada)
          -- del valor "OfSignature", que puede incluir el identificador del
          -- algoritmo utilizado para computar la firma --}
                                                  )
END -- of SIGNATURE

PROTECTED MACRO ::= SIGNATURE

END -- de Definiciones del Marco de Autenticación
```

ANEXO H

(a la Recomendación X.509)

## Definición de referencia de los identificadores de objeto para algoritmo

Este anexo no forma parte integrante de la Recomendación.

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 NSA.1, **AlgorithmObjectIdentifiers**.

```
AlgorithmObjectIdentifiers      {joint-iso-ccitt ds(5) modules(1)
                                    algorithmObjectIdentifiers(8)}

DEFINITIONS ::=
BEGIN

EXPORTS
    encryptionAlgorithm, hashAlgorithm, signatureAlgorithm,
    rsa,squareMod-n,sqMod-nWithRSA;

IMPORTS
    algorithm,authenticationFramework
        FROM UsefulDefinitions  {joint-iso-ccitt ds(5)modules(1)
                                 usefulDefinitions(0)}

    ALGORITHM  FROM AuthenticationFramework authenticationFramework;

-- categorías de identificador de objeto

encryptionAlgorithm OBJECT IDENTIFIER      ::=   {algorithm 1}

hashAlgorithm OBJECT IDENTIFIER            ::=   {algorithm 2}

signatureAlgorithm OBJECT IDENTIFIER       ::=   {algorithm 3}

-- algoritmos

rsa   ALGORITHM
      PARAMETER KeySize
      ::=   {encryptionAlgorithm 1}

KeySize ::=      INTEGER

sqMod-n ALGORITHM
      PARAMETER BlockSize
      ::=   {hashAlgorithm 1}

BlockSize ::=    INTEGER

sqMod-nWithRSA ALGORITHM
      PARAMETER KeyAndBlockSize
      ::=   {signatureAlgorithm 1}

KeyAndBlockSize ::= INTEGER

END -- de definiciones de identificadores de objeto para algoritmos
```

## SERIES DE RECOMENDACIONES DEL UIT-T

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

Impreso en Suiza
Ginebra, 2008