UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T H.235

(11/2000)

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

SERIE H: SISTEMAS AUDIOVISUALES Y MULTIMEDIOS

Infraestructura de los servicios audiovisuales – Aspectos de los sistemas

Seguridad y criptado para terminales multimedios de la serie H (basados en las Recomendaciones H.323 y H.245)

Recomendación UIT-T H.235

(Anteriormente Recomendación del CCITT)

RECOMENDACIONES UIT-T DE LA SERIE H

SISTEMAS AUDIOVISUALES Y MULTIMEDIOS

CARACTERÍSTICAS DE LOS SISTEMAS VIDEOTELEFÓNICOS	H.100–H.199
INFRAESTRUCTURA DE LOS SERVICIOS AUDIOVISUALES	
Generalidades	H.200–H.219
Multiplexación y sincronización en transmisión	H.220–H.229
Aspectos de los sistemas	**H.230–H.239**
Procedimientos de comunicación	H.240–H.259
Codificación de imágenes vídeo en movimiento	H.260–H.279
Aspectos relacionados con los sistemas	H.280–H.299
SISTEMAS Y EQUIPOS TERMINALES PARA LOS SERVICIOS AUDIOVISUALES	H.300–H.399
SERVICIOS SUPLEMENTARIOS PARA MULTIMEDIOS	H.450–H.499

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

Recomendación UIT-T H.235

Seguridad y criptado para terminales multimedios de la serie H (basados en las Recomendaciones H.323 y H.245)

Resumen

La presente Recomendación describe mejoras dentro del marco de las especificaciones de las Recomendaciones de la serie H.3xx para incorporar servicios de seguridad tales como *autenticación* y *privacidad* (criptado de datos). El esquema propuesto es aplicable a conferencias punto a punto y multipunto para cualesquiera terminales que utilicen UIT-T H.245 como su protocolo de control.

Por ejemplo, los sistemas H.323 funcionan por redes de paquetes que no proporcionan una calidad de servicio garantizada. Por la misma razón técnica de que la red de base no proporciona la calidad de servicio, la red no proporciona un servicio seguro. La comunicación segura en tiempo real por redes inseguras plantea generalmente dos problemas importantes: *autenticación* y *privacidad*.

La presente Recomendación describe la infraestructura de seguridad y técnicas de privacidad específicas que han de emplear los terminales multimedios de la serie H.3xx. Esta Recomendación aborda los aspectos relacionados con la conferencia interactiva, entre los que cabe citar la autenticación y privacidad de todos los trenes de medios en tiempo real que son intercambiados en la conferencia, aunque no está limitado estrictamente a éstos. La presente Recomendación proporciona el protocolo y algoritmos necesarios entre las entidades H.323.

La presente Recomendación utiliza las facilidades generales soportadas en UIT-T H.245 y como tal, cualquier norma que funcione junto con este protocolo de control puede utilizar este marco de seguridad. Se prevé que siempre que sea posible otros terminales de la serie H puedan interfuncionar y utilizar directamente los métodos descritos en esta Recomendación, en el que inicialmente no se prevé la implementación completa en todos los campos, sino que destacará específicamente la autenticación de puntos extremos y la privacidad de los medios.

La presente Recomendación incluye la capacidad de negociar servicios y funcionalidades de una manera genérica, y la selectividad en relación con técnicas criptográficas y capacidades utilizadas. La manera específica en que éstas se utilizan se relaciona con las capacidades de los sistemas, requisitos de aplicación y restricciones específicas de la política de seguridad. La presente Recomendación soporta diversos algoritmos criptográficos, con opciones variadas apropiadas para diferentes fines, por ejemplo, longitudes de claves. Ciertos algoritmos criptográficos pueden ser asignados a servicios de seguridad específicos (por ejemplo, uno para criptación rápida de tren de medios y otro para criptación de señalización).

Cabe señalar también que algunos algoritmos criptográficos o mecanismos pueden estar reservados para exportación u otros aspectos nacionales (por ejemplo, con longitudes de claves restringidas). La presente Recomendación soporta la señalización de algoritmos bien conocidos además de la señalización de algoritmos criptográficos no normalizados o privados. No hay algoritmos específicamente obligatorios, aunque se aconseja decididamente que los puntos extremos soportan el mayor número posible de algoritmos para lograr el interfuncionamiento. Esto es paralelo al concepto de que el soporte del protocolo H.245 no garantiza el interfuncionamiento entre códecs de dos entidades.

Esta versión de UIT-T H.235 sustituye a la versión 1 presentando varias mejoras, tales como la criptografía de curva elíptica, los perfiles de seguridad (el simple basado en contraseñas y el perfeccionado basado en firmas digitales), las nuevas contramedidas de seguridad (antiinundación de medios), el soporte del algoritmo de criptación avanzado (AES), el soporte para el servicio fuera del terminal, los identificadores de objeto definidos y los cambios incorporados de la guía del implementador de H.323.

Orígenes

La Recomendación UIT-T H.235, revisada por la Comisión de Estudio 16 (2001-2004) del UIT-T, fue aprobada por el procedimiento de la Resolución 1 de la AMNT el 17 de noviembre de 2000.

La primera versión de UIT-T H.235 fue aprobada por la Comisión de Estudio 16 del UIT-T el 6 de febrero de 1998.

Palabras clave

Autenticación, certificado, criptación, firma digital, gestión de claves, integridad, perfil de seguridad, seguridad de multimedios.

PREFACIO

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

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

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

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

NOTA

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

PROPIEDAD INTELECTUAL

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

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

ÍNDICE

		Página
1	Alcance	1
2	Referencias normativas	2
3	Términos y definiciones	3
4	Símbolos y abreviaturas	5
5	Convenios	5
6	Presentación del sistema	6
6.1	Resumen	6
6.2	Autenticación	6
	6.2.1 Certificados	7
6.3	Seguridad de establecimiento de la comunicación	7
6.4	Seguridad de control de la llamada (H.245)	7
6.5	Privacidad de trenes de medios	7
6.6	Elementos de confianza	8
	6.6.1 Depósito de claves	8
6.7	No repudio	9
7	Procedimientos de establecimiento de la conexión	9
7.1	Introducción	9
8	Señalización y procedimientos H.245	9
8.1	Funcionamiento seguro del canal H.245	9
8.2	Funcionamiento inseguro del canal H.245	9
8.3	Intercambio de capacidades	10
8.4	Cometido de terminal director	10
8.5	Señalización de canal lógico	10
9	Procedimientos multipunto	11
9.1	Autenticación	11
9.2	Privacidad	11
10	Señalización y procedimientos de autenticación	11
10.1	Introducción	11
10.2	Intercambio Diffie-Hellman con autenticación facultativa	11
10.3	Autenticación basada en abono	12
	10.3.1 Introducción	12
	10.3.2 Contraseña con criptación simétrica	13
	10.3.3 Contraseña con troceado	14

Página

10.3.4 Certificado con firma 15
10.3.5 Utilización de contraseñas y secreto compartido 16

11 Procedimiento de criptación de tren de medios 17
11.1 Claves de sesión de medios 18
11.2 Antiinundación de medios 19
11.2.1 Lista de identificadores de objeto 21

12 Recuperación tras error de seguridad 21

13 Autenticación asimétrica e intercambio de claves utilizando sistemas criptográficos de curva elíptica 22
13.1 Gestión de claves 22
13.2 Firma digital 23

Anexo A – ASN.1 del protocolo H.235 23

Anexo B – Aspectos específicos de H.323 27
B.1 Antecedentes 27
B.2 Señalización y procedimientos 28
B.2.1 Compatibilidad con la revisión 1 28
B.3 Aspectos relativos a RTP/RTCP 29
B.4 Señalización RAS/procedimientos de autenticación 30
B.4.1 Introducción 30
B.4.2 Autenticación de punto extremo – controlador de acceso (no basada en abono) 30
B.4.3 Autenticación de punto extremo – controlador de acceso (basada en abono) 31
B.5 Interacciones no relacionadas con terminales 33
B.5.1 Pasarela 33

Anexo C – Aspectos específicos del protocolo H.324 33

Anexo D – Perfil de seguridad básico 33
D.1 Introducción 33
D.2 Convenios de especificación 34
D.3 Alcance 35
D.4 Abreviaturas 35
D.5 Referencias normativas 36
D.6 Perfil de seguridad básico 37
D.6.1 Visión general 37
D.6.2 Autenticación e integridad 40

Página

D.6.3 Requisitos H.323 40
D.6.4 Escenario con encaminamiento directo 47
D.6.5 Soporte de los servicios fuera del terminal 47
D.6.6 Compatibilidad con H.235 Versión 1 47
D.6.7 Comportamiento multidifusión 48
D.7 Perfil de seguridad de criptación vocal 48
D.7.1 Gestión de claves 48
D.7.2 Actualización de claves y sincronización 50
D.7.3 DES triple en modo CBC exterior 51
D.8 Interceptación legal 51
D.9 Lista de mensajes de señalización asegurados 51
D.9.1 RAS H.225.0 52
D.9.2 Señalización de llamada H.225.0 52
D.9.3 Control de llamada H.245 52
D.10 Utilización de sendersID y de generalID 52
D.11 Lista de identificadores de objeto 53
D.12 Bibliografía 54

Anexo E – Perfil de firmas 55
E.1 Visión general 55
E.2 Convenios acerca de las especificaciones 56
E.3 Requisitos H.323 58
E.4 Servicios de seguridad 59
E.5 Detalles de las firmas digitales con parejas de claves privada/clave pública (Procedimiento II) 60
E.6 Procedimientos para la conferencia multipunto 61
E.7 Autenticación de extremo a extremo (Procedimiento III) 61
E.8 Autenticación solamente 63
E.9 Autenticación e integridad 64
E.10 Cálculo de la firma digital 65
E.11 Verificación de la firma digital 65
E.12 Tratamiento de los certificados 65
E.13 Ilustración del empleo del procedimiento II 66
E.13.1 Autenticación, integridad y no repudio de mensajes RAS 66
E.13.2 Autenticación solamente de mensajes RAS 67
E.13.3 Autenticación, integridad y no repudio de mensaje H.225.0 68
E.13.4 Autenticación e integridad de los mensajes H.245 69

Página

E.14 Compatibilidad con la Versión 1 de H.235 69
E.15 Comportamiento multidifusión 69
E.16 Lista de mensajes de señalización seguros 70
E.16.1 RAS H.225 70
E.16.2 Señalización de llamada H.225.0 70
E.17 Utilización de sendersID y generalID 70
E.18 Lista de identificadores de objeto 72

Apéndice I – Detalles de las implementaciones H.323 73
I.1 Métodos de relleno de texto cifrado 73
I.2 Nuevas claves 75
I.3 Elementos de confianza H.323 75
I.4 Ejemplos de implementaciones 75
I.4.1 Testigos 75
I.4.2 Utilización de testigos en los sistemas H.323 76
I.4.3 Utilización del valor aleatorio H.235 en sistemas H.323 77
I.4.4 Contraseña 77
I.4.5 IPSEC 78
I.4.6 Soporte de servicios fuera del terminal 79

Apéndice II – Detalles de implementaciones del protocolo H.324 81

Apéndice III – Otros detalles de implementaciones de la serie H 81

Apéndice IV – Bibliografía 81

Recomendación UIT-T H.235

Seguridad y criptado para terminales multimedios de la serie H (basados en las Recomendaciones H.323 y H.245)

1 Alcance

La finalidad primaria de la presente Recomendación es proporcionar la autenticación, privacidad e integridad dentro del marco de los protocolos vigentes de la serie H. El texto actual de esta Recomendación (2000) proporciona detalles sobre la implementación con UIT-T H.323. Se prevé que este marco funcione junto con otros protocolos de la serie H que utilizan el protocolo H.245 como su protocolo de control.

Entre los objetivos adicionales de esta Recomendación cabe citar:

1) La arquitectura de seguridad se debe desarrollar como un marco extensible y flexible para aplicar un sistema de seguridad para los terminales de la serie H. Esto se debe proporcionar mediante servicios flexibles e independientes y la funcionalidad que éstos suministran, e incluye la posibilidad de negociar y seleccionar las técnicas criptográficas empleadas, así como la manera en la cual éstas se utilizan.

2) Proporcionar seguridad para todas las comunicaciones establecidas como resultado de la aplicación de los protocolos H.3xx. Esto incluye los aspectos relativos al establecimiento de la conexión, control de la llamada e intercambio de medios entre todas las entidades. Este requisito comprende la utilización de comunicación confidencial (privacidad) y puede explotar funciones para autenticación de pares así como protección del entorno del usuario contra ataques.

3) La presente Recomendación no excluye la integración de otras funciones de seguridad en entidades H.3xx que puedan protegerlas contra ataques de la red.

4) La presente Recomendación no debe limitar la posibilidad de ampliar según proceda cualesquiera especificaciones de la Recomendación de la serie H.3xx. Esto puede incluir el número de usuarios asegurados y los niveles de seguridad proporcionados.

5) Cuando proceda, todos los mecanismos y facilidades deben ser proporcionados independientemente de cualquier transporte o topologías subyacentes. Para contrarrestar estas amenazas se pueden necesitar otros medios que están fuera del ámbito de la presente Recomendación.

6) Se prevé el funcionamiento en un entorno mixto (entidades seguras e inseguras).

7) La presente Recomendación debe proporcionar facilidades para distribuir claves de sesión asociadas con la criptografía utilizada. (Esto no supone que la gestión de certificados basada en claves públicas deba ser parte de la presente Recomendación.)

8) La presente Recomendación proporciona dos perfiles de seguridad que facilitan el interfuncionamiento. En el anexo D se describe un perfil de seguridad sencillo basado todavía en contraseñas seguras, mientras que en el anexo E se presenta un perfil de seguridad de firmas que despliega firmas digitales, certificados y una infraestructura de claves públicas que superan las limitaciones del anexo D.

La arquitectura de seguridad, descrita en la presente Recomendación, no supone que los participantes están familiarizados entre sí. Sin embargo, supone que se han tomado precauciones adecuadas para asegurar físicamente los puntos extremos de la serie H. Por consiguiente, se considera que la principal amenaza a la seguridad de las comunicaciones es la escucha furtiva en la red o algún otro método de desviar los trenes de medios.

La Recomendación H.323 proporciona los medios para conducir una conferencia de audio, vídeo y datos entre dos o más partes, pero no proporciona el mecanismo para que cada participante pueda autenticar la identidad de los otros participantes, ni proporciona los medios para que las comunicaciones sean privadas (es decir, criptado de los trenes).

Las Recomendaciones UIT-T H.323, H.324 y H.310 utilizan los procedimientos de señalización de canal lógico de UIT-T H.245, en los cuales se describe el contenido de cada canal lógico cuando se abre el canal. Se proporcionan procedimientos para indicar las capacidades del receptor y del transmisor, las transmisiones están limitadas a los receptores que pueden decodificar, y los receptores pueden pedir a los transmisores un modo deseado. Las capacidades de seguridad de cada punto extremo son indicadas de la misma manera que cualquier otra capacidad de comunicación.

Algunos terminales de la serie H (H.323) pueden ser utilizados en configuraciones multipunto. El mecanismo de seguridad descrito en esta Recomendación permitirá el funcionamiento seguro en estos entornos, incluido el funcionamiento de unidades de control multipunto (MCU) centralizadas y descentralizadas.

2 Referencias normativas

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes.

– UIT-T H.225.0 (2000*), Protocolos de señalización de llamadas y paquetización de trenes de medios para sistemas de comunicaciones multimedios basadas en paquetes*.

– UIT-T H.235 (1998), *Seguridad y criptado para terminales multimedios de la serie H (basados en las Recomendaciones H.323 y H.245)*.

– UIT-T H.245 (2000), *Protocolo de control para comunicación multimedios*.

– UIT-T H.323 (2000), *Sistemas de comunicación multimedios basados en paquetes*.

– UIT-T H.323 Anexo J (2000), *Seguridad para H.323 anexo F*.

– UIT-T Q.931 (1998), *Especificación de la capa 3 de la interfaz usuario-red de la red digital de servicios integrados para el control de la llamada básica*.

– UIT-T X.509 (2000) | ISO/CEI 9594-8:2001, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Marcos para certificados de claves públicas y de atributos.*

– UIT-T X.800 (1991), *Arquitectura de seguridad de la interconexión de sistemas abiertos para aplicaciones del CCITT*.

  ISO 7498-2:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture*.

– UIT-T X.803 (1994) | ISO/CEI 10745:1995, *Tecnología de la información – Interconexión de sistemas abiertos – Modelo de seguridad de capas superiores*.

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

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

– ISO/CEI 9797:1994, *Information technology – Security techniques – Data integrity mechanism using a cryptographic check function employing a block cipher algorithm*.

– ISO/CEI 9798-2:1999, *Information technology – Security techniques – Entity authentication – Part 2: Mechanisms using symmetric encipherment algorithms*.

– ISO/CEI 9798-3:1998, *Information technology – Security techniques – Entity authentication mechanisms – Part 3: Mechanism using digital signature techniques*.

– ISO/CEI 9798-4:1999, *Information technology – Security techniques – Entity authentication – Part 4: Mechanisms using a cryptographic check function*.

– ISO/CEI FCD 15946-1, *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 1: General*.

– ISO/CEI FCD 15946-2, *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 2: Digital signatures*.

– ATM Forum: af-sec-0100.002 (2001), *ATM Security Specification Version 1.*

– IETF RFC 1321 (1992), *The MD5 Message-Digest Algorithm*.

– IETF RFC 2104 (1997), *HMAC: Keyed-Hashing for Message Authentication*.

– IETF RFC 2138 (1997), *Remote Authentication Dial In User Service (RADIUS)*.

– IETF RFC 2246 (1999), *The TLS Protocol Version 1.0*.

– IETF RFC 2401 (1998), *Security Architecture for the Internet Protocol*.

– IETF RFC 2402 (1998), *IP Authentication Header*.

– IETF RFC 2407 (1998), *The Internet IP Security Domain of Interpretation for ISAKMP*.

– IETF RFC 2412 (1998), *The OAKLEY Key Determination Protocol*.

– IETF RFC 2437 (1998), *PKCS #1: RSA Encryption Version 2.0*.

– IETF RFC 2459 (1999), *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*.

3 Términos y definiciones

A los efectos de la presente Recomendación se aplican las definiciones que figuran en la cláusula 3/H.323, cláusula 3/H.225.0 y cláusula 3/H.245 junto con las de esta cláusula. Algunos de los siguientes términos se utilizan como se define en UIT-T X.800 | ISO 7498-2 y UIT-T X.803, UIT-T X.810 y UIT-T X.811.

3.1 control de acceso: Prevención del uso no autorizado de un recurso, incluida la prevención del uso de un recurso de una manera no autorizada (X.800).

3.2 autenticación: Provisión de seguridad de la identidad alegada de una entidad (X.811).

3.3 autorización: Concesión de permisos sobre la base de identificación autenticada.

3.4 ataque: Actividades realizadas para anular los mecanismos de seguridad de un sistema o aprovechar sus deficiencias. Los ataques directos a un sistema aprovechan las deficiencias en los algoritmos, principios o propiedades subyacentes de un mecanismo de seguridad. Los ataques indirectos anulan el mecanismo, o hacen que el sistema utilice el mecanismo incorrectamente.

3.5 certificado: Conjunto de datos relativos a la seguridad emitidos por una autoridad de seguridad o tercero de confianza, junto con información de seguridad que se utiliza para proporcionar los servicios de integridad y autenticación de origen de datos para los datos (X.810). En la presente Recomendación el término se relaciona con certificados de "clave pública" que son valores que representan una clave pública patentada (y otra información facultativa) verificada y firmada por una autoridad de confianza en un formato infalsificable.

3.6 cifra: Algoritmo criptográfico, una transformación matemática.

3.7 confidencialidad: Propiedad que impide la revelación de información a individuos, entidades o procesos no autorizados.

3.8 algoritmo criptográfico: Función matemática que calcula un resultado a partir de uno o varios valores de entrada.

3.9 cifrado: Cifrado (criptación) es el proceso que hace que los datos sean ilegibles para entidades no autorizadas aplicando un algoritmo criptográfico (un algoritmo de criptación). El descifrado (descriptación) es la operación inversa por la cual el texto cifrado se transforma en texto claro.

3.10 integridad: Propiedad de que los datos no han sido alterados de una manera no autorizada.

3.11 gestión de claves: Generación, almacenamiento, distribución, supresión, archivo y aplicación de claves de acuerdo con una política de seguridad (X.800).

3.12 tren de medios: Un tren de medios puede ser del tipo audio, vídeo o datos, o una combinación de cualquiera de ellos. Los datos de trenes de medios transportan datos de usuario o de aplicación (cabida útil) pero no datos de control.

3.13 no repudio: Protección contra la negación por una de las entidades que participan en una comunicación de haber participado en toda la comunicación o parte de ésta.

3.14 privacidad: Modo de comunicación en el cual sólo las partes habilitadas explícitamente pueden interpretar la comunicación. Esto se logra en general mediante criptación y claves compartidas para la cifra.

3.15 canal privado: Para la presente Recomendación, un canal privado es el resultante de negociación previa por un canal seguro. En este contexto, puede ser utilizado para manipular trenes de medios.

3.16 criptografía de claves públicas: Sistema de criptación que utiliza claves asimétricas (para criptación/descriptación) en el cual las claves tienen una relación matemática entre sí, que no puede ser calculada razonablemente.

3.17 perfil de seguridad: Conjunto (subconjunto) de características y procedimientos coherentes y con capacidad de interfuncionamiento entre sí que caen fuera del alcance de UIT-T H.235 y que son útiles para asegurar las comunicaciones multimedios H.323 entre las entidades involucradas en un escenario específico.

3.18 inundación: Ataque de denegación de servicio que tiene lugar cuando se envían en exceso a un sistema datos no autorizados. Un caso especial es la inundación de medios que se produce cuando se envían paquetes RTP en puertos UDP. Normalmente el sistema es inundado con paquetes; su procesamiento consume recursos preciosos del sistema.

3.19 algoritmo criptográfico simétrico (**basado en claves secretas**): Un algoritmo para realizar el cifrado o el algoritmo correspondiente para realizar el descifrado en el cual se requiere la misma clave para ambas operaciones (X.810).

3.20 amenaza: Violación potencial de la seguridad (X.800).

4 Símbolos y abreviaturas

En esta Recomendación se utilizan las siguientes siglas.

CRL	Lista de revocación de certificados (*certificate revocation list*)
DSS	Norma sobre firmas digitales (*digital signature standard*)
ECC y EC	Criptosistema de curva elíptica (véase la sección 8.7 *ATM Forum Security Specification Versión 1.1*). Un criptosistema de claves públicas (*elliptic curve cryptosystem. A public-key cryptosystem*)
EC-GDSA	Firma digital de curva elíptica con apéndice análoga al algoritmo de firma digital NIST (DSA) [*elliptic curve digital signature with appendix analog of the NIST digital signature algorithm (DSA)*]; (véase también [ISO/CEI 15946-2, capítulo 5]).
ECKAS-DH	Esquema de convenio de claves de curva elíptica – Diffie-Hellman – El esquema de convenio de claves Diffie-Hellman que utiliza criptografía de curva elíptica (*elliptic curve key agreement scheme – Diffie-Hellman. The Diffie-Hellman key agreement scheme using elliptic curve cryptography*)
IPSEC	Seguridad de protocolo Internet (*Internet protocol security*)
QOS	Calidad de servicio (*quality of service*)
RSA	Rivest, Shamir y Adleman (algoritmo de clave pública)
SDU	Unidad de datos de servicio (*service data unit*)
TLS	Seguridad de nivel de transporte (*transport level security*)

5 Convenios

En la presente Recomendación se utilizan los siguientes convenios:

– El tiempo futuro indica un requisito obligatorio.

– El condicional "debería" indica una acción aconsejada pero facultativa.

– La palabra "puede" indica una acción facultativa, en vez de una recomendación de que se haga algo.

Las referencias a cláusulas, subcláusulas, anexos y apéndices se relacionan con puntos de la presente Recomendación, a menos que se indique explícitamente otra Recomendación. Por ejemplo, "1.4" hace referencia a la cláusula 1.4 de la presente Recomendación; "6.4/H.245" hace referencia a la cláusula 6.4 de la Recomendación H.245.

La presente Recomendación describe el uso de "n" tipos de mensajes diferentes: H.245, RAS, Q.931, etc. Para distinguir entre los diferentes tipos de mensajes, se sigue el siguiente convenio: los nombres de mensajes y parámetros H.245 están formados por varias palabras que se representan en el tipo de letra negritas [**maximumDelayJitter** (**fluctuación de retardo de fase máxima**)]; los nombres de mensajes RAS se representan con abreviaturas de tres letras (**ARQ**); los nombres de mensajes Q.931 están formados por una o dos palabras cuyas letras iniciales apareces en mayúsculas [**Call Proceeding** (**Llamada en curso**)].

6 Presentación del sistema

6.1 Resumen

1) El canal de señalización de llamada se puede asegurar utilizando TLS [TLS] o IPSEC [IPSEC] en un puerto conocido seguro (H.225.0).
2) Los usuarios pueden ser autenticados durante la conexión de llamada inicial, en el proceso de asegurar el canal H.245 y/o intercambiando certificados por el canal H.245.
3) Las capacidades de criptación de un canal de medios son determinadas por extensiones del mecanismo de negociación de capacidades existente.
4) La distribución inicial de material de claves del terminal director se efectúa mediante mensajes **OpenLogicalChannel** (**Apertura canal lógico**) u **OpenLogicalChannelAck** (**Acuse apertura canal lógico**).
5) El recifrado se puede realizar mediante las instrucciones H.245: **EncryptionUpdateRequest** (**Petición actualización criptación**) y **EncryptionUpdate** (**Actualización criptación**).
6) La distribución de material de claves se protege haciendo funcionar el canal H.245 como un canal privado o protegiendo específicamente el material de claves mediante el uso de certificados intercambiados seleccionados.
7) Los protocolos de seguridad presentados se conforman con las normas publicadas de la ISO o con las normas propuestas de IETF.

6.2 Autenticación

El proceso de autenticación verifica que los respondedores son, de hecho, quienes dicen ser. La autenticación se puede realizar junto con el intercambio de certificados basados en claves públicas. Se puede efectuar también por un intercambio que utiliza un secreto compartido entre las entidades participantes. Éste puede ser una contraseña estática o alguna otra pieza previa de información.

La presente Recomendación describe el protocolo para intercambiar los certificados, pero no especifica los criterios por los cuales éstos son verificados y aceptados mutuamente. En general, los certificados dan cierta seguridad al verificador de que el presentador del certificado es quien dice ser. La intención del intercambio de certificados es autenticar al *usuario* del punto extremo, no simplemente al punto extremo físico. Cuando se utilizan certificados digitales, un protocolo de autenticación prueba que los respondedores poseen las claves privadas correspondientes a las claves públicas contenidas en los certificados. Esta autenticación protege contra ataques intermedios, pero no prueba automáticamente quiénes son los respondedores. Para esto se requiere normalmente que haya alguna política relativa a otro contenido de los certificados. Por ejemplo, para los certificados de autorización, el certificado contendría normalmente la identificación del proveedor de servicio junto con alguna forma de identificación de cuenta de usuario prescrita por el proveedor de servicio.

El marco de autenticación de la presente Recomendación no prescribe el contenido de los certificados (es decir, no especifica una política de certificado) además de lo requerido por el protocolo de autenticación. Sin embargo, una aplicación que utiliza este marco puede imponer requisitos de política de alto nivel, tales como presentar el certificado al usuario para aprobación. Esta política de alto nivel puede ser automatizada dentro de la aplicación o requerir la interacción humana.

Para la autenticación que no utiliza certificados digitales, la presente Recomendación proporciona la señalización para completar distintos casos de pregunta/respuesta. Este método de autenticación requiere la coordinación previa por las entidades comunicantes de modo que se pueda obtener un secreto compartido. Un ejemplo de este método sería un cliente de un servicio basado en abono.

Como una tercera opción, la autenticación puede ser completada dentro del contexto de un protocolo de seguridad distinto, tal como TLS [TLS] o IPSEC [IPSEC].

La autenticación bidireccional y unidireccional puede ser soportada por entidades pares. Esta autenticación se puede producir en algunos o en todos los canales de comunicación.

Todos los mecanismos de autenticación específicos descritos en la presente Recomendación son idénticos a los algoritmos desarrollados por la ISO, o derivados de éstos, como se especifica en las Partes 2 a 3 de ISO/CEI 9798, o están basados en protocolos IETF.

6.2.1 Certificados

La normalización de certificados, incluida su generación, administración y distribución, está fuera del alcance de la presente Recomendación. Los certificados utilizados para establecer canales seguros (señalización de llamada y/o control de llamada) se conformarán a los prescritos por cualquier protocolo que haya sido negociado para asegurar el canal.

Cabe señalar que para la autenticación que utiliza certificados de clave pública, los puntos extremos tienen que proporcionar firmas digitales utilizando el valor de clave privada asociado. El intercambio de certificados de clave pública por sí solo no protege contra ataques intermedios. Los protocolos H.235 cumplen este requisito.

6.3 Seguridad de establecimiento de la comunicación

Hay por lo menos dos razones para motivar la seguridad del canal de establecimiento de la comunicación (por ejemplo, H.323 que utiliza Q.931). La primera es la autenticación simple, antes de aceptar la llamada. La segunda razón es tener en cuenta la autorización de la llamada. Si esta funcionalidad se desea en el terminal de la serie H, se debe utilizar un modo seguro de comunicación (tal como TLS/IPSEC para H.323) antes del intercambio de mensajes de conexión de la llamada. Como otra posibilidad, la autorización se puede proporcionar sobre la base de una autenticación específica del servicio. Las constricciones de una política de autorización específica del servicio están fuera del alcance de la presente Recomendación.

6.4 Seguridad de control de la llamada (H.245)

El canal de control de llamada (H.245) debe estar asegurado también de alguna manera para proporcionar privacidad de los medios subsiguientes. El canal H.245 se asegurará utilizando cualquier mecanismo de privacidad negociado (esto incluye la opción "ninguno"). Los mensajes H.245 se utilizan para señalizar algoritmos de criptación y claves de criptación utilizados en los canales de medios privados compartidos. La capacidad de hacer esto, canal lógico por canal lógico, permite que diferentes canales de medios sean criptados por diferentes mecanismos. Por ejemplo, en conferencias multipunto centralizadas, es posible utilizar diferentes claves para los trenes a cada punto extremo. Esto puede permitir que los trenes de medios sean privados para cada punto extremo en la conferencia. Para utilizar los mensajes H.245 de una manera segura, todo el canal H.245 (canal lógico 0) se debe abrir de una manera segura negociada.

El mecanismo por el cual el canal H.245 es seguro depende de los terminales de la serie H participantes. El único requisito en todos los sistemas que utilizan esta estructura de seguridad es que cada uno tenga alguna manera de negociar y/o señalizar que el canal H.245 ha de funcionar de una manera particularmente segura antes de que sea iniciado realmente. Por ejemplo, H.323 utilizará los mensajes de señalización de conexión H.225.0 para realizar esto.

6.5 Privacidad de trenes de medios

La presente Recomendación describe la privacidad de medios para trenes de medios enviados por transportes basados en paquetes. Estos canales pueden ser unidireccionales con respecto a las caracterizaciones de canal lógico H.245. Los canales no tienen que ser unidireccionales en un nivel físico o de transporte.

Un primer paso para obtener la privacidad de los medios debe ser la provisión de un canal de control privado por el cual establecer material de claves criptográficas y/o establecer los canales lógicos que transportarán los trenes de medios criptados. Para esto, cuando se funciona en una conferencia segura, cualesquiera puntos extremos participantes pueden utilizar un canal H.245 criptado. De esta manera, la selección del algoritmo criptográfico y las claves de criptación transferidas en la instrucción **OpenLogicalChannel** H.245 están protegidas.

El canal seguro H.245 puede funcionar con diferentes características de los canales de medios privados mientras proporcione un nivel de privacidad mutuamente aceptable. Esto prevé mecanismos que protegen los trenes de medios y los canales de control para funcionar de una manera completamente independiente, proporcionando niveles totalmente diferentes de robustez y complejidad.

Si se requiere que el canal H.245 funcione de una manera no criptada, las claves de criptación de medios específicos pueden ser criptadas separadamente de la manera señalizada y acordadas por las partes participantes. Se puede utilizar un canal lógico del tipo **h235Control** (**Control h235**) para proporcionar el material que ha de proteger las claves de criptación de medios. Este canal lógico puede funcionar en un modo negociado adecuadamente.

La privacidad (criptación) de los datos transportados por canales lógicos tendrá la forma especificada por la **OpenLogicalChannel**. La información de encabezamiento específica de transporte no será criptada. La privacidad de datos se ha de basar en la criptación de extremo a extremo.

6.6 Elementos de confianza

La base para la autenticación (confianza) y la privacidad es definida por los terminales del canal de comunicación. Para un canal de establecimiento de conexión, ésta puede estar entre el llamante y un componente de la red anfitriona. Por ejemplo, un teléfono "confía" en que el conmutador de red lo conectará con el teléfono cuyo número ha marcado. Por este motivo, toda entidad que termina un canal de control H.245 criptada o cualesquiera canales lógicos del tipo **encryptedData** (**Datos criptados**) serán considerados un elemento de confianza de la conexión; esto incluye las unidades de control multipunto y las pasarelas. El resultado de confiar en un elemento es la confianza para revelar el mecanismo de privacidad (algoritmo y clave) a ese elemento.

Dado lo anterior, corresponde a los participantes en el trayecto de comunicación autenticar cualquiera y todos los elementos "de confianza". Esto se hará normalmente mediante el intercambio de certificados como se haría para la autenticación de extremo a extremo "normalizada". La presente Recomendación no requiere ningún nivel específico de autenticación, sino que aconseja que dicho nivel sea aceptable para todas las entidades que utilizan el elemento de confianza. Los detalles de un modelo de confianza y de una política de certificados quedan en estudio.

La privacidad se puede asegurar entre dos puntos extremos solamente si las conexiones entre elementos de confianza han demostrado estar protegidas contra ataques intermedios.

6.6.1 Depósito de claves

Aunque no se requiere específicamente para el funcionamiento, la presente Recomendación contiene disposiciones para que las entidades que utilizan el protocolo H.235 soporten la facilidad conocida como tercera parte confiable (TTP, *trusted third party*) dentro de los elementos de señalización.

Se debe soportar la posibilidad de recuperar las claves de criptación de medios perdidas en aquellas instalaciones en las que esta funcionalidad es deseada o requerida.

El depósito de claves es una facilidad a menudo denominada tercera parte confiable (TTP). Esta facilidad queda en estudio.

6.7 No repudio

Queda en estudio.

7 Procedimientos de establecimiento de la conexión

7.1 Introducción

Como se indica en la introducción del sistema, el canal de conexión de la llamada (H.225.0 para la serie H.323) y el canal de control de llamada (H.245) funcionarán en el modo seguro o inseguro negociado a partir del primer intercambio. Para el canal de conexión de la llamada, esto se hace previamente [para H.323 un TSAP seguro de TLS (puerto 1300) será utilizado para los mensajes Q.931]. Para el canal de control de llamada, el modo de seguridad es determinado por la información transferida en el protocolo de establecimiento de conexión inicial en uso por el terminal de la serie H.

Cuando no hay capacidades de seguridad superpuestas, el terminal llamado puede rechazar la conexión. El error devuelto no debe transferir información sobre cualquier discordancia de seguridad y el terminal llamante tendrá que determinar el problema por otros medios. Cuando el terminal llamante recibe un mensaje de ACUSE DE CONEXIÓN sin capacidades de seguridad suficientes, terminará la llamada.

Si los terminales llamante y llamado tienen capacidades de seguridad compatibles, ambos lados supondrán que el canal H.245 funcionará en el modo seguro negociado. La imposibilidad de establecer el canal H.245 en el modo seguro determinado debe considerarse un error de protocolo y la conexión será terminada.

8 Señalización y procedimientos H.245

En general, los aspectos de privacidad de los canales de medio son controlados de la misma manera que cualquier otro parámetro de codificación; cada terminal indica sus capacidades, la fuente de los datos selecciona un formato que ha de utilizar y el receptor acepta o rechaza el modo. Todos los aspectos del mecanismo independientes del transporte, tales como selección de algoritmo, se indican en elementos de canal lógico genéricos. Los elementos específicos de transporte, tales como la sincronización de algoritmos de clave/criptación son transferidos en estructuras específicas de transporte.

8.1 Funcionamiento seguro del canal H.245

Suponiendo que los procedimientos de conexión mencionados en la cláusula anterior (Procedimientos de establecimiento de la conexión) indiquen un modo de funcionamiento seguro, se llevará a cabo la toma de contacto y la autenticación negociadas para el canal lógico H.245 antes de que se intercambie cualquier mensaje H.245. Si se ha negociado, cualquier intercambio de certificados se producirá utilizando este mecanismo apropiado para los terminales de la serie H. Después de completar la seguridad del canal H.245, los terminales utilizarán el protocolo H.245 de la misma manera que si funcionasen en un modo inseguro.

8.2 Funcionamiento inseguro del canal H.245

Como otra posibilidad, el canal H.245 puede funcionar de una manera insegura y las dos entidades abren un canal lógico seguro con el cual efectuar la autenticación y/o la derivación de secreto compartido. Por ejemplo, se puede utilizar TLS (seguridad de nivel de transporte) o IPSEC (seguridad de protocolo Internet) abriendo un canal lógico con el **dataType** (**tipo de datos**) que

contiene un valor para **h235Control**. Este canal se utilizaría para derivar un secreto compartido que protege cualesquiera clave de sesión de medios o para transportar la **EncriptionSync** (**sincronización de criptación**).

8.3 Intercambio de capacidades

De acuerdo con los procedimientos de 8.3/H.245 (Procedimientos de intercambio de capacidades) y las Recomendaciones apropiadas relativas a sistemas de la serie H, los puntos extremos intercambian capacidades utilizando mensajes H.245. Estos conjuntos de capacidades pueden contener definiciones que indiquen parámetros de seguridad y criptación. Por ejemplo, un punto extremo pudiera proporcionar capacidades para enviar y recibir vídeo H.261. Puede señalizar también la posibilidad de enviar y recibir vídeo H.261 criptado.

Cada algoritmo de criptación que se utilice junto con un códec de medios determinado, supone una nueva definición de capacidad. Como con cualquier otra capacidad, los puntos extremos pueden suministrar códecs codificados independientes y dependientes en su intercambio. Esto permitirá a los puntos extremos ampliar sus capacidades de seguridad basadas en la tara y recursos disponibles.

Una vez completado el intercambio de capacidades, los puntos extremos pueden abrir canales lógicos seguros para los medios, de la misma manera que lo harían en un modo inseguro.

8.4 Cometido de terminal director

La determinación de terminal director-subordinado de H.245 se utiliza para establecer la entidad directora a los efectos del funcionamiento de canales bidireccionales y la resolución de otros conflictos. Este cometido de director se utiliza también en los métodos de seguridad. Aunque los modos de seguridad de un tren de medios son fijados por la fuente (en deferencia a las capacidades del receptor), el director es el punto extremo que genera la clave de criptación. Esta generación de la clave de criptación se hace con independencia de si el director es el receptor o la fuente de los medios criptados. Para efectuar el funcionamiento de canales multidistribución con claves compartidas, el controlador multipunto (también el director) debe generar las claves.

8.5 Señalización de canal lógico

Los puntos extremos abren canales lógicos de medios seguros de la misma manera que abren canales lógicos de medios inseguros. Cada canal puede funcionar de una manera completamente independiente con respecto a los otros canales, en particular cuando esto incumbe a la seguridad. El modo particular será definido en el campo **dataType** de **OpenLogicalChannel**. La clave de criptación inicial se transferirá en **OpenLogicalChannel** o **OpenLogicalChannelAck** dependiendo de la relación director/subordinado del originador de **OpenLogicalChannel**.

El **OpenLogicalChannelAck** actuará como una confirmación del modo de criptación. Si **OpenLogicalChannel** no es aceptable al recibiente, se devolverá **dataTypeNotSupported** (**tipo datos no soportado**) o **dataTypeNotAvailable** (**tipo datos no disponible**) (condición transitoria) en el campo de causa de **OpenLogicalChannelReject** (**rechazo apertura canal lógico**).

Durante el intercambio de protocolos que establece el canal lógico, la clave de criptación será transferida del terminal director al subordinado (con independencia de quién inició **OpenLogicalChannel**). Para los canales de medios abiertos por un punto extremo (que no sea el director), el director devolverá la clave de criptación inicial y el punto de sincronización inicial en **OpenLogicalChannelAck** (en el campo **encryptionSync**). Para los canales de medios abiertos por el director, **OpenLogicalChannel** incluirá la clave de criptación inicial y el punto de sincronización en el campo **encryptionSync**.

9 Procedimientos multipunto

9.1 Autenticación

La autenticación se producirá entre un punto extremo y la MC(U) (unidad de control multipunto) de la misma manera que se haría en una conferencia punto a punto. La MC(U) fijará la política relativa al nivel y rigor de autenticación. Como se indica en 6.6, se confía en la MC(U); los puntos extremos existentes en una conferencia pueden estar limitados por el nivel de autenticación empleado por la MC(U). Las nuevas instrucciones **ConferenceRequest/ConferenceResponse** (**petición conferencia/respuesta conferencia**) permiten que los puntos extremos obtengan de la MC(U) los certificados de otros participantes en la conferencia. Como se indica en los procedimientos H.245, los puntos extremos en una conferencia multipunto pueden solicitar cualquier otro certificado de punto extremo por medio del MC (control multipunto), pero no pueden realizar la autenticación criptográfica directa dentro del canal H.245.

9.2 Privacidad

La MC(U) ganará todos los intercambios director/subordinado y como tal suministrará las claves de criptación a los participantes en una conferencia multipunto. La privacidad para cada fuente dentro de una sesión común (suponiendo multidistribución) se puede lograr con claves individuales o comunes. Estos dos modos pueden ser elegidos arbitrariamente por la MC(U) y no serán controlables desde ningún punto extremo particular, salvo en modos permitidos por la política de la MC(U). En otras palabras, se puede utilizar una clave común a través de múltiples canales lógicos abiertos por diferentes fuentes.

10 Señalización y procedimientos de autenticación

10.1 Introducción

La autenticación se basa en general, bien en la utilización de un secreto compartido (usted está autenticado correctamente si conoce el secreto), bien en métodos de certificación que aplican claves públicas (usted prueba su identidad mediante el procesamiento de la clave privada correcta). Un secreto compartido y el empleo subsiguiente de la criptografía simétrica requiere que se produzca un contacto previo entre las entidades comunicantes. Un contacto cara a cara o contacto seguro previo puede ser sustituido por la generación o el intercambio de la clave secreta compartida en los métodos basados en la criptografía de claves públicas, por ejemplo, el intercambio de claves Diffie-Hellman. Las partes comunicantes en la generación y el intercambio de claves han de ser autenticadas mediante, por ejemplo, mensajes firmados digitalmente; en caso contrario, las partes de la comunicación no pueden estar seguras de con quien comparten el secreto.

Esta Recomendación presenta los métodos de autenticación basados en el abono, es decir, debe producirse un contacto previo para la compartición de un secreto, y se utilizarán métodos de autenticación que apliquen la criptografía de claves públicas para la autenticación o para la generación del secreto compartido.

10.2 Intercambio Diffie-Hellman con autenticación facultativa

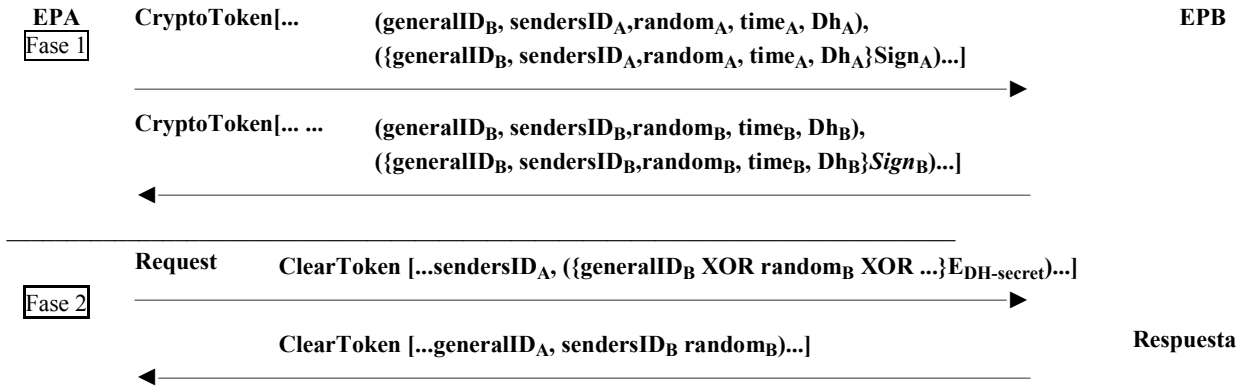
El propósito no es proporcionar autenticación absoluta a nivel de usuario. Este método proporciona la señalización para generar un secreto compartido entre dos entidades que pueden manipular material para comunicaciones privadas.

Al final de este intercambio ambas entidades poseerán una clave secreta compartida junto con un algoritmo elegido con el cual utilizar esta clave. Esta clave secreta compartida se puede utilizar en cualquier intercambio de petición/respuesta subsiguiente. Cabe señalar que, en casos muy raros, el intercambio Diffie-Hellman puede generar claves *débiles* conocidas para determinados algoritmos.

Cuando es así, cada entidad debe desconectar y reconectar para establecer un nuevo conjunto de claves.

La primera fase de la figura 1 siguiente muestra los datos intercambiados durante la negociación Diffie-Hellman. La segunda fase prevé que los mensajes de petición específicos de la aplicación o del protocolo sean autenticados por el respondedor. Obsérvese que se puede devolver un nuevo valor aleatorio con cada respuesta.

NOTA – Si el intercambio de mensajes se realiza por un canal inseguro, deben utilizarse las firmas digitales (u otro método de autenticación del origen de los mensajes) para autenticar las partes que compartirán el secreto. Se puede proporcionar también un elemento de firma facultativo, que se ilustra a continuación en ***cursivas***.



[... ...] indica una secuencia de testigos
() indica un testigo determinado, que puede contener múltiples elementos
{}E$_{DH-Secret}$ indica que los valores contenidos han sido criptados utilizando el secreto Diffie-Hellman
EPB sabe qué clave secreta compartida ha de utilizar para descifrar el identificador **generalID$_B$** asociándolo con el **generalID$_A$** que debe ser transferido también en el mensaje como **sendersID$_A$**. Obsérvese que el valor criptado en la fase 2 es transferido en el campo **generalID** de un **clearToken** para simplificar la codificación.

Figura 1/H.235 – Diffie-Hellman con autenticación facultativa

10.3 Autenticación basada en abono

10.3.1 Introducción

Aunque los procedimientos esbozados aquí (y los algoritmos de la ISO de los cuales se derivan) son bidireccionales, pueden ser utilizados solamente en un sentido si la autenticación se necesita solamente en ese sentido. Se describen los procedimientos de dos pasos y de tres pasos. La autenticación mutua (recíproca) de dos pasos sólo puede ejecutarse en un sentido cuando no es preciso autenticar los mensajes procedentes del sentido inverso. Estos intercambios suponen que cada extremo posee algún identificador bien conocido (como un identificador textual) que lo identifica inequívocamente. Para el procedimiento de dos pasos, se establece la hipótesis de que hay una referencia de tiempo mutuamente aceptable (de la cual derivar indicación de tiempo). La diferencia de hora que es aceptable es un asunto de la implementación local. El procedimiento de tres pasos utiliza un número de preguntas imprevisible generado aleatoriamente (que puede ser incrementado por un contador secuencial "aleatorio") como una pregunta procedente del autenticador. Este número aleatorio se utiliza para la protección contra los ataques de reproducción. A diferencia de los procedimientos de dos pasos, los procedimientos de tres pasos no autentican el primer mensaje inicial que contiene la pregunta del iniciador.

Hay tres variaciones diferentes que se pueden aplicar dependiendo de las necesidades:

1) Contraseña con criptación simétrica.

2) Contraseña con troceado.

3) Certificado con firma.

En todos los casos, el testigo contendrá la información descrita en las cláusulas siguientes según la variación elegida. Obsérvese que en todos los casos el **generalID** (**ID general**) puede ser conocido a través de la configuración o del directorio, en vez de en el intercambio de protocolos dentro de banda. Para simplificar el procesamiento en el receptor, el emisor debe incluir su identidad dentro de **sendersID** y fijar el **generalID** a la identificación del recipiente.

NOTA 1 – En todos los casos en los que son generadas indicaciones de tiempo y pasadas como parte de un intercambio de seguridad, los implementadores deben adoptar las precauciones que siguen. La granularidad de la indicación de tiempo debe ser suficientemente fina para que quede garantizado su incremento con cada mensaje. Si este incremento no está garantizado, pueden producirse ataques de reproducción (por ejemplo, si las indicaciones de tiempo sólo se incrementan de minuto en minuto, un punto extremo "C" puede engañar a un punto extremo "A" dentro del periodo de un minuto desde que el punto extremo "A" haya enviado un mensaje al punto extremo "B").

NOTA 2 – Si es de multidifusión, entonces el mensaje no está asegurado.

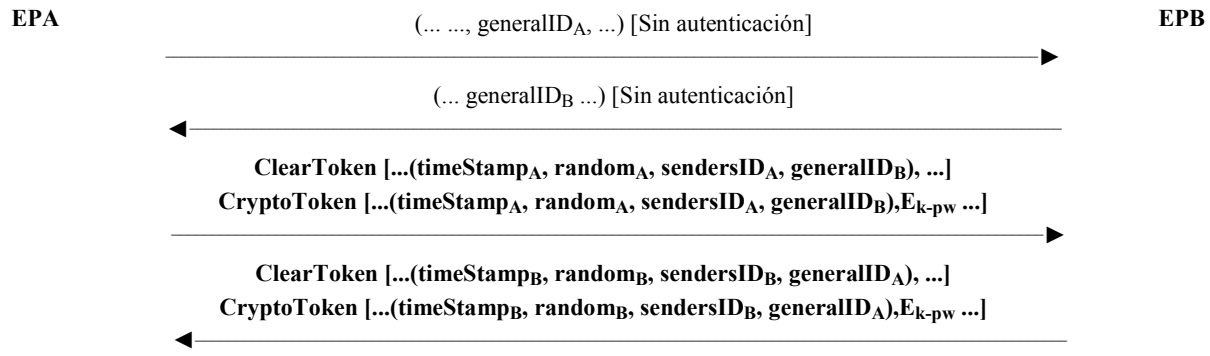
10.3.2 Contraseña con criptación simétrica

En las figuras 2a y 2b se muestra el formato de testigo y el intercambio de mensajes requeridos para realizar este tipo de autenticación en dos pasos o tres pasos, respectivamente. Este protocolo se basa en 5.2.1 ("two-pass") y 5.2.2 ("three-pass") de ISO/CEI 9798-2, y se supone que un identificador y la contraseña asociada son intercambiados durante el abono. La clave de criptación tiene una longitud de N octetos (según lo indicado por el AlgorithmID – ID de algoritmo), y se forma como sigue:

– Si la longitud de la contraseña = N, clave = contraseña.

– Si la longitud de la contraseña < N, la clave es rellenada con ceros.

– Si la longitud de la contraseña > N, los primeros N octetos son asignados a la clave, después el N + M-ésimo octeto de la contraseña se pone a XOR al Mmod(N)-ésimo octeto (para todos los octetos después de N), (es decir, todos los octetos de contraseña "suplementarios" son doblados repetidamente en la clave por XOR).



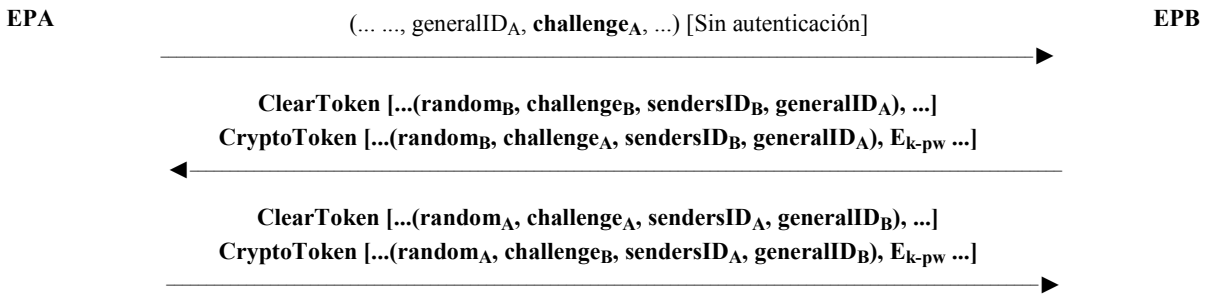
NOTA 1 – La devolución de testigo del EPB (punto extremo B) es facultativa; si se omite, sólo se logra una autenticación unidireccional.

NOTA 2 – **E_{k-pw}** indica valores que han sido criptados utilizando la clave "k" derivada de la contraseña "pw".

NOTA 3 – "**random**" es un contador monotónicamente creciente que realiza múltiples mensajes con la misma indicación de tiempo única.

NOTA 4 – En el tercer mensaje el EPA (punto extremo A) proporciona un **ClearToken** separado, que se identifica por el mismo OID que el OID del **CryptoToken**; y viceversa, sucede de manera similar para el 4º mensaje.

Figura 2a/H.235 – Contraseña con criptación simétrica; dos pasos



NOTA 1 – **challengeA** y la devolución del **CryptoToken** criptado de B a A no son necesarias si se desea una autenticación unidireccional.

NOTA 2 – **E_{k-pw}** indica valores que han sido criptados utilizando la clave "k" derivada de la contraseña "pw".

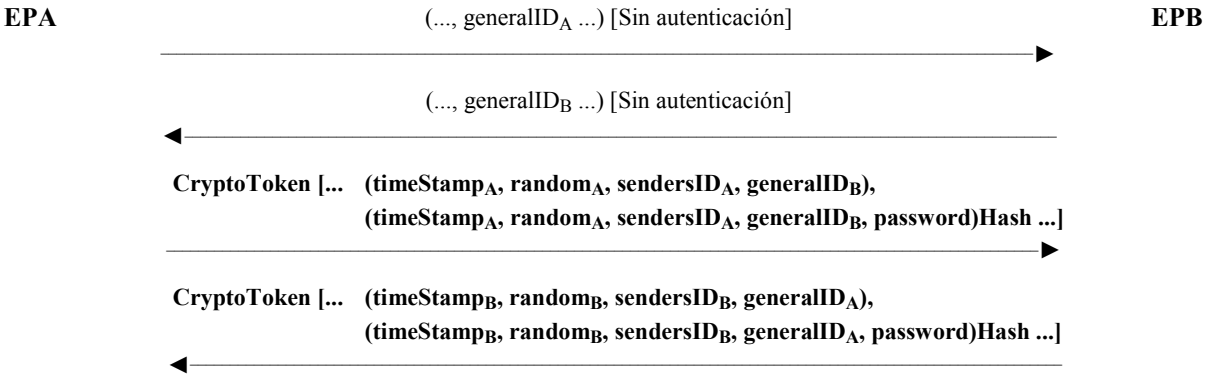
NOTA 3 – En el tercer mensaje el EPA proporciona una nueva **challenge$_A$** en texto claro en un **ClearToken** independiente, que es identificada por el mismo OID que el OID del **CryptoToken**. EPA también devuelve la **challenge$_B$** criptada como respuesta; y viceversa, sucede de manera similar para el 2º mensaje.

NOTA 4 – Para múltiples mensajes pendientes "**random**" (es decir, un contador monotónicamente creciente) deberá formular una pregunta única.

Figura 2b/H.235 – Contraseña con criptación simétrica; tres pasos

10.3.3 Contraseña con troceado

En las figuras 3a y 3b se muestra el formato de testigo y el intercambio de mensajes requeridos para realizar este tipo de autenticación para dos pasos o tres pasos, respectivamente. Este protocolo se basa en 5.2.1 y 5.2.2 de ISO/CEI 9798-4, y se supone que un identificador y la contraseña asociada son intercambiados durante el abono. El anexo D proporciona una descripción detallada del procedimiento de troceado de dos pasos.

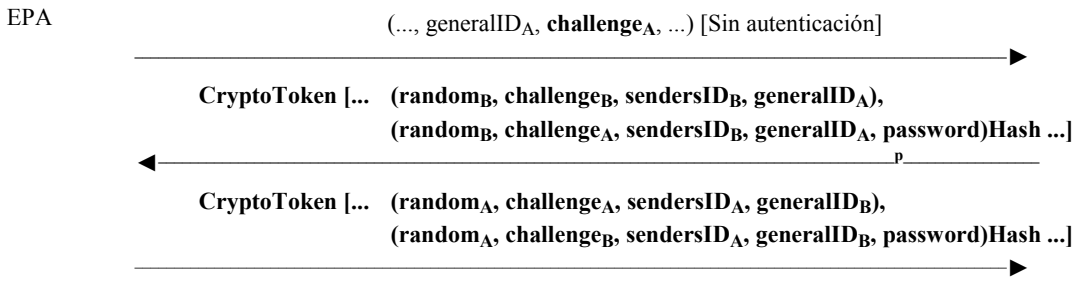


NOTA 1 – La devolución de testigo del EPB es facultativa; si se omite, sólo se logra una autenticación unidireccional.

NOTA 2 – **Hash** indica una función de troceado que opera sobre los valores contenidos.

NOTA 3 – "**random**" es un contador monotónicamente creciente que realiza múltiples mensajes con la misma indicación de tiempo única.

Figura 3a/H.235 – Contraseña con troceado; dos pasos



NOTA 1 – La devolución de testigo del EPB es facultativa; si se omite, sólo se logra una autenticación unidireccional.

NOTA 2 – **Hash** indica una función de troceado que opera sobre los valores contenidos.

NOTA 3 – En el tercer mensaje el EPA proporciona una nueva **challengeA** en texto claro dentro del **ClearToken** insertado en **cryptoHashedToken**. EPA también devuelve la **challengeB** troceada como respuesta; y viceversa, sucede de manera similar para el 2º mensaje.

NOTA 4 – Para múltiples mensajes pendientes **random** (es decir, un contador monotónicamente creciente) deberá formular una pregunta única.

Figura 3b/H.235 – Contraseña con troceado; tres pasos

NOTA 1 – La estructura **cryptoHashedToken** se utiliza para transferir los parámetros utilizados en este intercambio. En esta estructura están incluidas las versiones "claro" de los parámetros necesarios para calcular el valor de troceado. Los implementadores deberán incluir la indicación de tiempo en el **hashedVals** y *no* deberán incluir la contraseña. (Por ejemplo, ambas contraseñas y el '**generalID**' deben ser conocidos por el recipiente previamente; las primeras pueden omitirse.)

NOTA 2 – La función de troceado deberá aplicarse a la estructura **EncodedGeneralToken** que incluye al menos los campos ID, indicación de tiempo y contraseña. El valor de la contraseña NO deberá ser transferido en el **ClearToken.**

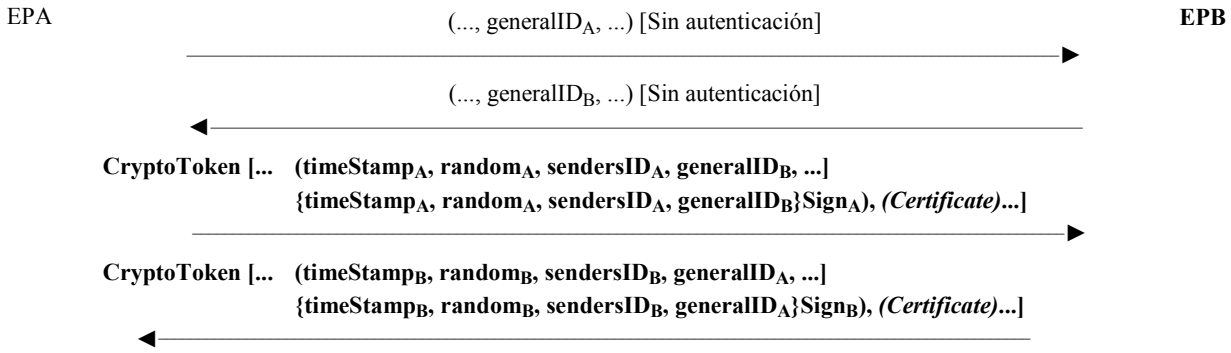
NOTA 3 – Las implementaciones deben garantizar que las contraseñas introducidas por el usuario transportan suficiente entropía. Las contraseñas que son demasiado cortas o que son vulnerables a los ataques de diccionario deben ser rechazadas. En determinados casos puede ser ventajosa la aplicación de frases de paso introducidas por el usuario a través de una función de troceado criptográfico y la utilización de los bits resultantes.

10.3.4 Certificado con firma

En las figuras 4a y 4b se muestra el formato de testigo y los mensajes intercambiados requeridos para realizar este tipo de autenticación. Este protocolo se basa en 5.2.1 de ISO/CEI 9798-3, y se supone que un identificador y el certificado asociado son asignados/intercambiados durante el abono. El anexo E proporciona una descripción detallada del procedimiento de firma de dos pasos.

NOTA 1 – Se puede proporcionar también un elemento de certificado facultativo, que se ilustra a continuación en ***cursivas***.

NOTA 2 – Si el mensaje es de multidifusión, el identificador del destino (**generalID$_B$** para mensajes originados en A y viceversa) no debe ser incluido en el **ClearToken.**



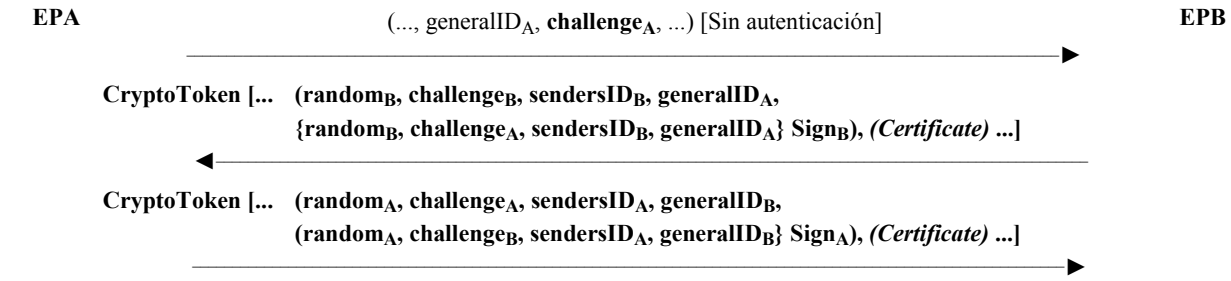
NOTA 1 – La devolución de testigo del EPB es facultativa; si se omite, sólo se logra una autenticación unidireccional.

NOTA 2 – Un certificado de tipo "pago" puede ser incluido facultativamente por el originador EPA.

NOTA 3 – **Sign** indica una función de firma (del certificado asociado) realizada en los valores contenidos.

NOTA 4 – "**random**" es un contador monotónicamente creciente que realiza múltiples mensajes con la misma indicación de tiempo única.

Figura 4a/H.235 – Certificado con firma; dos pasos



NOTA 1 – La devolución de testigo del EPB es facultativa; si se omite, sólo se logra una autenticación unidireccional.

NOTA 2 – Un certificado de tipo "pago" puede ser incluido facultativamente por el originador EPA.

NOTA 3 – **Sign** indica una función de firma (del certificado asociado) realizada en los valores contenidos.

NOTA 4 – En el tercer mensaje el EPA proporciona una nueva **challenge$_A$** en texto claro con el **GeneralToken** codificado insertado. El EPA también devuelve la **challenge$_B$** firmada como respuesta; y viceversa, sucede de manera similar para el 2º mensaje.

NOTA 5 – Para múltiples mensajes pendientes **random** (es decir, un contador monotónicamente creciente) deberá formular una pregunta única.

Figura 4b/H.235 – Certificado con firma; tres pasos

10.3.5 Utilización de contraseñas y secreto compartido

La presente Recomendación utiliza algunas técnicas de criptografía simétrica a efectos de autenticación, integridad y confidencialidad. Este texto usa los términos contraseña y secreto compartido cuando se refiere a técnicas simétricas. Se entiende por secreto compartido el término genérico que identifica una cadena de bits cualquiera. El secreto compartido puede ser asignado o configurado durante el proceso de suscripción de abono del usuario, o puede formar parte de un sistema de cálculo dentro de banda, por ejemplo, un secreto compartido derivado de Diffie-Hellman.

Una contraseña puede verse como una cadena de caracteres alfanuméricos que puede ser memorizada por los usuarios. Es obvio que el uso de las contraseñas debe hacerse con cuidado: las contraseñas sólo son suficientemente seguras cuando se escogen al azar dentro de una muestra suficientemente amplia, cuando portan suficiente entropía de manera tal que son impredecibles y cuando se cambian periódicamente. Las reglas para escoger y actualizar las contraseñas están fuera del alcance de esta Recomendación.

Una buena práctica, para aprovechar las ventajas de las contraseñas y los secretos compartidos, es la de transformar la cadena contraseña del usuario en una cadena de bits como el secreto compartido, usando una función criptográficamente fuerte de troceo unidireccional.

Ejemplo recomendado, cuando se usa el perfil de seguridad del anexo D, es la aplicación de la función de troceado SHA-1 a la cadena contraseña, con lo que se obtiene un secreto compartido de 20 bytes. La ventaja es que el resultado troceado no sólo oculta la contraseña real sino que también define un formato de cadena de bits de longitud fija sin realmente sacrificar entropía.

Esto es,

secreto compartido := SHA1 (contraseña)

11 Procedimiento de criptación de tren de medios

Los trenes de medios se codificarán utilizando el algoritmo y la clave presentados en el canal H.245. Las figuras 5 y 6 muestran el flujo general. Obsérvese que el encabezamiento de transporte se adjunta a la unidad de datos de servicio (SDU) de transporte después que la SDU ha sido criptada. Los segmentos opacos indican privacidad. A medida que el transmisor recibe nuevas claves y éstas son utilizadas en la criptación, el encabezamiento SDU indicará de alguna manera al receptor que ahora se está utilizando la nueva clave. Por ejemplo, en UIT-T H.323 el encabezamiento RTP (SDU) cambiará su tipo de cabida útil para indicar la conmutación a la nueva clave.

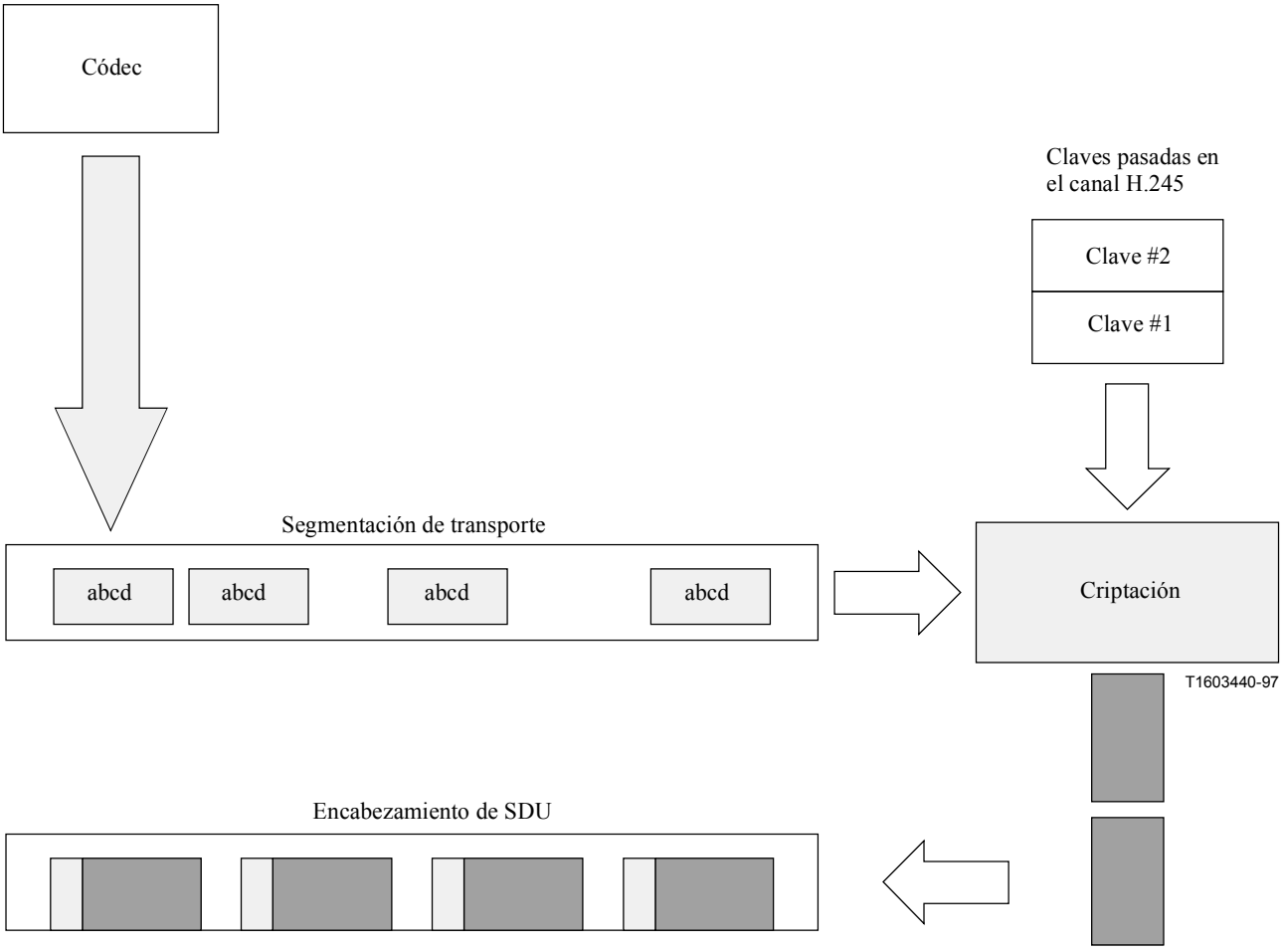


Figura 5/H.235 – Criptación de trenes de medios

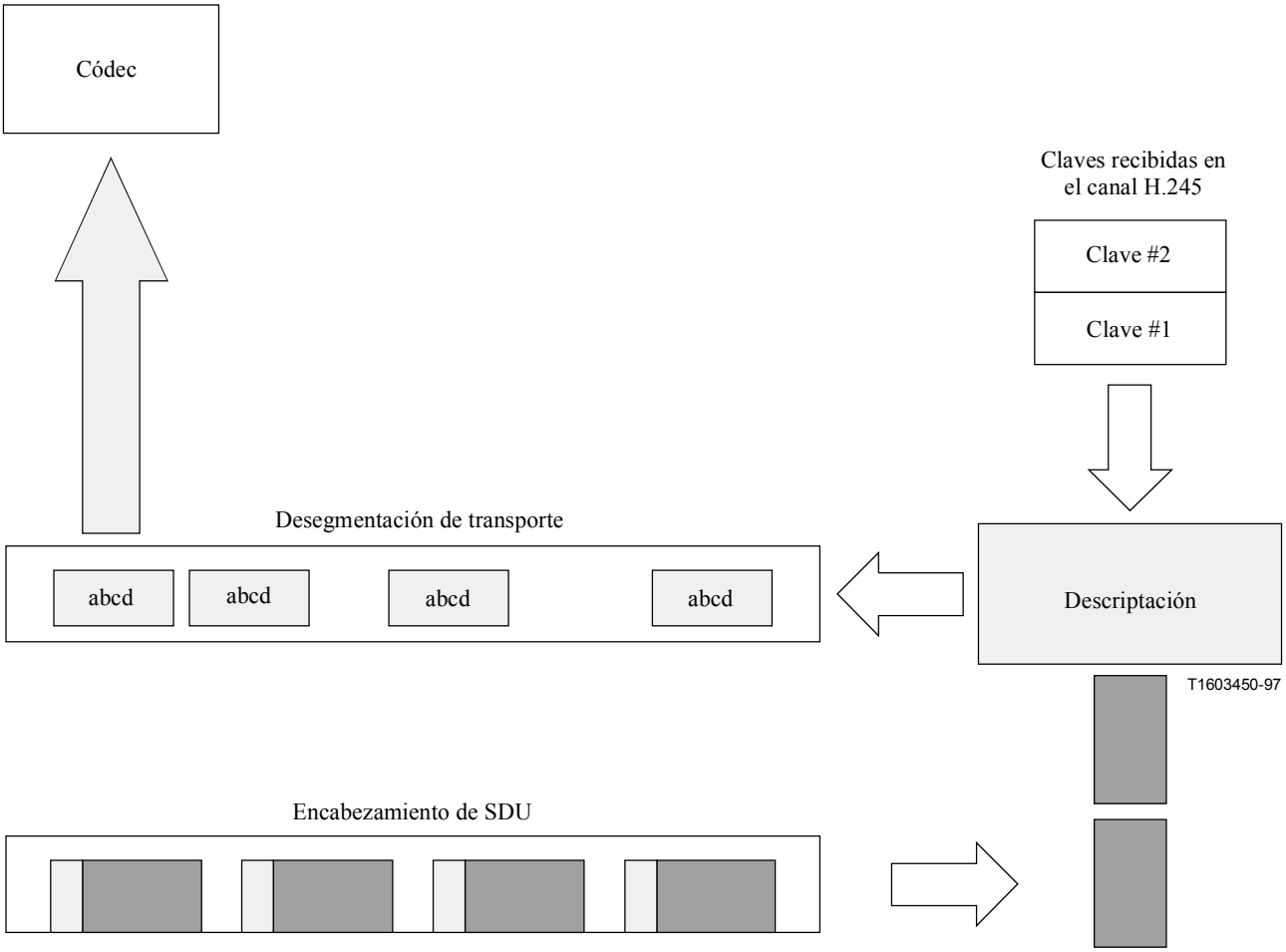


Figura 6/H.235 – Descriptación de trenes de medios

11.1 Claves de sesión de medios

h235Key (**clave h235**) se incluye en **encryptionUpdate** (**actualización de criptación**). **h235Key** está codificada en ASN.1 dentro del contexto del árbol ASN.1 del protocolo H.235 y se transfiere como una cadena de octetos opaca con respecto al protocolo H.245. Se puede proteger la clave utilizando uno de los tres mecanismos posibles a medida que son transferidos entre dos puntos extremos.

- Si el canal H.245 es seguro, no se aplica protección adicional al material de claves. La clave se transfiere en "claro" con respecto a este campo; se utiliza la opción ASN.1 de **secureChannel** (**canal seguro**).
- Si se ha establecido una clave y un algoritmo secretos fuera del canal H.245 (es decir, fuera del protocolo H.323 o en un canal lógico **h235Control**), el secreto compartido se utiliza para criptar el material de clave, y se incluye la clave cifrada resultante. En este caso, se utiliza la opción ASN.1 de **sharedSecret** (**secreto compartido**).
- Se pueden utilizar certificados cuando el canal H.245 no es seguro, pero se pueden utilizar también además para el canal H.245 seguro. Cuando se emplean certificados, el material de claves es cifrado utilizando la clave pública del certificado y el constructivo ASN.1 **certProtectedKey** (**clave protegida de certificado**).

En cualquier punto en una conferencia, un receptor (o un transmisor) puede solicitar una nueva clave (**encryptionUpdateRequest**). Una razón para hacer esto pudiera ser si se sospecha que se ha perdido la sincronización de uno de los canales lógicos. El terminal director que recibe esta petición generará

nuevas claves en respuesta a esta instrucción y puede decidir también asíncronamente distribuir nuevas claves y, si lo hace así, utilizará el mensaje **encryptionUpdate**.

Después de recibir una **encryptionUpdateRequest**, el terminal director enviará **encryptionUpdate**. Si se trata de una conferencia multipunto, el MC (también el director) distribuirá la nueva clave a todos los receptores antes de dar esta clave al transmisor. El transmisor de los datos por el canal lógico utilizará la nueva clave tan pronto sea posible después de recibir el mensaje.

Un transmisor (que se supone no es el director) puede solicitar también una nueva clave. Si el transmisor forma parte de una conferencia multipunto, el procedimiento será el siguiente:

- El transmisor enviará **encryptionUpdateRequest** al MC (director).
- El MC debe generar una nueva clave y enviar un mensajes **encryptionUpdate** a todos los participantes en la conferencia, salvo al transmisor.
- Después de distribuir las nuevas claves a todos los participantes, el MC enviará **encryptionUpdate** al transmisor que utilizará entonces la nueva clave.

11.2 Antiinundación de medios

El receptor de un tren de medios RTP puede desea contrarrestar los ataques de tipo inundación y de denegación del servicio en los puertos RTP/UDP descubiertos. Cuando tienen implementada la capacidad antiinundación, los receptores pueden determinar rápidamente si un paquete RTP obtenido procede de una fuente no autorizada y en tal caso descartarlo.

Cuando se fija, la capacidad antiinundación indica el empleo del mecanismo antiinundación:

- bien para datos de medios de texto claro sin criptación de medios (véase el caso 1 más abajo); o
- bien en combinación con datos de medios criptados cuando **EncryptionCapability** caracteriza un algoritmo de criptación (véase el caso 2 más abajo).

Ambas opciones proporcionan una **autenticación de paquetes RTP** de poco peso en campos seleccionados mediante un código de autenticación de mensajes (MAC, *message authentication code*) calculado. El MAC puede ser calculado utilizando los identificadores de objeto definidos en 11.2.1. Los algoritmos criptográficos están constituidos por:

- un algoritmo de criptación (por ejemplo, DES en modo MAC; véase ISO/CEI 9797). DES en MAC se indica mediante el OID "S", mientras que DES triple en MAC se indica mediante el OID "O"; o
- utilizando una función unidireccional criptográfica (por ejemplo, SHA1). Se utilizará el OID "M".

El algoritmo MAC se indica en el identificador de objeto de **antiSpamAlgorithm**. El OID del algoritmo indica también implícitamente el tamaño del MAC; por ejemplo, 1 bloque = 64 bits para DES MAC. Para ahorrar anchura de banda, el MAC puede ser truncado si bien sacrificando alguna seguridad; por ejemplo, pasando a un MAC de 32 bits; esto requiere utilizar entonces un identificador de objeto diferente. El método antiinundación es independiente de cualquier criptación de cabida útil adicional (véanse los casos 1 y 2 más adelante).

La antiinundación utiliza el siguiente formato de paquete RTP (véase la figura 7), en el que la secuencia de relleno RTP se interpreta como sigue (véase A.5/H.225.0).

- El bit P del encabezamiento RTP se fijará a 1.
- Se añadirán bytes de relleno al final de la cabida útil con el significado siguiente:

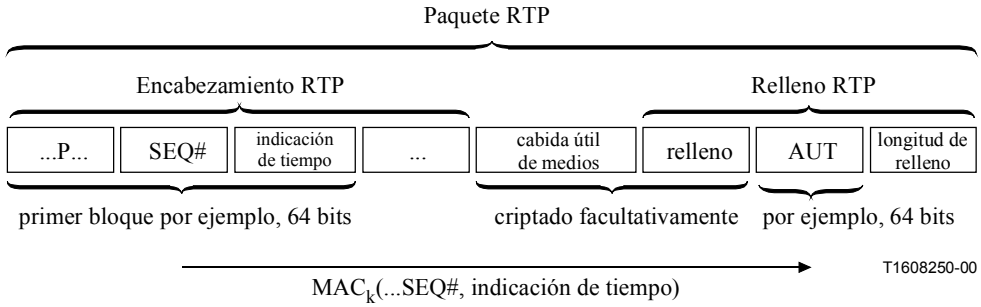


Figura 7/H.235 – Formato de paquete RTP para la antiinundación de medios

NOTA 1 – Si no se utiliza la antiinundación, tampoco se utilizan los campos AUT y longitud de relleno y se aplica el formato de paquete RTP normal.

1) Caso de antiinundación solamente:

Este caso se aplica cuando los datos de medios no están criptados y los campos de relleno se han dejado vacíos. El último octeto del relleno RTP contiene una cuenta del número de octetos que deberán ser ignorados al final del paquete RTP. Los otros bytes de relleno transportan el MAC. El MAC deberá ser calculado sobre el primer bloque criptográfico del encabezamiento RTP que incluye la indicación de tiempo variable y el número secuencial utilizando el algoritmo MAC negociado de **antiSpamAlgorithm** y aplicando el secreto simétrico. Un secreto compartido estático o configurado manualmente, o un secreto *k* compartido negociado dinámicamente puede utilizarse de conformidad con los procedimientos de UIT-T H.235. Para tamaños de bloque superiores (más de 64 bits), deberán tomarse algunos bits adicionales suficientes del encabezamiento RTP o incluso la primera cabida útil de medios.

Como clave para el cálculo de MAC se recomienda utilizar la clave obtenida a partir de la distribución de claves de sesión de medios H.235; aún cuando la clave de sesión aplicada no se utiliza para la criptación de cabida útil. Para la gestión de claves se puede utilizar una conexión rápida segura con establecimiento de claves (véase anexo J/H.323) o la asignación manual de claves. El emisor calcula el MAC como se ha descrito anteriormente e incluye el resultado en el campo MAC del campo AUT del relleno RTP. El emisor y el receptor conocen el tamaño del campo AUT y la longitud del MAC mediante el **antiSpamAlgorithm**.

La verificación del MAC en el lado receptor debería realizarse cuanto antes, si fuera posible ya dentro de la pila RTP o a más tardar antes de la descriptación o descompresión de la cabida útil. El receptor recalcula en primer lugar el MAC del mismo modo que lo hizo el emisor y compara el MAC calculado con el MAC entregado en el relleno RTP. Si existe discordancia entre los MAC, ello significa que el encabezamiento RTP ha sido modificado en tránsito ha sido enviado por una entidad no autorizada que no es propietaria de la clave. Por ello, el paquete RTP autenticado equivocadamente deberá ser descartado y el evento puede ser registrado; esto probablemente indica una tentativa de ataque de denegación del servicio. En caso contrario, el paquete RTP autenticado puede ser procesado posteriormente, el relleno RTP es eliminado y la cabida útil es suministrada a través del códec.

NOTA 2 – El cálculo/verificación del MAC ligero con criptación DES implica sólo una operación de criptación única; alternativamente, se calcula el MAC SHA1 sobre una parte pequeña de los paquetes de longitud fija, de modo que las operaciones criptográficas consumen decididamente recursos de procesamiento mínimos.

2) Caso del método antiinundación y criptación de la cabida útil:

Este caso se aplica cuando se efectúa una criptación de los datos de medios y se invoca el método antiinundación. Cuando la cabida útil no cae sobre las fronteras exactas de los bloques, se han de añadir algunos bytes de relleno adicionales a la cabida útil delante del MAC. La criptación de la cabida útil de medios es conforme con esta cláusula 11.

EncryptionCapability define el algoritmo de criptación de cabida útil mientras que **antiSpamAlgorithm** define el método antiinundación. Por motivos de seguridad, la criptación de medios y el MAC deberán utilizar diferentes claves de sesión. La clave *k* de MAC se calcula suministrando la clave de criptación *K* a través de la función de troceado unidireccional SHA1;

k = SHA1(*K*); deberán tomarse suficientes bits del resultado troceado en el orden de bytes de red. Cuando el **antiSpamAlgorithm** indica un algoritmo de criptación, los bits recopilados deberán formar una clave de criptación correcta; por ejemplo, fijando los bits de paridad de DES.

Después de que el receptor haya verificado con éxito la autenticidad del paquete RTP, se descripta la cabida útil y se descarta el relleno RTP. El procedimiento general es conforme con el caso 1 anterior.

11.2.1 Lista de identificadores de objeto

En el cuadro 1 se listan todas las referencias de los OID.

Cuadro 1/H.235 – Identificadores de objeto utilizados para la antiinundación

Referencia del identificador de objeto	Valor del identificador de objeto	Descripción
"M"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 8}	antiinundación que utiliza HMAC-SHA1-96
"N"	{iso(1), identified-organization(3), oiw(14), secsig(3), algorithm(2), desMAC(10)}	antiinundación que utiliza MAC DES (56 bits) (véase ISO/CEI 9797) con MAC de 64 bits.
"O"	{iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) desEDE(17)}	antiinundación que utiliza DES triple en MAC (168 bits) (véase ISO/CEI 9797)

12 Recuperación tras error de seguridad

Esta Recomendación no especifica ni recomienda métodos por los cuales los puntos extremos puedan supervisar su privacidad absoluta. Sin embargo, sí recomienda acciones que se han de ejecutar cuando se detecta la pérdida de privacidad.

Si cualquiera de los dos puntos extremos detecta una brecha en la seguridad del canal de conexión de la llamada (por ejemplo, H.225.0 para H.323), debe cerrar inmediatamente la conexión aplicando los procedimientos de protocolo apropiados al punto extremo en cuestión [para 8.5/H.323 con la excepción del paso 5)].

Si cualquiera de los dos puntos extremos detecta una brecha en la seguridad del canal H.245 o del canal lógico (**h235Control**) de datos seguro, debe cerrar inmediatamente la conexión aplicando los procedimientos de protocolo apropiados al punto extremo en cuestión [para 8.5/H.323 con la excepción del paso 5)].

Si cualquier punto extremo detecta una pérdida de privacidad en uno de los canales lógicos, debe solicitar inmediatamente una nueva clave (**encryptionUpdateRequest**) y/o cerrar el canal lógico. A discreción de la MC(U) una pérdida de privacidad en el canal lógico puede provocar el cierre de todos los otros canales lógicos y/o la creación de nuevas claves a discreción de la MC(U). La MC(U)

enviará **encryptionUpdateRequest**, **encryptionUpdate** a cualquier y a todos lo puntos extremos afectados.

A discreción de la MC(U), un error de seguridad habido en un canal puede provocar el cierre de las conexiones en todos los puntos extremo de la conferencia, terminándola así.

13 Autenticación asimétrica e intercambio de claves utilizando sistemas criptográficos de curva elíptica

Esta Recomendación proporciona técnicas de curva elíptica perfeccionadas con aplicaciones a la firma, la gestión de claves y la criptación. Una de las ventajas principales con respecto a las técnicas asimétricas "clásicas" como el algoritmo RSA son:

- Las claves criptográficas más cortas ofrecen una seguridad comparable al algoritmo RSA: Los criptosistemas de curva elíptica tienen longitudes típicas de claves de 160 bits; es decir, ofrecen una seguridad equivalente a una clave RSA de 1024 bits. Las claves más cortas consumen menos memoria de almacenamiento y hacen los sistemas criptográficos de curva elíptica especialmente atractivos para su implementación en las tarjetas inteligentes, y en cualquier otro dispositivo con necesidades de memoria pequeñas. En el contexto de H.323, los tipos de puntos extremos simples de audio asegurados (SASET, *secured audio simple endpoint types*) basados en el anexo J/H.323 debido a su bajo precio resultan muy adecuados para el despliegue de las técnicas de curva elíptica.
- La velocidad mejorada de procesamiento que se alcanza en las implementaciones tanto de soporte físico como de soporte lógico: Las claves más cortas mejoran la velocidad de procesamiento. Como resultado, las respuestas interactivas (del usuario) son más rápidas.

En [*ATM Forum Security Specificación Version 1.1*, sección 8.7] puede verse la información básica, la explicación y los procedimientos de procesamiento de la criptografía de curva elíptica. Se recomienda codificar los puntos elípticos en su notación no comprimida afín sin utilizar el método de compresión/descompresión de punto. En [ISO/CEI 15946-1] e [ISO/CEI 15946-2] se dispone de más información sobre este tema.

13.1 Gestión de claves

Los esquemas del convenio de claves Diffie-Hellman basados en la curva elíptica son similares al caso mod-p clásico definido también en la presente Recomendación. Se presentan dos situaciones:

- curvas elípticas sobre un campo primo: **eckasdhp** contiene los parámetros Diffie-Hellman y de curva elíptica;
- curvas elípticas de característica 2: **eckasdh2** contiene los parámetros Diffie-Hellman y de curva elíptica.

La estructura ECKASDH soporta cualquiera de los dos casos. En [ISO/CEI 15946-1] se da una lista de algunos ejemplos de curvas elípticas. Se puede utilizar también cualquier otra curva elíptica adecuada.

Como se dispone de una estructura secuenciada del **ClearToken**, las señalizaciones **dhkey** y **eckasdhkey** no se deberían producir a la vez: sólo una de ellas deberá estar presente cuando se aplica el intercambio de claves Diffie-Hellman.

Observación: No se deben confundir los parámetros secretos elegidos aleatoriamente, ***a*** por la parte A o ***b*** por la parte B, con los coeficientes Weierstrass comunes ***a***, ***b***.

13.2 Firma digital

El campo **ECGDSASignature** transporta los valores **r** y **s** de la firma digital basada en la curva elíptica calculada. En la sección 8.7.3 de *ATM Security Specification Version 1.1* y en el capítulo 5 de ISO 15946-2 se proporciona más información acerca del algoritmo de firmas EC-GDSA.

La firma digital basada en la curva elíptica **ECGDSA** deberá ser codificada en ASN.1 e introducida a continuación en el campo **signature** del macro **SIGNED** de esta Recomendación. Para la firma digital el emisor deberá incluir un identificador de objeto en el **algorithmOID** mediante el cual el recipiente sea capaz de determinar la utilización de una firma digital de curva elíptica.

ANEXO A

ASN.1 del protocolo H.235

```
H235-SECURITY-MESSAGES DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

-- EXPORTS All

ChallengeString         ::= OCTET STRING (SIZE(8..128))
TimeStamp               ::= INTEGER(1..4294967295) -- seconds since 00:00 1/1/1970 UTC
RandomVal               ::= INTEGER -- 32-bit Integer
Password                ::= BMPString (SIZE (1..128))
Identifier              ::= BMPString (SIZE (1..128))
KeyMaterial             ::= BIT STRING(SIZE(1..2048))

NonStandardParameter ::= SEQUENCE
{
        nonStandardIdentifier   OBJECT IDENTIFIER,
        data                    OCTET STRING
}

-- if local octet representations of these bit strings are used they shall
-- utilize standard Network Octet ordering (e.g. Big Endian)
DHset ::= SEQUENCE
{
        halfkey         BIT STRING (SIZE(0..2048)), -- = g^x mod n
        modSize         BIT STRING (SIZE(0..2048)), --  n
        generator       BIT STRING (SIZE(0..2048)), -- g
        ...
}

ECpoint ::= SEQUENCE -- uncompressed (x, y) affine coordinate representation of an elliptic curve point
{
        x               BIT STRING (SIZE(0..511)) OPTIONAL,
        y               BIT STRING (SIZE(0..511)) OPTIONAL,
        ...
}

ECKASDH::= CHOICE -- parameters for elliptic curve key agreement scheme Diffie-Hellman
{
        eckasdhp SEQUENCE -- parameters for elliptic curves of prime field
        {
                public-key      ECpoint, -- This field contains representation of the ECKAS-DHp public key value.
                        --This field contains the initiator's ECKAS-DHp public key value (aP) when this information
                        -- element is sent from originator to receiver. This field contains the responder's ECKAS-DHp
                        -- public key value (bP) when this information element is sent back from receiver
                        -- to originator.
                modulus         BIT STRING (SIZE(0..511)), -- This field contains representation of the
```

```
                -- ECKAS-DHp public modulus value (p).
        base            ECpoint, -- This field contains representation of the ECKAS-DHp public base (P).
        weierstrassA    BIT STRING (SIZE(0..511)), --This field contains representation of the
                -- ECKAS-DHp Weierstrass coefficient (a).
        weierstrassB    BIT STRING (SIZE(0..511)) --This field contains representation of the
                -- ECKAS-DHp Weierstrass coefficient (b).
    },

    eckasdh2 SEQUENCE -- parameters for elliptic curves of characteristic 2
    {
        public-key      ECpoint, -- This field contains representation of the ECKAS-DH2 public key value.
                -- This field contains the initiator's ECKAS-DH2 public key value (aP) when this information
                -- element is sent from originator to receiver. This field contains the responder's ECKAS-DH2
                -- public key value (bP) when this information element is sent back from receiver to originator.
        fieldSize       BIT STRING (SIZE(0..511)), -- This field contains representation of the
                -- ECKAS-DH2 field size value (m).
        base            ECpoint, -- This field contains representation of  the ECKAS-DH2 public base (P).
        weierstrassA    BIT STRING (SIZE(0..511)), --This field contains representation of the
                -- ECKAS-DH2 Weierstrass coefficient (a).
        weierstrassB    BIT STRING (SIZE(0..511)) --This field contains representation of the
                -- ECKAS-DH2 Weierstrass coefficient (b).
    },
    ...
}

ECGDSASignature::= SEQUENCE -- parameters for elliptic curve digital signature algorithm
{
    r       BIT STRING (SIZE(0..511)), -- This field contains the representation of the r component of the
                -- ECGDSA  digital signature.
    s       BIT STRING (SIZE(0..511)) -- This field contains the representation of the s component of the
                -- ECGDSA  digital signature.
}

TypedCertificate ::= SEQUENCE
{
    type            OBJECT IDENTIFIER,
    certificate     OCTET STRING,
    ...
}


AuthenticationBES ::= CHOICE

{
    default     NULL, -- encrypted ClearToken
    radius      NULL, -- RADIUS-challenge/response
    ...

}

AuthenticationMechanism ::= CHOICE
{
    dhExch          NULL, -- Diffie-Hellman
    pwdSymEnc       NULL, -- password with symmetric encryption
    pwdHash         NULL, -- password with hashing
    certSign        NULL, -- Certificate with signature
    ipsec           NULL, -- IPSEC based connection
    tls             NULL,
    nonStandard     NonStandardParameter, -- something else.
    ...,
    authenticationBES  AuthenticationBES -- user authentication for BES
}

ClearToken      ::= SEQUENCE  -- a "token" may contain multiple value types.
{
```

```
        tokenOID            OBJECT IDENTIFIER,
        timeStamp           TimeStamp OPTIONAL,
        password            Password OPTIONAL,
        dhkey               DHset OPTIONAL,
        challenge           ChallengeString OPTIONAL,
        random              RandomVal OPTIONAL,
        certificate         TypedCertificate OPTIONAL,
        generalID           Identifier OPTIONAL,
        nonStandard         NonStandardParameter OPTIONAL,
        ...,
        eckasdhkey          ECKASDH OPTIONAL, -- elliptic curve Key Agreement Scheme-Diffie
                                              -- Hellman Analogue (ECKAS-DH)
        sendersID           Identifier OPTIONAL
}

--      An object identifier should be placed in the tokenOID field when a
--      ClearToken is included directly in a message (as opposed to being
--      encrypted).  In all other cases, an application should use the
--      object identifier { 0 0 } to indicate that the tokenOID value is not present.
--
-- Start all the cryptographic parameterized types here...
--

SIGNED { ToBeSigned } ::= SEQUENCE {
        toBeSigned          ToBeSigned,
        algorithmOID        OBJECT IDENTIFIER,
        paramS              Params,       -- any "runtime" parameters
        signature           BIT STRING -- could be an RSA or an ASN.1 coded ECGDSASignature
} ( CONSTRAINED BY { -- Verify or Sign Certificate -- } )


ENCRYPTED { ToBeEncrypted } ::= SEQUENCE {
        algorithmOID            OBJECT IDENTIFIER,
        paramS                  Params,       -- any "runtime" parameters
        encryptedData           OCTET STRING
} ( CONSTRAINED BY { -- Encrypt or Decrypt -- ToBeEncrypted } )

HASHED { ToBeHashed } ::= SEQUENCE {
        algorithmOID            OBJECT IDENTIFIER,
        paramS                  Params,       -- any "runtime" parameters
        hash                    BIT STRING
} ( CONSTRAINED BY { -- Hash -- ToBeHashed } )

IV8 ::= OCTET STRING (SIZE(8)) -- initial value for 64-bit block ciphers
IV16 ::= OCTET STRING (SIZE(16)) -- initial value for 128-bit block ciphers

-- signing algorithm used must select one of these types of parameters
-- needed by receiving end of signature.

Params ::= SEQUENCE {
        ranInt              INTEGER OPTIONAL, -- some integer value
        iv8                 IV8 OPTIONAL,    -- 8 octet initialization vector
        ...,
        iv16                IV16 OPTIONAL   -- 16 octet initialization vector
}

EncodedGeneralToken ::= TYPE-IDENTIFIER.&Type (ClearToken -- general usage token -- )
PwdCertToken ::= ClearToken (WITH COMPONENTS {..., timeStamp PRESENT, generalID PRESENT})
EncodedPwdCertToken ::= TYPE-IDENTIFIER.&Type (PwdCertToken)

CryptoToken::= CHOICE
{
```

```
        cryptoEncryptedToken SEQUENCE -- General purpose/application specific token
        {
                tokenOID    OBJECT IDENTIFIER,
                token       ENCRYPTED { EncodedGeneralToken }
        },
        cryptoSignedToken  SEQUENCE -- General purpose/application specific token
        {
                tokenOID    OBJECT IDENTIFIER,
                token       SIGNED { EncodedGeneralToken }
        },
        cryptoHashedToken SEQUENCE -- General purpose/application specific token
        {
                tokenOID            OBJECT IDENTIFIER,
                hashedVals          ClearToken,
                token HASHED { EncodedGeneralToken }
        },
        cryptoPwdEncr     ENCRYPTED { EncodedPwdCertToken },
        ...
}

-- These allow the passing of session keys within the H.245 OLC structure.
-- They are encoded as standalone ASN.1 and based as an OCTET STRING within H.245
H235Key     ::=CHOICE  -- this is used with the H.245 "h235Key" field
{
        secureChannel           KeyMaterial,
        sharedSecret            ENCRYPTED {EncodedKeySyncMaterial},
        certProtectedKey        SIGNED { EncodedKeySignedMaterial },
        ...
}

KeySignedMaterial ::= SEQUENCE {
        generalId           Identifier, -- slave's alias
        mrandom             RandomVal, -- master's random value
        srandom             RandomVal OPTIONAL, -- slave's random value
        timeStamp           TimeStamp OPTIONAL, -- master's timestamp for unsolicited EU
        encrptval           ENCRYPTED {EncodedKeySyncMaterial }
}
EncodedKeySignedMaterial ::= TYPE-IDENTIFIER.&Type (KeySignedMaterial)

H235CertificateSignature ::= SEQUENCE
{
        certificate             TypedCertificate,
        responseRandom          RandomVal,
        requesterRandom         RandomVal OPTIONAL,
        signature               SIGNED { EncodedReturnSig },
        ...
}

ReturnSig ::= SEQUENCE {
        generalId               Identifier, -- slave's alias
        responseRandom          RandomVal,
        requestRandom           RandomVal OPTIONAL,
        certificate             TypedCertificate OPTIONAL -- requested certificate
}

EncodedReturnSig ::= TYPE-IDENTIFIER.&Type (ReturnSig)
KeySyncMaterial   ::= SEQUENCE
{
        generalID           Identifier,
        keyMaterial         KeyMaterial,
        ...
```

}
EncodedKeySyncMaterial::=TYPE-IDENTIFIER.&Type (KeySyncMaterial)

END *-- End of H235-SECURITY-MESSAGES DEFINITIONS*

ANEXO B

Aspectos específicos de H.323

B.1 Antecedentes

En la figura B.1 se muestra una visión general del alcance de la presente Recomendación en el marco de UIT-T H.323.

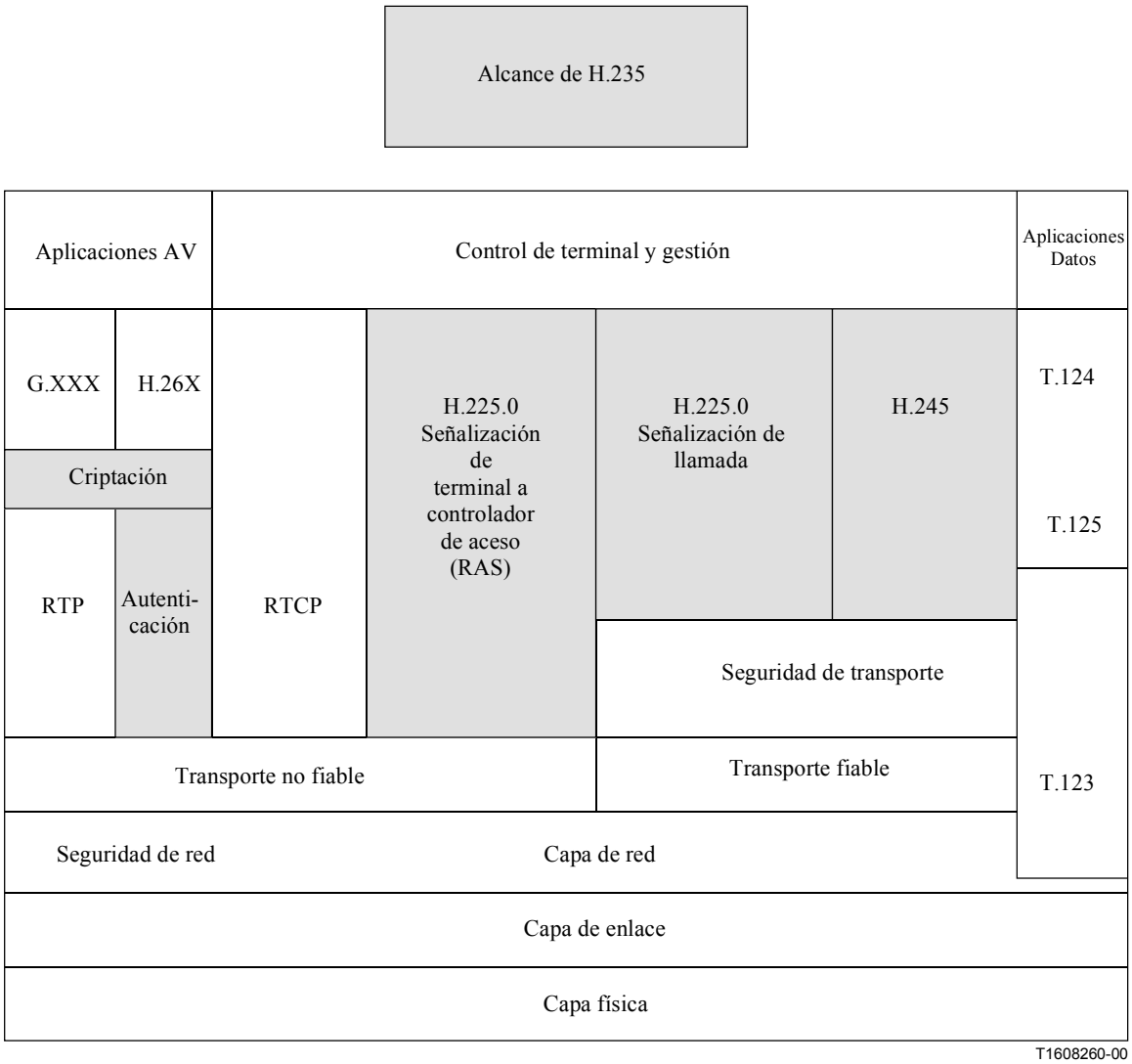


Figura B.1/H.235 – Visión general

Para el protocolo H.323, la señalización del uso de TLS, IPSEC o un mecanismo patentado en el canal de control H.245 se producirá en el canal H.225.0 seguro o inseguro durante el intercambio inicial de mensajes Q.931.

B.2 Señalización y procedimientos

Se aplicarán los procedimientos indicados en la cláusula 8/H.323 (Procedimientos de señalización de la llamada). Los puntos extremos H.323 tendrán la capacidad de codificar y reconocer la presencia (o ausencia) de requisitos de seguridad (para el canal H.245) señalizado en los mensajes H.225.0.

Cuando el propio canal H.225.0 ha de ser asegurado, se seguirán los mismos procedimientos indicados en la cláusula 8/H.323. La diferencia de funcionamiento es que las comunicaciones sólo se producirán después de conectar con el identificador de TSAP y utilizar los modos de seguridad predeterminados (por ejemplo, TLS). Debido a que los mensajes H.225.0 son intercambiados primero cuando se establecen comunicaciones H.323, no puede haber negociaciones de seguridad "dentro de banda" para el canal H.225.0. En otras palabras, ambas partes deben conocer *a priori* que están utilizando un modo de seguridad particular. Para H.323 en IP, se utiliza un puerto bien conocido alternativo (1300) para comunicaciones TLS.

Una finalidad de los intercambios H.225.0 en lo que concierne a su relación con la seguridad H.323, es proporcionar un mecanismo para establecer el canal H.245 seguro. Facultativamente puede haber autenticación durante el intercambio de mensajes H.225.0. Esta autenticación puede estar basada en certificado o en contraseña, utilizando criptación y/troceado (por ejemplo, firma). Los aspectos específicos de estos modos de funcionamiento se describen en 10.2 a 10.3.4.

Un punto extremo H.323 que recibe un mensaje ESTABLECIMIENTO con la **h245SecurityCapability** (**Capacidad seguridad h245**) fijada responderá con el correspondiente **h245SecurityMode** (**Modo de seguridad h245**) aceptable en el mensaje CONEXIÓN. En el caso en que no haya capacidades superpuestas, el terminal llamado puede rechazar la conexión enviando **Release Complete (Liberación completa)** con el código de motivo fijado a ***SecurityDenied*** (***Seguridad denegada***). No se prevé que este error transporte ninguna información sobre cualquier discordancia de seguridad y el terminal llamante tendrá que determinar el problema por algún otro medio. Cuando el terminal llamante recibe un mensaje CONEXIÓN sin un modo de seguridad suficiente o aceptable, puede terminar la llamada con **Release Complete** con el motivo ***SecurityDenied***. Cuando el terminal llamante recibe un mensaje CONEXIÓN sin ninguna capacidad de seguridad, puede terminar la llamada con **Release Complete** con ***undefinedReason*** (***motivo no definido***).

Si el terminal llamante recibe un modo **h245Security** (**Seguridad h245**) aceptable, abrirá y utilizará el canal H.245 en el modo seguro indicado. El hecho de no poder establecer el canal H.245 en el modo seguro determinado se debe considerar como un error de protocolo y la conexión es terminada.

B.2.1 Compatibilidad con la revisión 1

Un punto extremo capaz de seguridad no devolverá ningún campo, indicaciones o estado relacionados con la seguridad al punto extremo que no es capaz de ofrecer seguridad. Si la parte llamada recibe un mensaje ESTABLECIMIENTO que no contiene capacidades y/o testigo de autenticación **H245Security**, puede devolver **ReleaseComplete** para rechazar la conexión, pero en este caso utilizará el código ***UndefinedReason***. De manera correspondiente, si una parte llamante recibe un mensaje CONEXIÓN sin **H245SecurityMode** (**Modo seguridad H245**) y/o testigo de autenticación habiendo enviado un mensaje ESTABLECIMIENTO con **H245Security** y/o testigo de autenticación, puede también terminar la conexión emitiendo un mensaje **ReleaseComplete** con un código ***UndefinedReason***.

B.3 Aspectos relativos a RTP/RTCP

La utilización de criptación en el tren RTP seguirá la metodología general recomendada en el documento referenciado en [RTP]. La criptación de los medios se producirá de manera independiente, paquete por paquete[1]. El encabezamiento RTP (incluido el encabezamiento de cabida útil) no será criptado. La sincronización de nuevas claves y textos criptados se basa en el tipo de cabida útil dinámica.

La clave de criptación inicial es presentada por el terminal director junto con el número de cabida útil dinámica (mediante **EncryptionSync** en UIT-T H.245). El receptor o receptores del tren de medios comenzará el uso inicial de la clave al recibir el número de esta cabida útil en el encabezamiento RTP. El punto extremo director puede distribuir nuevas claves en cualquier momento. La sincronización de la clave más nueva con el tren de medios será indicada por el cambio del tipo de cabida útil a un nuevo valor dinámico. Obsérvese que los valores específicos no tienen importancia, mientras cambien para cada nueva clave que se distribuye.

Se supone que esta criptación se aplica sólo a la cabida útil en cada paquete RTP, los encabezamientos RTP permanecen en claro. Se supone que todos los paquetes RTP deben ser un múltiplo de octetos completos. El modo de encapsular los paquetes RTP en la capa de transporte o de red no es pertinente a la presente Recomendación. Todos los modos deben tener en cuenta los paquetes perdidos (o fuera de secuencia), además del relleno de paquetes a un múltiplo de octetos apropiado.

El descifrado del tren debe ser independiente con respecto a los paquetes que se puedan perder, cada paquete es descifrable por sí mismo. Dos requisitos del modo algoritmo de bloque funcionarán como sigue:

a) *Vectores de inicialización*

La mayor parte de los modos de bloque conllevan algún "encadenamiento"; cada ciclo de criptación depende en cierta manera de la salida del ciclo anterior. Por consiguiente, al comienzo de un paquete, se debe proporcionar algún valor de bloque inicial [generalmente denominado un vector de inicialización (IV, *initialization vector*)] para comenzar el proceso de criptación. Con independencia del número de octetos de tren que son procesados en cada ciclo de criptación, la longitud de IV es siempre igual a la longitud de un bloque. Todos los modos, salvo el modo libro de código electrónico (ECB, *electronic code book*) requieren un IV. En todos los casos, el IV se construirá a partir de los primeros octetos de B (donde B es el tamaño de bloque): (Seq# + indicación de tiempo). Este esquema se debe repetir hasta que se hayan generado octetos suficientes. Cabe señalar que el IV generado de esta manera puede producir un esquema de clave que se considera "débil" para un algoritmo determinado.

b) *Relleno*

Los modos ECB y CBC procesan siempre el tren de entrada un bloque cada vez y mientras CFB y OFB pueden procesar la entrada en cualquier número de octetos, N (≤ B), se recomienda que N = B.

Se dispone de dos métodos para tratar paquetes cuya cabida útil no es un múltiplo de bloques:

1) Apropiación de texto cifrado para ECB y CBC; relleno de ceros para CFB y OFB.
2) Relleno de la manera prescrita por [RTP, sección 5.1].

---

[1] Debe señalarse que si el tamaño del paquete RTP es superior al tamaño de MTU, la pérdida (o fragmento (parcial) hará que el paquete RTP completo sea indescifrable.

[RTP, sección 5.1] describe un método de relleno en el cual la cabida útil es rellenada hasta un múltiplo de bloque, el último octeto es fijado con el número de octetos de relleno (incluido el último), y el bit P fijado en el encabezamiento RTP. El valor de relleno debe ser determinado por el convenio normal del algoritmo de cifrado.

Todas las implementaciones H.235 soportarán ambos esquemas. El esquema en uso puede ser deducido como sigue: si el bit P está fijado en el encabezamiento RTP, el paquete tiene relleno. Si el paquete no es un múltiplo de B y el bit P no está fijado, se aplica el apropiación de texto cifrado, en los demás casos el paquete es un múltiplo de B, y no se aplica relleno.

La integridad y protección contra reproducción del tren RTP queda en estudio.

La aplicación de técnicas criptográficas a los elementos RTCP requiere también estudio.

B.4 Señalización RAS/procedimientos de autenticación

B.4.1 Introducción

Este anexo no proporciona explícitamente ninguna forma de privacidad de mensajes entre controladores de acceso y puntos extremos. Se pueden utilizar dos tipos de autenticación. El primer tipo es la criptación simétrica que no requiere contacto previo entre el punto extremo y el controlador de acceso. El segundo tipo es el abono que tendrá dos formas, contraseña o certificado. Todas estas formas se derivan de los procedimientos indicados en 10.1, 10.2.2, 10.2.3 y 10.2.4. En este anexo, las etiquetas genéricas (EPA y EPB) utilizadas en las cláusulas mencionadas representarán respectivamente al punto extremo y al controlador de acceso.

B.4.2 Autenticación de punto extremo – controlador de acceso (no basada en abono)

Este mecanismo puede proporcionar al controlador de acceso un enlace criptográfico que un punto extremo determinado registrado previamente, es el mismo que emite los subsiguientes mensajes RAS. Cabe señalar que esto no puede proporcionar ninguna autenticación del controlador de acceso al punto extremo, a menos que se incluya el elemento de firma facultativo. El establecimiento de la relación de identidad se produce cuando el terminal emite **GRQ** como se indica en 7.2.1/H.323. El intercambio Diffie-Hellman se producirá junto con los mensajes **GRQ** y **GCF** como se indica en la primera fase de 10.1. Esta clave secreta compartida será utilizada en cualquier **RRQ/URQ** subsiguiente del terminal al controlador de acceso. Si un controlador de acceso funciona en este modo y recibe **GRQ** sin un testigo que contiene *DHset* o un valor de algoritmo aceptable, devolverá un código de motivo **securityDenial** (**denegación seguridad**) en el **DRJ**.

La clave secreta compartida Diffie-Hellman creada durante el intercambio **GRQ/GCF** se puede utilizar para autenticación en los siguientes mensajes **xRQ**. Se aplicarán los siguientes procedimientos para completar este modo de autenticación.

Terminal (**xRQ**):

1) El terminal proporcionará toda la información en el mensaje como se describe en las cláusulas pertinentes de UIT-T H.225.0.
2) El terminal criptará **GatekeeperIdentifier** (Identificador de controlador de acceso) (devuelto en el **GCF**) utilizando la clave secreta compartida negociada. Ésta será transferida en un **clearToken** (**testigo claro**) (véase 10.2) como el **generalID**.

Los 16 bits del **random** (**aleatorio**) y después la **requestSeqNum** (**petición número secuencia**) se pondrán a XOR con cada 16 bits del **GatekeeperIdentifier**. Si **GatekeeperIdentifier** no termina en una frontera 16 par, los últimos 8 bits del **GatekeeperIdentifier** se pondrán a XOR con el octeto menos significativo del valor aleatorio y después **requestSeqNum**. El **GatekeeperIdentifier** será criptado utilizando el algoritmo seleccionado en **GCF** (algorithmOID) y utilizando todo el secreto compartido.

El ejemplo a continuación ilustra este procedimiento:

RND16: valor de 16 bits del valor aleatorio

SQN16: valor de 16 bits de requestSeqNum

BMPX: el carácter BMP X-ésimo GatekeeperIdentifier

BMP1' = (BMP1) XOR (RND16) XOR (SQN16)

BMP2' = (BMP2) XOR (RND16) XOR (SQN16)

BMP3' = (BMP3) XOR (RND16) XOR (SQN16)

BMP4' = (BMP4) XOR (RND16) XOR (SQN16)

BMP5' = (BMP5) XOR (RND16) XOR (SQN16)

:

:

BMPn' = (BMPn) XOR (RND16) XOR (SQN16)

Para enlazar criptográficamente esto y los mensajes siguientes con el registrador original (el punto extremo que emitió **RRQ**), se utilizará el valor **random** más reciente (este valor puede ser uno más nuevo que el valor devuelto en **RCF**, de un ulterior mensaje **xCF**).

Controlador de acceso (**xCF/xRJ**):

1) El controlador de acceso criptará su **GatekeeperIdentifier** (según el procedimiento anterior) con la clave secreta compartida asociada con el punto extremo alias y comparará esto con el valor en **xRQ**.
2) El controlador de acceso devolverá **xRJ** si los dos valores criptados no concuerdan.
3) Si el **GatekeeperIdentifier** concuerda, el controlador de acceso aplicará cualquier lógica local y responderá con **xCF** o **xRJ**.
4) Si **xCF** es enviado por el controlador de acceso, debe contener un **EndpointIdentifier** (**Identificador de punto extremo**) asignado y un nuevo valor aleatorio en el campo **random** de un **clearToken**.

Véase la segunda fase de la figura 1 para una representación gráfica de este intercambio. El controlador de acceso sabe la clave secreta compartida que ha de utilizar para descifrar el identificador de controlador de acceso mediante el nombre alias en el mensaje.

B.4.3 Autenticación de punto extremo – controlador de acceso (basada en abono)

Todos los mensajes RAS que no sean GRQ/GCF deben contener los testigos de autenticación requeridos por el modo de funcionamiento específico. Hay tres variaciones diferentes que se pueden aplicar según las necesidades y el entorno:

1) Contraseña con criptación simétrica.
2) Contraseña con troceado.
3) Certificado con firmas.

En todos los casos el testigo contendrá la información descrita en las siguientes subcláusulas de acuerdo con la variación elegida. Si un controlador de acceso funciona en un modo seguro y recibe un mensaje RAS sin un valor de testigo aceptable, devolverá un código de motivo **securityDenial** en el mensaje de rechazo. En todos los casos, el testigo devuelto del controlador de acceso es facultativo; si se omite, sólo se logra la autenticación unidireccional.

B.4.3.1 Contraseña con criptación simétrica

La fase de descubrimiento del controlador de acceso (GRQ, GCF y GRJ) puede ser insegura tal como se muestra en la figura B.2 o segura usando para ello los **cryptoTokens**.

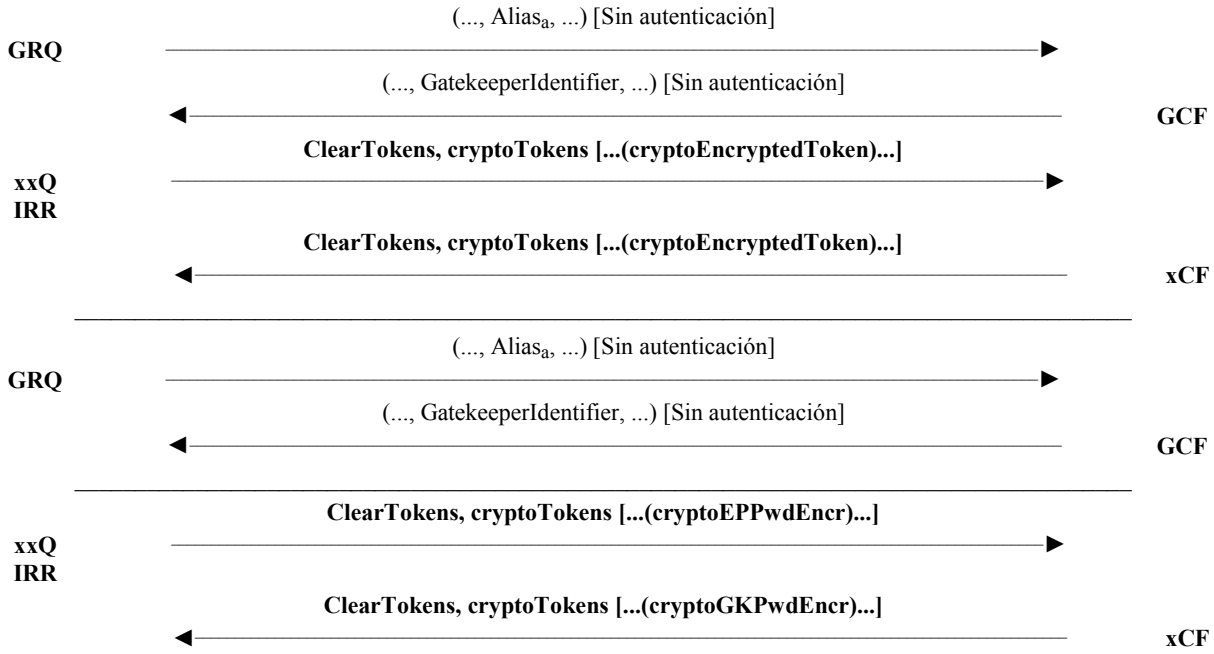


Figura B.2/H.235 – Contraseña con criptación simétrica

B.4.3.2 Contraseña con troceado

La fase de descubrimiento del controlador de acceso (GRQ, GCF y GRJ) puede ser insegura tal como se muestra en la figura B.3 o segura de acuerdo con el anexo D usando para ello los **cryptoTokens**.

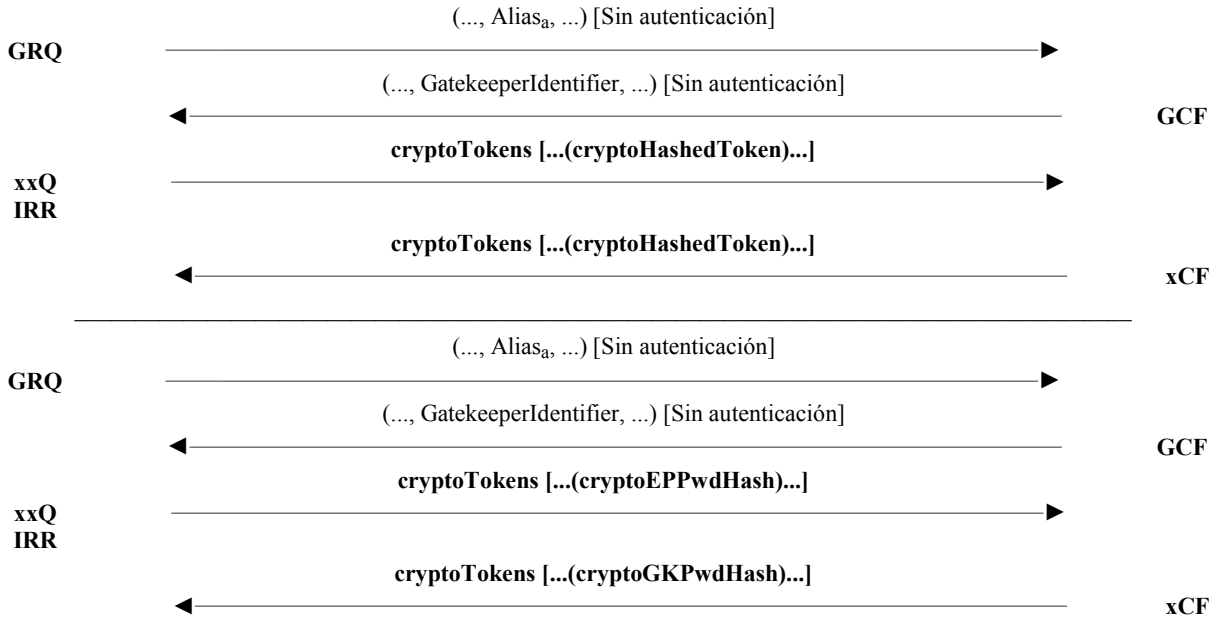


Figura B.3/H.235 – Contraseña con troceado

B.4.3.3 Certificado con firmas

La fase de descubrimiento del controlador de acceso (GRQ, GCF y GRJ) puede ser insegura como se muestra en la figura B.4 o segura de acuerdo con el anexo E usando para ello los **cryptoTokens**.

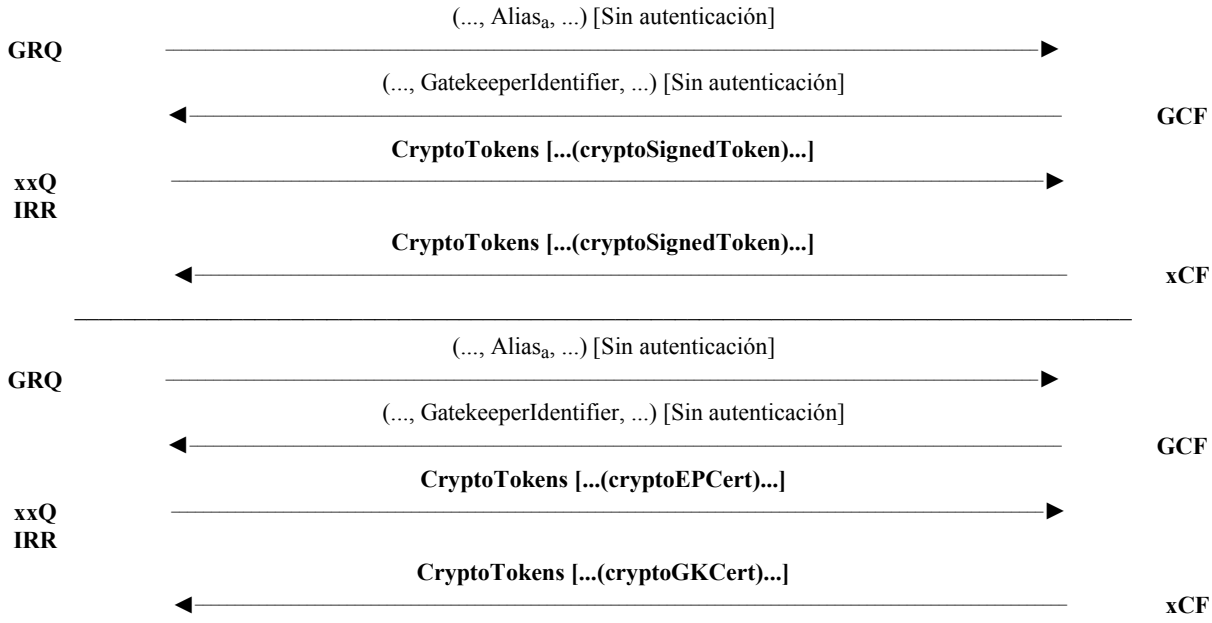


Figura B.4/H.235 – Certificado con firmas

B.5 Interacciones no relacionadas con terminales

B.5.1 Pasarela

Como se indica en 6.6, se debe considerar que una pasarela H.323 es un elemento de confianza. Esto incluye pasarelas de protocolo (H.323-H.320, etc.) y cabeceras de seguridad (apoderados/cortafuegos). La privacidad de los medios puede ser asegurada entre el punto de extremo y el dispositivo de pasarela comunicante, pero lo que se produce en el extremo distante de la pasarela se debe considerar inseguro por defecto.

ANEXO C

Aspectos específicos del protocolo H.324

Queda en estudio.

ANEXO D

Perfil de seguridad básico

D.1 Introducción

En este anexo se describen perfiles de seguridad básicos simples. Los perfiles de seguridad especificados se basan en los perfiles de UIT-T H.235, los perfiles ETSI y los perfiles IMTC. Los perfiles de seguridad seleccionan las características de seguridad apropiadas a partir de UIT-T H.235 con su rico conjunto de opciones.

D.2 Convenios de especificación

Para la comprensión de los términos utilizados en este anexo, son necesarias algunas explicaciones:

Este anexo define el **perfil de seguridad básico**. El perfil de seguridad básico proporciona la seguridad básica por medios sencillos que utilizan técnicas criptográficas seguras basadas en contraseñas. El perfil de seguridad básico puede utilizar el **perfil de seguridad de criptación vocal** para lograr la confidencialidad de la voz en caso necesario. En el anexo E, puede verse un perfil de seguridad más perfeccionado que aplica las firmas digitales y supera las limitaciones del perfil de seguridad básico.

Este anexo utiliza los campos H.235 para la provisión de servicios de seguridad de autenticación/integridad en mensajes de señalización H.323. Diferentes identificadores de objeto (véase D.11) determinan la seguridad de servicio realmente seleccionada y la versión de protocolo de la presente Recomendación que se está utilizando. El procedimiento I) especifica el modo de implementar los servicios de seguridad mediante determinados mecanismos de seguridad, como las técnicas simétricas (troceado codificado). A lo largo del texto se hace referencia a los identificadores de objeto mediante un símbolo (por ejemplo, "A").

Si bien el servicio de integridad de mensajes siempre proporciona, además, la autenticación de los mismos, la inversa no siempre es cierta. En la práctica, el servicio combinado de autenticación e integridad explota el mismo material de claves sin que haga más débil la seguridad.

Además, toda la información de seguridad salto por salto es introducida en el elemento **CryptoHashedToken**. Esta información es recalculada en cada salto.

Por regla general, la contraseña, la clave de sesión y el secreto compartido tienen en común que todos son utilizados en la criptografía simétrica entre dos (o más) entidades. La diferencia entre una contraseña y un clave de sesión/secreto compartido es el modo en que las claves son aplicadas realmente, por ejemplo, contraseñas para la autenticación y la autorización, claves de sesión para la criptación. El término secreto compartido es tan indeterminado que de hecho no se refiere a ninguna utilización especifica.

La **contraseña** (que también puede ser contemplada como un secreto compartido) es utilizada para la autenticación/integridad de mensajes RAS y H.225.0, puesto que este elemento puede ser introducido por el usuario. La contraseña tiene normalmente un tiempo de vida largo; la contraseña se conoce *a priori* y puede ser definida como parte del proceso global de abono del usuario. Algunos algoritmos (por ejemplo, la canalización de la contraseña a través de un algoritmo de troceado) pueden transformar la contraseña para un procesamiento más conveniente en los protocolos a fin de que tenga una longitud fija.

La **clave de sesión** para la criptación de trenes de medios es, por otra parte, generada por el terminal director exclusivamente para una sesión RTP específica (en una OLC); como máximo para una llamada. La clave de sesión generada es criptada con una clave que se deriva del **secreto compartido** Diffie-Hellman convenido que han calculado ambos puntos extremos. En este caso, el secreto compartido Diffie-Hellman actúa como una clave maestra para la protección de la(s) clave(s) de sesión.

El **ClearToken** (**testigo claro**) H.235 ofrece un campo denominado **random** que contiene un entero de 32 bits. Este campo es inutilizado en el siguiente sentido: **random** es realmente un número monotónicamente creciente que arranca en un valor cualquiera y se incrementa con cada mensaje saliente. El campo **random** se utiliza como un valor de "aleatorización" adicional a la entrada de la función de troceado cifrado en el caso de que se envíen varios mensajes uno inmediatamente después de otro, que transportan sin embargo identificaciones de tiempo idénticas. Esto puede suceder cuando el reloj UTC no proporciona una resolución de reloj suficiente. En esencia, el valor de troceado producido o el valor de comprobación de la integridad parecen diferentes debido al cambio del valor de **random**. Se trata de un contrarrestar los ataques de reproducción. Para simplificar la implementación, aquí se prefiere un contador creciente que una secuencia verdaderamente aleatoria.

El recipiente puede guardar las parejas **timestamp/random** recibidas durante el periodo definido por una ventana[2] de tiempo local. Se puede identificar un ataque de reproducción cuando la misma pareja **timestamp/random** ocurre dos veces.

Este perfil define "fijar el **generalID** en el **ClearToken** al identificador del recipiente". Esto de hecho significa que, para los mensajes RAS, este identificador es el identificador[3] de punto extremo o del GK, y para los mensajes de señalización de llamada H.225.0 es el identificador de punto extremo llamado. El **sendersID** deberá ser fijado a la cadena de identificación del emisor.

Un **block** (**bloque**) se refiere a la unidad básica de bits empaquetados que el descifrador de bloques es capaz de criptar/descriptar en una operación criptográfica elemental; para la DES (norma de criptación de datos) y la DES triple el tamaño del bloques es de 64 bits.

Para evitar referencias a una marca registrada (RC2®), este anexo hace referencia en realidad a un algoritmo de criptación "compatible con RC2").

Esta Recomendación utiliza términos relativos a la seguridad muy conocidos como clave, gestión de claves y SET, que tienen significados distintos en otros contextos (por ejemplo, "key pad", gestión de claves de características Q.931/Q.932 y protocolo de transacciones electrónicas seguras).

D.3 Alcance

Este anexo describe la seguridad simple para entidades H.323. El perfil de seguridad puede ser aplicado por terminales H.323 seguros, incluido el **terminal telefónico simple seguro** [Tipo de punto extremo de audio simple de seguridad (SASET)] – definido en este anexo (véase D.6); el perfil de seguridad puede ser aplicado también por otras entidades H.323, como las pasarelas, los controladores de acceso y las MCU.

D.4 Abreviaturas

BES Servicio fuera del terminal (*back-end service*)

CBC Concatenación de bloques cifrados (*cipher block chaining*)

DES Norma de criptación de datos (*data encryption standard*)

DH Diffie-Hellman

ECB Libro de código electrónico (*electronic code book*)

EP Punto extremo (*endpoint)*

ETSI Instituto Europeo de Normas de las Telecomunicaciones (*European Telecommunications Standards Institute*)

GK Controlador de acceso (*gatekeeper*)

HMAC Código de autenticación de mensaje troceado (*hashed message authentication code*)

IMTC Consorcio de teleconferencias multimedios internacionales (*international multimedia teleconferencing consortium*)

IPSEC Seguridad del protocolo Internet (*Internet protocol security*)

IV Vector de inicialización (*initialization vector*)

MAC Código de autenticación de mensaje (*message authentication code*)

MD5 Message Digest 5 (*message digest 5*)

---

[2] La ventana de tiempo compensa las variaciones del tiempo sincronizado y el retardo de tránsito de la red.

[3] Dependiendo de que la dirección sea de EP a GK, o viceversa.

OID Identificador de objeto (*object identifier*)

PFS Secreto perfecto hacia delante (*perfect forward secrecy*)

RAS Registro, admisión y situación (*registration, admission and status*)

RSA Rivest, Shamir y Adleman

RTP Protocolo en tiempo real (*real-time protocol*)

SASET Tipo de punto extremo de audio simple de seguridad (*secure audio simple endpoint type*)

SET Tipo de punto extremo simple (*simple endpoint type*)

SHA Algoritmo troceado asegurado (*secure hash algorithm*)

TCP Protocolo de control de transmisión (*transmission control protocol*)

TIPHON Armonización de telecomunicaciones y protocolo Internet por las redes (*telecommunications and Internet protocol harmonization over networks*)

TLS Seguridad de capa de transporte (*transport layer security*)

UIT Unión Internacional de Telecomunicaciones

VoIP Voz sobre el protocolo Internet (*voice over Internet protocol*)

D.5 Referencias normativas

DES [FIPS-46-2] US National Bureau of Standards, "Data Encryption Standard", Federal Information Processing Standard, (FIPS) Publication 46-2, diciembre de 1993, http://www.itl.nist.gov/div897/pubs/fip46-2.htm.

[FIPS-74] US National Bureau of Standards, "Guidelines for Implementing and Using the Data Encryption Standard", Federal Information Processing Standard (FIPS) Publication 74, abril de 1981, http://www.itl.nist.gov/div897/pubs/fip74.htm.

[FIPS-81] US National Bureau of Standards, "DES Modes of Operation", Federal Information Processing Standard (FIPS) Publication 81, diciembre de 1980, http://www.itl.nist.gov/div897/pubs/fip81.htm.

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

[H.225.0] UIT-T H.225.0, Versión 2, *Protocolos de señalización de llamadas y paquetización de trenes de medios para sistemas de comunicación multimedios por paquetes*, 1998.

[H.235v1] UIT-T H.235, Versión 1, *Seguridad y criptado para terminales multimedios de la serie H (basados en las Recomendaciones H.323 y H.245)*, 1998.

[H.235v2] UIT-T H.235, Versión 2, *Seguridad y criptado para terminales multimedios de la serie H (basados en las Recomendaciones H.323 y H.245)*, 2000.

[H.245] UIT-T H.245, Versión 7, *Protocolo de control para comunicación multimedios*, 2000.

[H.323] UIT-T H.323, Versión 4, *Sistemas de comunicación multimedios basados en paquetes*, 2000.

[H.323, Anexo F] UIT-T H.323, anexo F, *Tipos de punto extremo simples*, 1999.

[RFC 2268] RIVEST (R.): *A Description of the RC2® Encryption Algorithm*, RFC 2268, marzo de 1998.

D.6 Perfil de seguridad básico

Esta cláusula describe una línea básica para el perfil de seguridad simple.

D.6.1 Visión general

El perfil de seguridad básico gobierna el modelo con encaminamiento por controlador de acceso. La seguridad básica es aplicable en los entornos administrados con contraseñas/claves simétricas asignadas entre las entidades (terminal – controlador de acceso, controlador de acceso – controlador de acceso, pasarela – controlador de acceso).

Las características proporcionadas por estos perfiles incluyen:

– Para mensajes RAS, H.225.0 y H.245:
  - La autenticación de usuario a una entidad deseada con independencia del número de saltos[4] del nivel de aplicación que atraviesa el mensaje.
  - La integridad del propio mensaje de señalización, incluidas las porciones (campos) críticas de los mensajes que llegan a una entidad, con independencia del número de saltos del nivel de aplicación que atraviesa el mensaje.
  - La autenticación e integridad del mensaje de señalización salto por salto del nivel de aplicación proporciona estos servicios de seguridad para el mensaje completo.

– Para el tren de medios:
  - La confidencialidad del tren de medios es proporcionada por criptación simétrica.

Mediante la provisión, de manera adecuada, de los servicios de seguridad anteriores se consigue frustrar varios ataques. Estos incluyen:

- Los ataques de denegación de servicio: una comprobación rápida de los valores del troceado criptográfico puede evitar tales ataques.
- Ataques "intermedios": la autenticación e integridad de los mensajes salto por salto en el nivel de aplicación previene contra tales ataques cuando el ataque intermedio se produce en un salto del nivel de aplicación, es decir un encaminador hostil.
- Ataques de reproducción: estos ataques se evitan mediante el empleo de indicaciones de tiempo y números secuenciales.
- Engaño: la autenticación del usuario evita tales ataques.
- Asalto a la conexión: La autenticación/integridad de cada mensaje de señalización evita tales ataques.
- La escucha furtiva del tren de medios es contrarrestada mediante la criptación y el uso de claves secretas.

Otros puntos destacados del perfil de seguridad simple incluyen:

- La utilización de algoritmos robustos, bien conocidos y ampliamente desplegados basados en material IMTC/ETSI/IETF.
- La capacidad de un despliegue por fases basado en el requisito de seguridad del modelo comercial.

---

[4] Salto tiene aquí el sentido de un elemento de red H.235 de confianza (por ejemplo, GK, GW, MCU, apoderado, cortafuegos). Por tanto, la seguridad salto por salto en el nivel de aplicación cuando se utiliza con técnicas simétricas no proporciona una verdadera seguridad de extremo a extremo entre terminales.

- Su aplicabilidad en distintos escenarios de despliegue, tales como los grupos cerrados, los entornos escalables y las conferencia multipunto.

En el cuadro D.1 se resumen los procedimientos definidos en este anexo por los perfiles de seguridad para satisfacer los diferentes requisitos de seguridad. El cuadro incluye el perfil de seguridad básico (sombreado vertical – azul en la copia electrónica) y el perfil de seguridad de criptación vocal (sombreado horizontal – verde en la copia electrónica).

Cuadro D.1/H.235 – Resumen de los perfiles de seguridad del anexo D

Servicios de seguridad	Funciones de llamada							
	RAS	H.225.0		H.245[a)]	RTP		
Autenticación	Contraseña HMAC-SHA1-96	Contraseña HMAC-SHA1-96		Contraseña HMAC-SHA1-96			
No repudio							
Integridad	Contraseña HMAC-SHA1-96	Contraseña HMAC-SHA1-96		Contraseña HMAC-SHA1-96			
Confidencialidad					DES de 56 bits	compatible con RC2 de 56 bits	DES triple de 168 bits
Control de acceso							
Gestión de claves	Asignación de contraseña basada en abono	Asignación de contraseña basada en abono	Inter-cambio de claves Diffie-Hellman autentica-das	Gestión de claves de sesión H.235 integrada (distribución de claves, actualización de claves utilizando DES de 56 bits/compatible con RC2 de 56 bits/DES triple de 168 bits)			

a) H.245 tunelizada o H.245 insertada en una conexión rápida H.225.0.

Para la autenticación, el usuario deberá utilizar un esquema basado en contraseñas. El esquema basado en contraseñas se recomienda decididamente para la autenticación debido a su simplicidad y facilidad de implementación. El troceado de todos los campos en los mensajes H.225.0 es el enfoque recomendado para la integridad de los mensajes (también cuando se utiliza el esquema de contraseñas).

Las entidades H.323 seguras que disponen de este perfil de seguridad verifica la autenticación junto con la integridad utilizando el mismo mecanismos de seguridad común.

Para el caso de la confidencialidad de la voz facultativa, se propone un esquema de criptación que utilice compatible con RC2, DES o DES triple basado en el modelo comercial y en el requisito de exportabilidad. Algunos entornos que ya está ofreciendo cierto grado de confidencialidad posiblemente no necesiten la criptación vocal. En este caso, tampoco será necesario el convenio de claves Diffie-Hellman y otros procedimientos de gestión de claves.

Cuando las entidades H.323 despliegan el perfil de seguridad de criptación vocal, deberán implementar la norma DES de 56 bits como algoritmo de criptación por defecto; Las entidades pueden implementar la norma DES triple de 168 bits, aunque pueden implementar la criptación exportable utilizando la compatible con RC2 de 56 bits.

Los métodos de control de acceso no se describen explícitamente; estos métodos se pueden implementar localmente tras la recepción de la información transportada en los campos de señalización H.235 (ClearToken, CryptoToken).

La presente Recomendación no describe los procedimientos para la asignación de claves secretas/contraseñas basada en abono y su gestión y administración. Tales procedimientos pueden aplicarse en otros métodos que no forman parte de este anexo.

Las entidades de comunicación involucradas son capaces de determinar implícitamente la utilización, bien del perfil de seguridad básico o bien del perfil de seguridad de firmas mediante la evaluación de los identificadores de objeto de seguridad señalados de los mensajes (**tokenOID**, y **algorithmOID**; véase también D.11).

D.6.1.1 Perfil de seguridad básico

El perfil de seguridad básico es aplicable en un entorno en el cual se pueden asignar claves simétricas/contraseñas suscritas a las entidades H.323 aseguradas (terminales, ...) y elementos de red (GKs, apoderados). El perfil proporciona la autenticación e integridad del mensaje RAS, H.225.0 y H.245 tunelizado utilizando el troceado HMAC-SHA1-96 basado en contraseñas especificado por el procedimiento I. El establecimiento de comunicación de H.225.0 utilizando FastStart (GK a GK o terminal a terminal) incluye la gestión de claves integrada de Diffie-Hellman.

La zona de sombreado vertical (azul en la copia electrónica) del cuadro D.2 representa el perfil de seguridad básico.

Cuadro D.2/H.235 – Perfil de seguridad básico

Servicios de seguridad	Funciones de llamada			
	RAS	H.225.0	H.245	RTP
Autenticación e integridad	Contraseña HMAC-SHA1-96	Contraseña HMAC-SHA1-96	Contraseña HMAC-SHA1-96	
No repudio				
Confidencialidad				
Control de acceso				
Gestión de claves	Asignación de contraseña basada en abono	Asignación de contraseña basada en abono		

Facultativamente, el perfil de seguridad de criptación vocal puede combinarse suavemente con el perfil de seguridad básico. Los trenes de audio pueden ser criptados mediante el perfil de seguridad de criptación vocal desplegando la norma DES, compatible con RC2 o DES triple y utilizando el procedimiento de intercambio de claves Diffie-Hellman autenticado.

El perfil de seguridad básico ordena el procedimiento de conexión rápida con elementos de gestión de claves integrados. Los medios de señalización son proporcionados también para la sincronización y actualización de claves H.245 tunelizadas. Para llamadas de larga duración, estos mensajes requieren la tunelización de H.245 dentro de los mensajes H.225.0.

D.6.1.2 Perfil de seguridad de criptación vocal

El perfil de seguridad de criptación vocal no es un perfil independiente como el perfil de seguridad básico. Es más bien una opción del perfil de seguridad básico y se puede utilizar junto con él. Este perfil también depende de ciertos servicios de seguridad como parte de los procedimientos de señalización de llamada y de establecimiento de la conexión; por ejemplo, el convenio de claves Diffie-Hellman y otras funciones de gestión de claves.

Las entidades H.323 pueden implementar el perfil de seguridad de criptación vocal para conseguir la confidencialidad de la conversación. Se ofrecen a tal fin tres algoritmos de criptación: el esquema propuesto consiste en la criptación que utiliza la norma compatible con RC2, la DES o la DES triple basada en el modelo comercial y el requisito de exportabilidad. Algunos entornos que ofrecen ya un cierto grado de confidencialidad no necesitarán posiblemente la criptación vocal. En este caso, tampoco se necesita el convenio de claves Diffie-Hellman y otros procedimientos de gestión de claves.

Cuando despliegan el perfil de seguridad de criptación vocal, las entidades H.323 deberán implementar la DES de 56 bits como algoritmo de criptación por defecto. Las entidades pueden implementar la DES triple de 168 bits, si bien pueden implementar la criptación exportable utilizando la compatible con RC2 de 56 bits.

TEl perfil de seguridad de criptación vocal se especifica en la cláusula D.2.

Cuadro D.3/H.235 – Perfil de criptación vocal

Servicios de seguridad	Funciones de llamada					
	RAS	H.225.0	H.245	RTP		
Autenticación e integridad						
No repudio						
Confidencialidad				DES de 56 bits	compatible con RC2 de 56 bits	DES triple de 168 bits
Control de acceso						
Gestión de claves		intercambio de claves Diffie-Hellman autenticadas	Gestión de claves de sesión H.235 integrada (distribución de claves, actualización de claves)			

D.6.2 Autenticación e integridad

Este anexo utiliza los términos que siguen para la provisión de los servicios de seguridad.

- **Autenticación e integridad**: Servicio de seguridad combinado, parte del perfil de seguridad básico, que soporta la integridad de los mensajes junto con la autenticación del usuario. El usuario se puede autenticar aplicando correctamente un clave secreta compartida. Ambos servicios de seguridad son proporcionados por el mismo mecanismo de seguridad.

Cuando se utilizan técnicas de claves simétricas, la autenticación/integridad de los servicios de seguridad se aplican solamente salto por salto.

D.6.3 Requisitos H.323

Se supone que las entidades H.323 que implementan este perfil de seguridad básico soportan las siguientes características H.323:

- Conexión rápida
- Modelo con encaminamiento por controlador de acceso.

D.6.3.1 Sinopsis

Describimos el siguiente procedimiento para su utilización en este perfil.

El procedimiento I es un mecanismo de autenticación de mensajes de señalización basado en claves simétricas simples que utiliza una contraseña compartida por dos entidades (por ejemplo, controlador de acceso y punto extremo H.323). Este procedimiento proporciona la autenticación e integridad de los mensajes RAS, Q931 y H.245 (véase D.6.3.2).

Dependiendo de la política de seguridad, la autenticación puede ser unilateral, o mutua (recíproca) si se aplica también la autenticación/integridad en el sentido inverso y se proporciona por tanto una seguridad mayor. El controlador de acceso decide si se aplica también la autenticación/integridad en el sentido inverso.

Los controladores de acceso que detectan que ha fallado la autenticación y/o que ha fallado la validación de la integridad en un mensaje de señalización de llamada o RAS recibido de un punto extremo asegurado o controlador de acceso par, responde con un mensaje de rechazo que señala el fallo de seguridad fijando el motivo del rechazo a **securityDenial**.

Existe una señalización H.235 implícita para indicar el uso del procedimiento I y el mecanismo de seguridad aplicado que se basa en el valor de los identificadores de objeto (véase también D.11) y los campos de mensaje rellenos.

Ese perfil no utiliza los campos ICV H.235; en su lugar, los valores criptográficos de comprobación de la integridad son tratados como valores de troceado criptográfico e introducidos en los campos de troceado **CryptoToken**.

D.6.3.2 Detalles de la autenticación de mensajes señalización basada en claves simétricas (Procedimiento I)

Cuando se emplea el procedimiento I deberán seguirse los pasos a continuación:

- Se utiliza un valor de troceado de 12 bytes (96 bits) con los algoritmos HMAC-SHA1-96 para la generación del autenticador. Si la clave es generada a partir de una contraseña, *deberá* utilizarse el mecanismo descrito en 10.3.5, para el cálculo de dicha clave derivada de la contraseña.

  NOTA 1 – Cuando la clave secreta se deriva de una contraseña introducida por el usuario se debe tener cuidado de garantizar una aleatorización suficiente. Se recomienda, por ejemplo, utilizar secretos verdaderamente aleatorios para la clave secreta, o para garantizar que las contraseñas son suficientemente largas.

- El campo **CryptoH323Token** en cada mensaje RAS/H.225.0 deberá contener los siguientes campos:
  - **nestedCryptoToken** conteniendo un **CryptoToken** que a su vez contiene el **cryptoHashedToken** que contiene los campos siguientes:
    - **tokenOID** puesto a: "A", indicando que el cálculo de la autenticación/integridad incluye todos los campos del mensaje RAS/H.225.0.
    - **hashedVals**, que contiene el campo **ClearToken** utilizado con los siguientes campos:
      - **tokenOID** puesto a: "T", indicando que se está utilizando el **ClearToken** para la autenticación/integridad del mensaje.
      - t**imeStamp** que contiene la indicación de tiempo.
      - **random**, que contiene un número secuencial monotónicamente creciente. Este número permite confeccionar dos mensajes con la misma indicación de tiempo única (dentro de la resolución de reloj).
      - **generalID**, que contiene el identificador del recipiente (sólo en el caso de mensajes unidifusión).
      - **sendersID**, que contiene el identificador del emisor.

– **dhkey**, utilizado para pasar los parámetros Diffie-Hellman especificados en esta Recomendación durante **Setup** y **Connect**.
  • **halfkey**, que contiene la clave pública aleatoria de una parte.
  • **modsize**, que contiene el número primo DH (véase el cuadro D.4).
  • **generator**, que contiene el grupo Diffie-Hellman (véase el cuadro D.4).

NOTA 2 – Cuando el perfil de seguridad básico se utiliza sin el perfil de seguridad de criptación vocal, no han de enviarse entonces parámetros Diffie-Hellman; en su lugar, los **halfkey**, **modsize** y **generator** pueden fijarse a la representación binaria de 0 por simplicidad.

– **token**, que contiene **HASHED** con los campos:
  • **algorithmOID** puesto a "U", indicando el uso de HMAC-SHA1-96.
  • **params** puesto a NULO (NULL).
  • **hash**, conteniendo el autenticador calculado utilizando HMAC-SHA1-96. El autenticador puede ser calculado sobre
    – todos los campos RAS/H.225.0 del mensaje si el **tokenOID** en el **CryptoHashedToken** es fijado a "A" (que indica autenticación e integridad).

tokenOID "A" se utiliza para la protección de las H323-UU-PDU tunelizadas, incluidos todos los contenidos de mensaje H.245; el cálculo del troceado se efectuará sobre el mensaje **H.225.0 PDU** completo con todos los campos, de conformidad con el procedimiento descrito en la sección D.6.3.3.2.

• El autenticador se verifica en el extremo de cada rama de terminación de canal (EP1-GK1, GK1-GK2, GK2-EP2, EP1-GK2, GK1-EP2 o EP1-EP2, por ejemplo), y es recalculado antes del envío del mensaje a la rama siguiente.

NOTA 3 – El autenticador se calcula mensaje por mensaje.

NOTA 4 – Deberá utilizarse el método de relleno dentro la norma SHA1 [ISO 10118-3].

NOTA 5 – Cuando se utiliza la combinación de autenticación e integridad el autenticador se calcula sobre el mensaje completo.

NOTA 6 – Para evitar que se puedan producir ataques de reproducción, se recomienda decididamente que las implementaciones garanticen que se cambia la contraseña (clave) antes de una inversión (compleción del ciclo) del número secuencial monotónicamente creciente.

NOTA 7 – El recipiente es capaz de detectar la utilización del procedimiento I mediante la evaluación del **algorithmOID** dentro del **EncodedGeneralToken** troceado (detectando la presencia de "U").

D.6.3.3 Cálculo del troceado basado en contraseñas

Tanto el emisor como el receptor de un mensaje de autenticación/integridad calculan el troceado de claves sobre todos los campos del mensaje codificado en ASN.1 (utilizando el OID "A").

D.6.3.3.1 HMAC-SHA1-96

HMAC-SHA1-96 es el valor de troceado criptográfico de 96 bits truncado del cálculo de SHA1 de 160 bits. Los 96 bits más a la izquierda de la representación por bytes de red del valor de troceado se utilizaran como resultado. RFC 2104 describe el procedimiento con la clave secreta *K* fijada al secreto (= SHA1-contraseña troceada) compartido y *text* fijado al valor de memoria intermedia del mensaje.

D.6.3.3.2 Autenticación e integridad

Para la autenticación y la integridad de los mensajes (en caso de aplicarse un OID "A"), el procedimiento es el siguiente:

El emisor de un mensaje deberá calcular el troceado como sigue:

1) fijará el valor de troceado a un esquema por defecto específico de 96 bits de longitud. El esquema exacto de bits no importa aquí, pero constituye una buena elección un esquema de bits único que no aparezca en el mensaje restante;

2) codificará en ASN.1 el mensaje completo;

3) localizará[5] el esquema por defecto en el mensaje codificado; sobrescribirá el esquema de bits encontrado con los 96 bits cero;

4) calculará el valor de troceado criptográfico en el mensaje codificado en ASN.1 utilizando HMAC-SHA1-96 (véase D.6.3.3.1);

5) sustituirá el esquema por defecto en el menaje codificado por el valor de troceado calculado.

El recipiente recibe el mensaje y procede como sigue:

1) decodifica el mensaje en ASN.1;

2) extrae el valor de troceado recibido y lo guarda en un RV variable local;

3) busca y localiza el valor de troceado RV en el mensaje codificado recibido;

NOTA – En circunstancias poco frecuentes en que la subcadena de valor de troceado puede aparecer varias veces en el mensaje completo, deberán repetirse los pasos 3-6 sucesivamente arrancando de una posición de búsqueda diferente.

4) sobrescribe el esquema de bits en el mensaje codificado con los 96 bits cero;

5) calcula el valor de troceado criptográfico en el mensaje codificado utilizando HMAC-SHA1-96 (véase D.6.3.3.1);

6) compara RV con el valor de troceado calculado. El mensaje sólo se considera incorrupto si los dos valores de troceado son iguales; en este caso la autenticación ha tenido éxito y el procedimiento se detiene;

7) en los demás casos el recipiente repite los pasos 3-7 restableciendo RV a la ubicación anterior y buscando otra concordancia. Si ninguna comprobación de concordancia da como resultado una comparación de valores de troceado correcta, la autenticación ha fallado y el mensaje ha sido alterado (accidental o deliberadamente) durante el tránsito.

D.6.3.4 Ilustración de la utilización del procedimiento I

En las figuras D.1 a D.3 se representa la presencia de claves compartidas en el extremo de canales de comunicación para las diferentes combinaciones de canales H.225.0 con encaminamiento directo y por controlador de acceso. Con independencia del modelo de llamada, una clave secreta está siempre presente entre un EP y su GK a fin de proporcionar la autenticación e integridad del mensaje RAS. Cuando un canal RAS y un canal H.225.0 terminan entre los mismos dos nodos, se puede utilizar la misma clave para proporcionar la autenticación e integridad de ambos mensajes RAS y H.225.0.

La figura D.1 muestra el escenario más escalable en el que dos puntos extremos se encuentran dentro de zonas que aplican el modelo con encaminamiento por controlador de acceso. Todos los GK involucrados comparten mutuamente claves. Para que sea escalable, se recomienda el escenario representado en la figura D.1. Obsérvese que este escenario no proporciona una verdadera seguridad de extremo a extremo entre puntos extremos; toda la seguridad depende de los controladores de acceso intermedios de confianza.

---

[5] Esto puede implicar algunos pasos de prueba y error en el caso poco frecuente de que el esquema por defecto aparezca más de una vez en el mensaje.

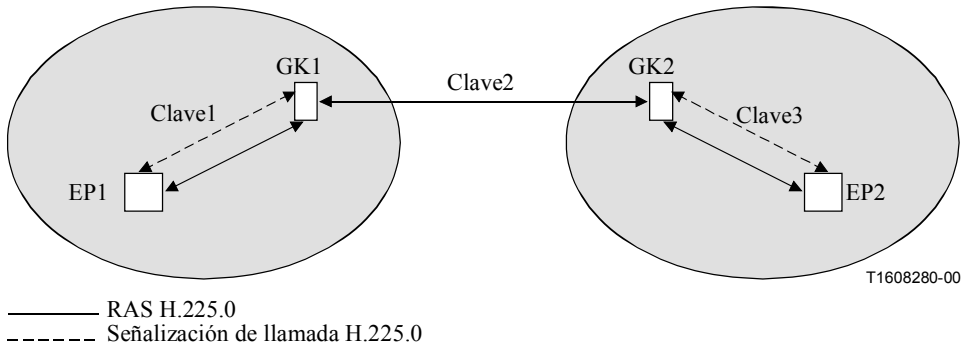


Figura D.1/H.235 – Ilustración de la utilización del procedimiento I en un escenario GK-GK con ambos EP en zonas de encaminamiento por controlador de acceso

La figura D.2 muestra un escenario mixto en el cual un EP se encuentra dentro de una zona en la es aplicable el modelo con encaminamiento por controlador de acceso mientras que el otro EP se encuentra en una zona donde es aplicable el modelo de encaminamiento directo. Este escenario puede darse en entornos cerrados en los cuales el número de EP2 y de GK1 es limitado.

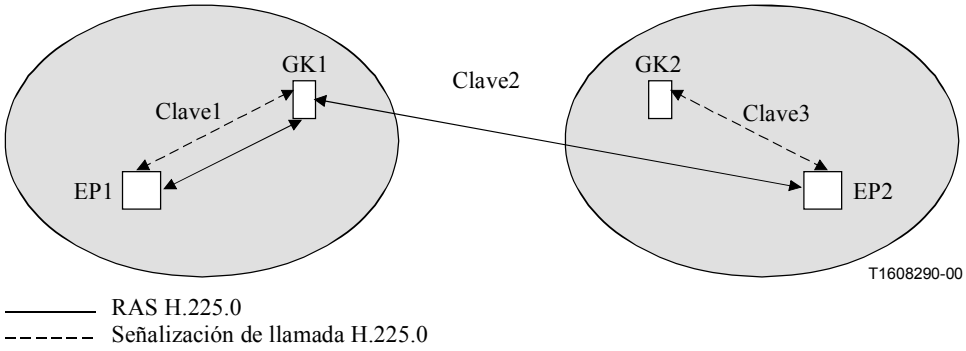


Figura D.2/H.235 – Ilustración de la utilización del procedimiento I en un escenario mixto con EP1 en una zona de encaminamiento por controlador de acceso y EP2 en una zona de encaminamiento directo

La figura D.3 muestra un escenario en el cual ambos EP se encuentran en zonas que aplican el modelo de GK con encaminamiento directo. Este escenario no es muy escalable cuando están implicados muchos EP. En principio, se recomienda la utilización en su lugar del anexo E con los procedimientos II/III. Para este escenario específico y los procedimientos I, II o III se necesitan también medidas de seguridad adicionales[6], las cuales no se describen en esta Recomendación; este tema queda en estudio. Obsérvese que este escenario proporciona una verdadera seguridad de extremo a extremo entre puntos extremos, sin que dependa de nodos intermedios de confianza.

[6] Que protejan contra el fraude y la utilización incorrecta de llamadas por medio de la autorización de la llamada con testigos de acceso en controladores de acceso H.323, por ejemplo.

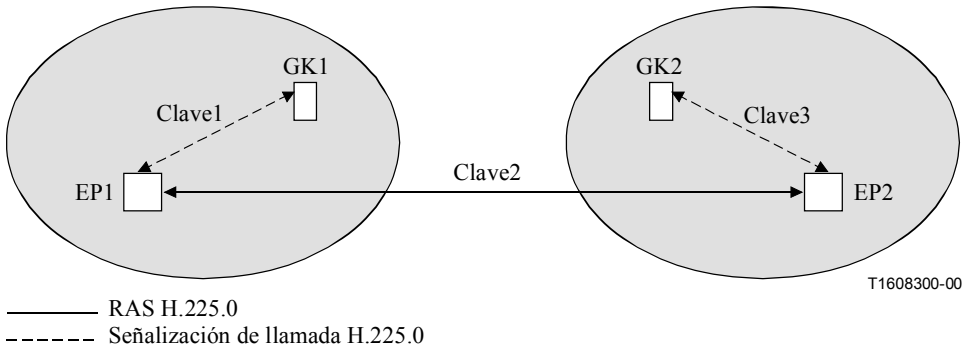


Figura D.3/H.235 – Ilustración de la utilización del procedimiento I en un escenario con ambos EP en zonas que utilizan un modelo de GK con encaminamiento directo

Consideremos el caso de la figura D.1 donde tres contraseñas son compartidas por parejas: entre EP1-GK1, entre GK1-GK2 y entre GK2-EP2, respectivamete. A partir de estas contraseñas se generan tres claves de 20 bytes – *Key1 (clave 1), Key2 (clave 2)* y *Key3 (clave 3)* – basándose en el procedimiento descrito en 10.3.2. Para conseguir una seguridad máxima se recomienda hacer independientes cada una de las tres contraseñas/claves aleatorias.

Más adelante se detalla el procedimiento para la autenticación/integridad de los mensajes RAS H.225.0 y H.245. El ejemplo de descripción representa parámetros específicos en un modelo con encaminamiento por controlador de acceso; también son posibles otras combinaciones válidas y útiles de identificadores de objeto en diferentes escenarios.

NOTA – Los escenarios que se muestran en las figuras anteriores no se escalan bien cuando el número de claves (contraseñas) simétricas compartidas entre GK (figura D.1), entre GK y EP distantes (figura D.2) o entre los EP (figura D.3) es demasiado grande.

D.6.3.4.1 Autenticación e integridad de los mensajes RAS

Consideremos el caso en que EP1 desea enviar un mensaje RAS – por ejemplo, un mensaje **ARQ** (petición de admisiones) – a GK1. EP1 genera una indicación de tiempo y un número secuencial y los incluye en los campos **timeStamp** y **random** respectivamente, junto con el alias del GK1 en el campo **generalID** y el ID de EP en el campo **sendersID**. Estos campos están presentes en el campo **ClearToken** del **hashedVals** presente en el **cryptoHashedToken** del campo **CryptoToken** del **cryptoH323Token** del mensaje **ARQ**.

El **tokenOID** dentro del **cryptoHashedToken** se fija a "A", indicando que todos los campos en el mensaje **ARQ** están troceados. El **HASHED** dentro de **token** en **cryptoHashedToken** tiene el **algorithmOID** puesto a "U", indicando el uso de HMAC-SHA1-96, y **params** puesto a NULO. EP1 calcula entonces el autenticador basado en el HMAC-SHA1-96 utilizando la clave de 12 bytes *Key 1*. El autenticador es calculado sobre el mensaje RAS entero.

EP1 incluye el autenticador calculado dentro de **hash** en el campo t**oken** del campo **cryptoHashedToken** del **CryptoToken** presente en el **cryptoH323Token** del mensaje **ARQ**. El mensaje **ARQ** es enviado entonces al GK1.

Tras la recepción del mensaje **ARQ**, GK1 verifica el autenticador basándose en varios criterios que incluyen:

- Vida de **timestamp** y unicidad del **random**.
- Identidad del **generalID** e identificador propio.
- Concordancia del autenticador en el mensaje **ARQ** con el calculado por GK1.

D.6.3.4.2 Autenticación e integridad de los mensajes H.225.0

Consideremos el caso en que EP1 desea enviar un mensaje H.225.0 – por ejemplo, un mensaje **Setup** – a EP2. EP1 genera una indicación de tiempo y un número secuencial y los incluye en los campos **timeStamp** y **random** respectivamente, junto con el alias del GK1 en el **generalID** y el ID de EP en el campo **sendersID**. EP1 calcula también media clave Diffie-Hellman e incluye los parámetros Diffie-Hellman **halfkey**, **modsize** y **generator** en el campo **dhkey** del **ClearToken**. Estos campos están presentes en el campo **ClearToken** de **hashedVals** presente en el **cryptoHashedToken** del campo **CryptoToken** del **cryptoH323Token** del mensaje **Setup**.

El **tokenOID** dentro del **cryptoHashedToken** se fija a "A", indicando con ello que todos los campos en el mensaje **Setup** están troceados. El **HASHED** dentro de **token** en **cryptoHashedToken** tiene el **algorithmOID** puesto a "U", indicando la utilización de HMAC-SHA1-96, y **params** puesto a NULO. EP1 calcula a continuación el autenticador basado en el algoritmo HMAC-SHA1 utilizando la clave de 12 bytes, *Key1*. El autenticador es calculado de conformidad con el método de troceado elegido (A) tomando en consideración el mensaje H.225.0 completo.

EP1 incluye el autenticador calculado dentro de **hash** en el campo **token** del campo **cryptoHashedToken** del **CryptoToken** presente en el **cryptoH323Token** del mensaje **Setup**. A continuación se envía el mensaje **Setup** a GK1.

Tras la recepción del mensaje **Setup**, GK1 verifica el autenticador basándose en varios criterios que incluyen:

- La vida de la **timestamp** y la unicidad del r**andom**.
- La identidad del **generalID** y el identificador propio.
- La verificación de parámetros Diffie-Hellman, por ejemplo, probando si el primo de 1024 bits y el generador son correctos. La prueba de seguridad de los parámetros Diffie-Hellman es un proceso que consume tiempo y solamente puede realizarse cuando la política local lo requiere.
- Concordancia del autenticador en el mensaje **Setup** con el calculado por GK1.

Si el autenticador es verificado con éxito, GK1 calcula un nuevo autenticador para insertarlo (sustituirlo) en el mensaje **Setup** antes de reenviarlo a GK2 como sigue. GK1 reemplaza los campos **timeStamp**, **random**, **sendersID** y **generalID** en el campo **ClearToken** de **hashedVals** utilizando valores pertinentes a la rama GK1-GK2. El campo **timestamp** contiene la indicación de tiempo actual, el campo **random** contiene el siguiente número secuencial monotónicamente creciente para la rama GK1-GK2, el campo **generalID** contiene el alias de GK2 y el **sendersID** contiene el alias de GK1. GK1 incluye también los parámetros Diffie-Hellman recibidos en el campo **dhkey** del **ClearToken**.

GK1 calcula después un nuevo autenticador para el mensaje **Setup** utilizando la clave *Key2 (Clave2*) y el algoritmo HMAC-SHA1-96 (**algorithmOID**="U"), lo inserta en **hash** dentro de **token** y pasa el mensaje **Setup** al GK2.

Tras la recepción del mensaje **Setup**, GK2 verifica el autenticador, calcula un nuevo autenticador después de modificar los campos **ClearToken** en **hashedVals** adecuadamente, lo inserta en el campo **hash** y pasa el mensaje **Setup** al EP2.

D.6.3.4.3 Autenticación e integridad de los mensajes H.245

Consideremos el caso en el que EP1 desea enviar un mensaje H.245 – por ejemplo, un mensaje **TerminalCapabilitySet** – a EP2. EP1 comprueba si es necesario enviar un mensaje H.225.0 a GK1. En caso afirmativo, el mensaje H.245 es tunelizado dentro de dicho mensaje H.225.0. Los campos dentro del mensaje H.225.0 se fijan del modo descrito anteriormente para la transmisión de un mensaje H.225.0. Puesto que el mensaje H.245 está tunelizado, la **h323-uu-pdu** en el mensaje **h323-UserInformation** tiene sus campos fijados como sigue:

- el campo **h323-message-body** es puesto al tipo de mensaje H.225.0 que está siendo transmitido;
- **h245Tunneling** se fija a VERDADERO (TRUE);
- **h245Control** contiene la cadena de octetos PDU H.245.

EP1 genera un **CryptoToken** para el mensaje H.225.0, pone **tokenOID** a "A" indicando autenticación e integridad, fija **timeStamp**, **random**, **sendersID**, **generalID** y **tokenOID** a "T" en el **ClearToken** del **hashedVals**, fija **algorithmOID** a "U" indicando la utilización de HMAC-SHA1-96 y **hash** al autenticador troceado calculado sobre todos los campos del mensaje **H323-UU-PDU**.

Sin embargo, si no hay ningún mensaje H.225.0 pendiente de transmisión, el mensaje H.245 es tunelizado dentro de un mensaje **facility** H.225.0 ad-hoc. La **h323-uu-pdu** en el mensaje **h323-UserInformation** tiene sus campos fijados como sigue:

- el campo **h323-message-body** es fijado a **facility** que contiene:
  - **reason** puesto a **undefinedReason**;
  - **tokens** y **cryptoTokens** fijados como para cualquier mensaje H.225.0;
- **h245Tunneling** fijado a VERDADERO (TRUE);
- **h245Control** contiene la cadena de octetos PDU H.245.

Tal como se ha descrito anteriormente, EP1 genera un **CryptoToken** como parte del mensaje **facility** de H.225.0. El mensaje **facility** es a continuación transmitido por EP1 a GK1.

En cualquiera de los dos casos (si está pendiente de transmisión un mensaje H.225.0 o si se utiliza un mensaje **facility** H.225.0 ad hoc), GK1 verifica el autenticador tras la recepción del mensaje. A continuación, si está pendiente de transmisión un mensaje H.225.0 para la rama GK1-GK2, el mensaje H.245 es tunelizado dentro de ese mensaje; en caso contrario, es tunelizado dentro de un mensaje **facility** H.225.0 ad hoc. Al igual que en el caso de la transmisión de un mensaje H.225.0, se calcula un nuevo autenticador para el mensaje H.225.0 antes de su transmisión de GK1 a GK2. El proceso se repite para la rama GK2-EP2.

D.6.4 Escenario con encaminamiento directo

Las entidades H.323 aseguradas no sólo puede comunicarse dentro del entorno con encaminamiento por controlador de acceso como se señala en esta Recomendación, sino que puede, desplegar también el modo de encaminamiento directo. Este modelo con encaminamiento directo requiere medidas de seguridad adicionales (testigos de acceso) que no son necesarias en los entornos con encaminamiento por controlador de acceso más sencillos. El aseguramiento del modelo con encaminamiento directo queda por tanto en estudio.

D.6.5 Soporte de los servicios fuera del terminal

La entidades H.323 aseguradas pueden utilizar servicios fuera del terminal de conformidad con el procedimiento descrito en I.4.6.

D.6.6 Compatibilidad con H.235 Versión 1

Aunque estos perfiles de seguridad se han desarrollado pensando en H.235 versión 2 [H.235 (2000)], se pueden también aplicar a la H.235 versión 1 [H.235 (1998)] con algunas modificaciones menores. Un recipiente es capaz de detectar la presencia de la versión de protocolo H.235 del emisor mediante la evaluación de los identificadores de objeto de perfil de seguridad (véase D.11).

Implementaciones de H.235 versión 1 [H.235 (1998)]:

- no fijar o evaluar el **sendersID** en el **ClearToken**;
- no puede utilizar los servicios fuera del terminal como en D.6.5.

D.6.7 Comportamiento multidifusión

Los mensajes de multidifusión H.225.0, como GRQ o LRQ, no deberán incluir un CryptoToken de conformidad con el procedimiento I. Cuando tales mensajes son enviados en unidifusión deberán incluir un CryptoToken.

D.7 Perfil de seguridad de criptación vocal

El procedimiento general establece un secreto compartido (intercambio Diffie-Hellman) entre las dos partes comunicantes al iniciarse una conexión. Este secreto compartido se utiliza entonces para proteger (un conjunto de) claves de medios que son utilizadas para criptar las sesiones de medios (RTP).

El perfil de seguridad de criptación vocal es una mejora facultativa del perfil de seguridad básico y del perfil de seguridad de firmas; su empleo puede negociarse como parte de la negociación de capacidades de seguridad del terminal. En los contextos en que la confidencialidad de la conversación está asegurada por otros medios, no es necesario implementar la criptación de medios y los procedimientos de gestión de claves correspondientes (convenio de claves Diffie-Hellman, actualización de claves y sincronización).

Los algoritmos de criptación elegidos son compatible con RC2, DES y DES triple. Debe observarse que, como una implementación del algoritmo DES triple se puede también utilizar para el algoritmo DES, el resultado es una implementación compacta. Con independencia del algoritmo de criptación de medios específico que se haya elegido, deberán seguirse de manera explícita las opciones a continuación.

- Generación, si es necesario, del vector inicialización (IV) como se especifica en B.3 a).
- Relleno, si es necesario, de acuerdo con la descripción del anexo B.

La cabida útil[7] audio es criptada mediante el algoritmo de criptación negociado ("X", "Y" o "Z") operando en el modo CBC de conformidad con los procedimientos descritos en la cláusula 11 y en el anexo B y con los métodos de relleno de texto cifrado de I.1.

D.7.1 Gestión de claves

- Durante la secuencia **Setup**-a-**Connect** se efectúa un intercambio Diffie-Hellman (DH) – este intercambio dota a ambos puntos extremos de un secreto compartido. El campo **ClearToken** de los campos **CryptoToken** deberá contener un **dhkey**, utilizado para transferir los parámetros como se especifica en esta Recomendación. **halfkey** contiene la clave pública aleatoria de una parte, **modsize** contiene el primo DH y **generator** contiene el grupo DH. Los parámetros DH que han de utilizarse se indican en el cuadro 4. Para más detalles, véase [RFC 2412, apéndice E2]. Obsérvese que, como los mensajes H.225.0 están autenticados (como se ha descrito anteriormente mediante el procedimiento I), el intercambio DH está ahora autenticado.
- Durante la operación FastStart, el llamador (fuente del mensaje **Setup**) presenta su testigo DH y las estructuras FastStart soportadas. Deben ofrecerse ambos canales H235Cap y nonH235Cap. El **mediaWaitForConnect** debe fijarse a VERDADERO[8].
- Durante la operación FastStart, el llamado (fuente del mensaje **Connect**) presenta su testigo DH y las estructuras FastStart aceptadas. La clave de sesión es incluida en el campo

---

[7] Sin el encabezamiento de cabida útil.

[8] Obsérvese que en este caso, si el llamado envía medios criptados al llamador (lo que teóricamente puede realizar porque posee las direcciones RTP/RTCP del llamador), el llamador no será capaz de descifrarlos sin el secreto compartido proporcionado por el mensaje de conexión (aviso, llamada en curso). (A los fines de la relación de seguridad, el llamado es a priori el director.)

encyptionSync. La propia clave de sesión es ella misma criptada con el secreto compartido DH, del mismo modo que en la operación non-FastStart.

- Durante el intercambio de capacidades H.245, los puntos extremos presentan los valores de **H235Capability** para los códecs que ellos soportan. Cada códec está asociado con una capacidad H.235 determinada. Estas capacidades deben indicar el soporte compatible con RC2 de 56 bits (OID – "X"), y el soporte DES de 56 bits (OID – "Y"), y pueden indicar el soporte DES triple de 168 bits (OID – "Z").

  Los algoritmos de criptación negociados y sus modos de operación para la criptación de trenes de medios se utilizarán también para garantizar la distribución de la clave de sesión. El algoritmo de criptación de la clave de medios deberá operar en el mismo modo de encadenamiento que el algoritmo de criptación de medios.

- Las respuestas **OpenLogicalChannel(Ack)** son emitidas con la clave de sesión creada (maestra) incluida en el campo **encryptionSync**. La propia clave de sesión es criptada con el secreto compartido DH en la forma descrita más abajo[9].

- El **OpenLogicalChannel** transporta tanto los **forwardLogicalChannelParameters** como los **reverseLogicalChannelParameters** con **dataType** proporcionando **h235Media** con **encryptionAuthenticationAndIntegrity** donde, en la **encryptionCapability**, deberá estar presente un **MediaEncryptionAlgorithm** como máximo.

NOTA – Cuando no se dispone de ningún algoritmo de criptación en ambos lados, el tren de medios puede dejarse sin criptar o puede abortarse la conexión, dependiendo de la política de seguridad seguida.

- La clave de sesión criptada deberá cursarse en el H.235Key/**sharedSecret** dentro del campo **encryptionSync**. La clave de sesión deberá ser cursada en el campo **keyMaterial** del **KeySyncMaterial**. El **KeySyncMaterial** es criptado utilizando:
  - 56 bits del secreto compartido, arrancando con los bits menos significativos del secreto Diffie-Hellman para el OID "X" o el OID "Y";
  - todos los bits del secreto compartido para el OID "Z," arrancando con los bits menos significativos del secreto DH.

Debe incluir el valor de **generalID** para proporcionar un nivel mínimo de autenticación de la fuente de la clave de sesión (véase también D.7.2). El recipiente debe verificar la corrección del **generalID** recibido.

Cada entidad deberá tomar los bits menos significativos adecuados del secreto Diffie-Hellman compartido común para la clave de criptación de claves (clave maestra); es decir, los 56 bits menos significativos del secreto Diffie-Hellman para el OID "X" o el OID "Y" y los 168 bits menos significativos del secreto Diffie-Hellman para el OID "Z".

---

[9] Obsérvese que no hay prescrito ningún método para la generación de claves de sesión utilizadas para la criptación de los medios. La generación de estos valores es incumbencia de la implementación afectada por los recursos locales, la política y los algoritmos de criptación utilizados. Debe evitarse la generación de claves débiles.

Cuadro D.4/H.235 – Grupos Diffie-Hellman

OID	Descripción del grupo D-H
"X" (compatible con RC2), "Y" (DES)	Mod-P, cualquier primo de 512 bits adecuado
"Z" (DES triple)	Mod-P, primo de 1024 bits Primo = $2^{1024} - 2^{960} - 1 + 2^{64} \times \{ [2^{894}\ \text{pi}] + 129093 \}$ = (179769313486231590770839156793787453197860296048756011706444 423684197180216158519368947833795864925541502180565485980503 646440548199239100050792877003355816639229553136239076508735 759914822574862575007425302077447712589550957937778424442426 617334727629299387668709205606050270810842907692932019128194 467627007)$_{10}$ Generador[a] = 2
[a] El generador se utiliza para generar el testigo DH.	

D.7.2 Actualización de claves y sincronización

La tasa de renovación de claves *deberá* ser tal que no se cripten más de 2^{32} bloques con la misma clave. Las implementaciones *deberían* renovar las claves antes de que se hayan criptado 2^{30} bloques utilizando la misma clave (véase 10.3). Las dos entidades involucradas tienen libertad para intercambiar la clave de sesión de medios con la frecuencia que consideren necesaria de acuerdo con su política de seguridad. Por ejemplo, el terminal director puede distribuir una nueva clave de sesión utilizando la **encryptionUpdate** del mensaje **miscellaneousCommand**. Por otra parte, el terminal subordinado puede solicitar una nueva clave de sesión al terminal director para cambiar la actual utilizando la **encryptionUpdateRequest** del mensaje **miscellaneousCommand**.

El mensaje **MiscellaneousCommand** contiene la **encryptionUpdate** cuya **encryptionSynch** está fija con los siguientes parámetros:

- **synchFlag**: el nuevo número de cabida útil RTP dinámica que indica la conmutación de clave;
- **h235key**: que cursa la nueva clave de sesión criptada. Es un parámetro **H235Key** codificado en ASN.1 H.235 pasado como una cadena de octetos.

El campo **sharedSecret** dentro de la estructura **H235Key** utiliza los siguientes campos

- **algorithmOID**: puesto a "X" para el compatible con RC2 de 56 bits, puesto a "Y" para el DES de 56 bits o puesto a "Z" para el DES triple de 168 bits. Éste es el algoritmo de criptación cuya clave de sesión de medios está siendo encriptada.

NOTA 1 – El algoritmo de criptación de clave de sesión es el algoritmo de criptación de medios negociado.

- **paramS**: puesto al valor inicial. **iv8** contiene un esquema de bits de bloques de 64 bits aleatorios que genera el iniciador. Este campo no se usa en el modo CBC y se debe poner a NULO (NULL).
- **encryptedData**: puesto al resultado del **KeySyncMaterial** criptado.

Como parte del **KeySyncMaterial**:

- **generalID**: identificador de la fuente que distribuye la clave.
- **keyMaterial**: puesto a la nueva clave de sesión. Para DES y compatible con RC2 ésta es un clave de 56 bits, para DES triple es una clave de 168 bits. El terminal director deberá generar una nueva clave de sesión que cumpla al menos los siguientes criterios de seguridad: no ha de ser una clave DES débil o semidébil y utilizará una fuente aleatoria suficientemente segura.

El mensaje **MiscellaneousCommand** contiene la **encryptionUpdateRequest** que a su vez contiene el **keyProtectionMethod** en el que la bandera de **sharedSecret** es puesta a VERDADERO.

NOTA 2 – Como la actualización y sincronización de claves depende de mensajes H.245 que no son transportados durante la conexión rápida, es necesario utilizar la tunelización H.245 para las entidades H.323 aseguradas. Por ello, la actualización y sincronización de claves sólo se puede utilizar en el perfil de seguridad de firmas.

D.7.3 DES triple en modo CBC exterior

La DES triple de 168 bits en modo CBC exterior, como se ilustra en la figura D.4, *debería* utilizarse dentro de este perfil de seguridad. En la figura, cada k_i se refiere a una clave de 56 bits. Una clave de 56 bits diferente *deberá* utilizarse dentro de cada bloque de criptación (E, *encryption*) y descriptación (D, *decryption*). No se conoce que ninguna de las 64 claves débiles para DES ocasione alguna debilidad dentro de la DES triple. Sin embargo, las implementaciones que cumplan este perfil deberían rechazar la clave cuando está implicada una clave DES débil [véase RFC 2405].

En [Schneier] y [RFC 2405] puede encontrarse más información sobre DES triple.

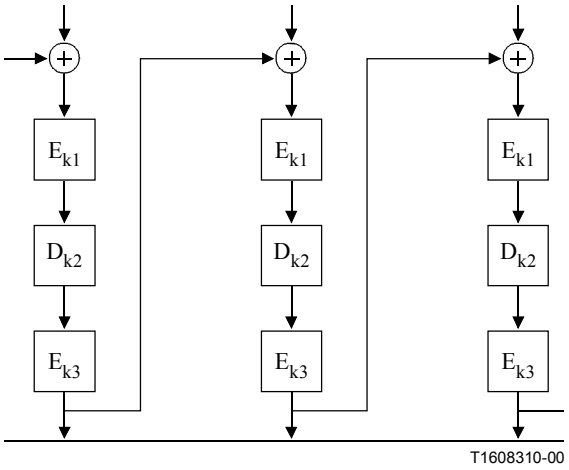


Figura D.4/H.235 – Criptación DES triple en modo CBC exterior

D.8 Interceptación legal

Queda en estudio (véase [LI]).

D.9 Lista de mensajes de señalización asegurados

En esta cláusula se presenta un resumen de cómo y por que medios el anexo D asegura los distintos mensajes de señalización H.323.

D.9.1 RAS H.225.0

Mensaje RAS H.225.0	Campos de señalización H.235	Autenticación e integridad
Cualquiera	cryptoTokens	Procedimiento I

D.9.2 Señalización de llamada H.225.0

Mensaje de señalización de llamada H.225.0	Campos de señalización H.235	Autenticación e integridad
UUIE Aviso, UUIE Llamada en curso, UUIE Conexión, UUIE Establecimiento, UUIE Facilidad, UUIE Progresión, UUIE Información, UUIE Liberación completa	cryptoTokens	Procedimiento I

D.9.3 Control de llamada H.245

Los mensajes H.245 con destino a, o procedentes de, entidades H.323 aseguradas deberán ser transportados como parte de la conexión rápida segura o ser tunelizados utilizando el mensaje **UUIE Facilidad** H.225.0.

D.10 Utilización de sendersID y de generalID

El **ClearToken** guarda los campos **sendersID** (identificador del emisor) y **generalID**. Cuando se dispone de información de identificación, el **sendersID** debe igualarse al identificador del controlador de acceso (GKID, *gatekeeper identifier*) para los mensajes iniciados por el controlador de acceso y al identificador de punto extremo (EPID, *endpoint identifier*) para los mensajes iniciados por el punto extremo. Cuando se dispone de información de identificación, el **generalID** debe igualarse al GKID para los mensajes iniciados por el punto extremo y al EPID para los mensajes iniciados por el controlador de acceso. Cuando no se dispone de información de identificación o cuando la radiodifusión/multidifusión es ambigua es porque falta el campo o porque éste debería contener una cadena nula. El cuadro D.5 resume la situación.

Cuadro D.5/H.235 – Identificadores de objeto utilizados en el anexo D

Mensaje	sendersID	generalID
GRQ unidifusión	**EPID** si está disponible en su defecto **NULL**	**GKID**
GRQ multidifusión	**EPID** si está disponible en su defecto **NULL**	
GCF, **GRJ**	**GKID**	**EPID** si está disponible en su defecto **NULL**
RRQ inicial	**EPID** si está disponible en su defecto **NULL**	**GKID**
RCF	**GKID**	**EPID**
RRJ	**GKID**	
URQ, **UCF**, **URJ**, **BRQ**, **BCF**, **BRJ**, **DRQ**, **DCF**, **DRJ**, **NSM**, **RIP**, **SCI**, **SCR**, **XRS** (EP a GK)	**EPID**	**GKID**
URQ, **UCF**, **URJ**, **BRQ**, **BCF**, **BRJ**, **DRQ**, **DCF**, **DRJ**, **NSM**, **RIP**, **SCI**, **SCR**, **XRS** (GK a EP)	**GKID**	**EPID**
ARQ, **IRQ**, **RAI**	**EPID**	**GKID**
ACF, **ARJ**, **BCF**, **LCF**, **LRJ**, **IRR**, **IRQ**, **RAC**, **LCF**, **LRJ**, **IACK**, **INAK**	**GKID**	**EPID**
LRQ unidifusión (EP a GK)	**EPID**	**GKID**
LRQ unidifusión (GK a GK)	**GKID**	**GKID**
LRQ multidifusión	**EPID**	
NOTA – GKID es el identificador del controlador de acceso, EPID es el identificador de punto extremo. Un espacio en blanco equivale a una cadena de identificación faltante o nula.		

D.11 Lista de identificadores de objeto

En el cuadro D.6 se listan todos los OID referenciados (véase también [OIW] y [WEBOIDs]). No hay identificadores de objeto para H.235v1 [H.235 (1998)] ni para H.235v2 [H.235 (2000)].

Cuadro D.6/H.235 – Identificadores de objeto utilizados en el anexo D

Referencia de identificador de objeto	Valor(es) de identificador de objeto	Descripción
"A"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 1} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 1}	Utilizado en los procedimientos I para el CryptoToken-tokenOID, indicando que el troceado incluye todos los campos del mensaje RAS/ H.225.0 (autenticación e integridad).
"T"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 5} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 5}	Utilizado en el procedimiento I para el ClearToken-tokenOID, indicando que el ClearToken se está utilizando para la autenticación e integridad de mensajes.
"U"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 6} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 6}	Utilizado en el procedimiento I para el algoritmo OID, indicando la utilización de HMAC-SHA1-96.
"X"	{iso(1) member-body(2) us(840) rsadsi(113549) encryptionalgorithm(3) 2}	Criptación vocal utilizando compatible con RC2 (56 bit) o compatible con RC2 en modo CBC y grupo DH de 512 bits
"Y"	{iso(1), identified-organization(3), oiw(14), secsig(3), algorithm(2), descbc(7)}	Criptación vocal utilizando DES (56 bit) en modo CBC y grupo DH de 512 bits
"Z"	{iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) desEDE(17)}	Criptación vocal utilizando DES triple (168-bit) en modo CBC exterior y grupo DH de 1024 bits

D.12 Bibliografía

[FIPS PUB 180-1] NIST, FIPS PUB 180-1: Secure Hash Standard, abril de 1995, http://csrc.nist.gov/fips/fip180-1.ps

[LI] Draft DRT/TIPHON-08003 V0.0.9, "Lawful Interception – Internal LI Interface", agosto de 2000.

[OIW] Stable Implementation – Agreements for Open Systems Interconnection Protocols: Part 12 – OS Security; Output from the December 1994 Open Systems Environment Implementors' Workshop (OIW); http://nemo.ncsl.nist.gov/oiw/agreements/stable/OSI/12s_9412.txt

[RFC 2405] C. Madson, N. Doraswamy "The ESP DES-CBC Cipher Algorithm With Explicit IV", RFC 2405, *Internet Engineering Task Force,* 1998

[WEBOIDs] http://www.alvestrand.no/objectid/top.html

ANEXO E

Perfil de firmas

E.1 Visión general

Este anexo describe un perfil de firmas, el cual despliega firmas digitales que son propuestas como una opción. Las entidades de seguridad H.323 (terminales, controladores de acceso, MCU, etc.) pueden implementar este perfil de seguridad de firmas para mejorar la seguridad, o siempre que se desee.

El perfil de seguridad de firmas gobierna el modelo con encaminamiento por controlador de acceso y está basado en las técnicas de tunelización H.245; el soporte de modelos diferentes del modelo con encaminamiento por controlador de acceso queda en estudio.

El perfil de seguridad de firmas es aplicable a la telefonía IP "global" escalable; este perfil de seguridad supera las limitaciones del perfil de seguridad básico simple del anexo D. Por ejemplo, el perfil de seguridad de firmas no depende de la administración de secretos compartidos mutuos de los saltos en diferentes dominios. Proporciona la tunelización de mensajes H.245 para la integridad de mensajes H.245 y contiene también disposiciones para el no repudio de los mensajes. El perfil de seguridad de firmas soporta la seguridad salto por salto y la autenticación de extremo a extremo verdadera, con el uso simultáneo de controladores de acceso intermedios o apoderados H.235.

Estos perfiles proporcionan las siguientes características, para los mensajes RAS, H.225.0 y H.245:

- la autenticación del usuario a una entidad deseada independientemente del número de saltos[10] del nivel de aplicación que el mensaje atraviesa.
- La integridad de todos los mensajes, o porciones (campos) críticas de los mismos, que llegan a una entidad, con independencia del número de saltos del nivel de aplicación que el mensaje atraviesa. La integridad del propio mensaje mediante la generación de un número aleatorio resistente es también facultativa.
- La autenticación, integridad y no repudio del mensaje salto por salto a nivel de aplicación proporciona estos servicios de seguridad para el mensaje completo.
- Se puede proporcionar también el no repudio de mensajes intercambiados entre dos entidades independientemente del número de saltos del nivel de aplicación que el mensaje atraviesa. Es particular, el no repudio es proporcionado para porciones (campos) críticas del mensaje. Tal puede ser, por ejemplo, el caso de un EP que envía un mensaje ESTABLECIMIENTO a su controlador de acceso y ambos (el EP y el controlador de acceso) son divididos por uno o más apoderados.

Mediante la provisión de manera adecuada de los servicios de seguridad anteriores se frustran varios ataques. Estos ataques son:

- Ataques de denegación de servicio: una comprobación rápida de las firmas digitales puede proteger contra tales ataques.
- Ataques intermedios: la autenticación e integridad de los mensajes salto por salto al nivel de aplicación previene contra tales ataques cuando el punto intermedio se encuentra en un salto del nivel de aplicación, es decir un encaminador hostil. Cuando el punto de ataque intermedio es una entidad del nivel de aplicación, tales ataques se evitan utilizando la autenticación e integridad de extremo a extremo para porciones seleccionadas del mensaje.

---

[10] "Salto" tiene aquí el sentido de un elemento de red H.235 de confianza (por ejemplo, GK, GW, MCU, apoderado, cortafuegos). Por tanto, la seguridad salto por salto en el nivel de aplicación cuando se utiliza con técnicas simétricas no proporciona una verdadera seguridad de extremo a extremo entre terminales.

- Ataques de reproducción: estos ataques se evitan mediante la utilización de indicaciones de tiempo y números secuenciales.
- Engaño: la autenticación del usuario evita estos ataques.
- Asalto a la conexión: el uso de la autenticación/integridad para cada mensaje de señalización evita estos ataques.

E.2 Convenios acerca de las especificaciones

En caso necesario, el perfil de seguridad de firmas puede utilizar el **perfil de seguridad de criptación vocal** del anexo D para conseguir la confidencialidad de la conversación.

Los procedimientos II y III especifican la forma de implementar los servicios de seguridad para diferentes escenarios, como el método salto por salto y el método de extremo a extremo, mediante diferentes mecanismos de seguridad tales como las técnicas criptográficas asimétricas (firma digital).

Si bien el servicio de integridad de mensajes también proporciona siempre la autenticación del mensaje, la inversa no es cierta. En el modo de autenticación solamente, la integridad se asegura solamente para un subconjunto determinado de campos del mensaje. Este modelo se aplica a los servicios de integridad realizados por medios asimétricos (por ejemplo, firmas digitales). Con ello, en la práctica, el servicio combinado de autenticación y seguridad explota el mismo material de claves sin que con ello introduzca una debilidad en la seguridad.

Además, la información de seguridad salto por salto se introduce en el elemento **CryptoSignedToken**. Esta información se recalcula en cada salto de conformidad con el procedimiento II.

Por otra parte, la información de seguridad de extremo a extremo – posible solamente cuando se utiliza el apoderado H.323 y el procedimiento III – calcula básicamente información similar a la introducida en el **CryptoSignedToken**, pero almacena esta información en un **CryptoToken** independiente del mensaje. Esta información no es modificada en el tránsito. Un identificador de objeto separado permite distinguir entre los **CryptoTokens** de salto por salto y de extremo a extremo.

Autoridades de certificación: Autoridades de certificación (CA, *certification authorities*), cuando se utilizan en el contexto de la firma electrónica, que certifican las claves de verificación públicas mediante la expedición de "Certificados".

Depósitos de certificados (*certificate revocation lists*): Los depósitos de certificados (por ejemplo, un Directorio X.500) mantienen los certificados de usuario y las listas de revocación de certificados (CRL). Pueden garantizar que esta información se encuentre accesible, pero no son responsables del contenido y la exactitud de la información que reciben de las CA o las RA.

Firma digital: Transformación criptográfica (que utiliza una técnica criptográfica asimétrica) de la representación numérica de un mensaje de datos, de modo que cualquier persona que tenga el mensaje firmado y la clave pública pertinente puede determinar:

i) que la transformación se creó utilizando la clave privada correspondiente a la clave pública pertinente; y

ii) que el mensaje firmado no ha sido alterado desde que se realizó la transformación criptográfica.

Proveedores de estado de certificado en línea: El protocolo de estado de certificado en línea (OCSP, *on-line certificate status protocol*) permite a las aplicaciones determinar el estado de revocación de un certificado identificado. El OCSP puede utilizarse para satisfacer algunos de los requisitos operacionales de la provisión de información de revocación del modo más oportuno posible en el tiempo mediante listas CRL. Los proveedores de estado de certificado en línea pueden considerarse una alternativa a la utilización de las CRL fuera de línea.

Apoderado – sustituto (proxy): El apoderado (sustituto) es una entidad H.323 similar a un controlador de acceso. El apoderado puede ser un nodo de red separado o estar cosituado con la funcionalidad de una entidad H.323, como uno de los controladores de acceso. El apoderado puede realizar tareas de seguridad como la verificación de firmas y certificados y el control de acceso.

Autoridades de registro: Las autoridades de registro actúan como intermediarios entre los usuarios y las CA. Reciben peticiones de los usuario y las transmiten a las CA en forma adecuada.

Autoridades de indicaciones de tiempo: Las autoridades de indicaciones de tiempo son obligatorias para el no repudio en caso de que la clave se haya perdido o esté comprometida. En la práctica estas autoridades proporcionan a cualquiera una contrafirma, incluido un tiempo fiable, sobre un troceado y un identificador de troceado.

Proveedor de servicio de confianza: Entidad que puede ser utilizada por otras entidades como intermediario de confianza en una comunicación o proceso de verificación, o como proveedor de confianza del servicio de información.

El perfil de seguridad de firmas se propone como una opción. Este perfil de seguridad es aplicable en ambientes en los cuales pueda haber muchos terminales y donde la asignación de claves simétricas/contraseñas no es factible, por ejemplo, en los escenarios de escala global o de gran escala. El perfil de seguridad de firmas proporciona servicios de seguridad adicionales para el no repudio mediante certificados y firmas digitales. Las firmas digitales pueden utilizar el troceado SHA1 o MD5 y proporcionar la autenticación y/o la integridad (véanse los procedimientos II y III).

Las entidades H.323 que utilizan autenticación e integridad o la autenticación solamente en un modo salto por salto deberán utilizar el procedimiento II. Las entidades H.323 que utilizan sencillamente la autenticación solamente no implementarían la integridad. Las entidades H.323 con autenticación solamente utilizarán el procedimiento II para la autenticación verdadera de extremo a extremo.

El perfil de seguridad de firmas permite tunelizar de modo seguro las PDU de control de llamada H.245 dentro de mensajes de facilidad H.225.0. El mecanismo de sincronización y de actualización de claves H.245 necesita la tunelización, de utilidad, por ejemplo, en las llamadas de larga duración[11].

En el cuadro E.1, la zona sombreada vertical (amarillo en la copia electrónica) representa el ámbito del perfil de seguridad de firmas. Cuando se omite la integridad, indicada por la zona sombreada horizontal (naranja en la copia electrónica), resulta el perfil de seguridad autenticación solamente. Dentro del perfil de seguridad de firmas cabe elegir entre firmas digitales RSA-SHA-1 o RSA-MD5. El perfil de seguridad de criptación vocal del anexo D (véase D.7) podría utilizarse facultativamente junto con el perfil de seguridad de firmas.

[11] La actualización para la codificación vocal G.711 de seguridad debe producirse a más tardar después de 2^{30} bloques de 64 bits, lo que significa más de 12 días de conversación.

Cuadro E.1/H.235 – Perfil de seguridad de firmas

Servicios de seguridad	Funciones de llamada			
	RAS	H.225.0	H.245[a)]	RTP
Autenticación	SHA1/ MD5	SHA1/ MD5	SHA1/ MD5	
	firma digital	firma digital	firma digital	
No repudio	SHA1/ MD5	SHA1/ MD5	SHA1/ MD5	
	firma digital	firma digital	firma digital	
Integridad	SHA1/ MD5	SHA1/ MD5	SHA1/ MD5	
	firma digital	firma digital	firma digital	
Confidencialidad				
Control de acceso				
Gestión de claves	asignación de certificado	asignación de certificado		

a) H.245 tunelizada o H.245 insertada en conexión rápida H.225.0.

NOTA 1 – El perfil de seguridad de firmas ha de ser soportado también por otras entidades H.235 (por ejemplo, apoderados H.235, controladores de acceso, pasarelas).

NOTA 2 – Los bits de utilización de claves disponibles en el certificado podrían también determinar el servicio de seguridad proporcionado por un terminal (por ejemplo, afirmación del no repudio).

Para la autenticación, el usuario debería utilizar un esquema de firma de claves privadas/públicas. Este esquema proporciona normalmente la mejor integridad y el no repudio de la llamada.

La presente Recomendación no describe los procedimientos para:

- El registro, certificación y asignación de certificados desde un centro de confianza y la asignación de claves privadas/públicas, los servicios de directorio, los parámetros de CA específicos, la revocación de certificados, la actualización/recuperación de pares de claves y otros procedimientos operacionales y de gestión de certificados tales como la entrega de certificados o claves públicas/privadas y la instalación de dichos procedimientos en los terminales.

Tales procedimientos pueden aplicarse por medios que no forman parte del presente anexo.

Las entidades de comunicación involucradas son capaces de determinar implícitamente la utilización, bien de los perfiles de seguridad básicos del anexo D, o bien de este perfil de seguridad de firmas mediante la evaluación de los identificadores de objeto de seguridad señalados en los mensajes (**tokenOID**, y **algorithmOID**; véase también E.18).

E.3 Requisitos H.323

Se supone que las entidades H.323 que implementen este perfil de seguridad soportan las siguientes características:

- Conexión rápida.
- Modelo con encaminamiento por controlador de acceso.

E.4 Servicios de seguridad

En este anexo se utilizan los siguientes términos para la provisión de servicios de seguridad.

- **Autenticación solamente**: Este servicio de seguridad del perfil de seguridad de firmas soporta la autenticación de usuario, en cuyo caso el usuario autentica cuando es aplicada correctamente la firma digital de alguna pieza de datos por la clave privada. Hay que señalar que este servicio de seguridad no proporciona contramedidas frente a operaciones arbitrarias de corte e inserción, manipulación de mensajes o ataques fraudulentos. La autenticación solamente puede ser útil para los apoderados de seguridad que verifican la autenticidad del mensaje (autenticación del origen de los datos) cuando se reenvía[12] el mensaje a otro destino (por ejemplo, un controlador de acceso). No obstante, la autenticación solamente también puede ser aplicada salto por salto. El procedimiento III especifica este servicio de seguridad para un escenario de extremo a extremo mientras que el procedimiento II especifica este servicio de seguridad para el caso salto por salto.
- **Autenticación e integridad**: Éste es un servicio de seguridad combinado que soporta la integridad de los mensajes junto con la autenticación de usuario. El usuario autentica cuando es aplicada correctamente la firma digital de alguna pieza de datos por la clave privada. Además de esto, el mensaje es protegido contra el fraude. Ambos servicios son proporcionados por el mismo mecanismo de seguridad. La autenticación e integridad combinadas sólo son posibles sobre la base de salto por salto. El procedimiento II especifica este servicio de seguridad.

NOTA – Cuando se aplican firmas digitales se puede soportar un servicio de seguridad de no repudio; esto depende también de la fijación de los bits de utilización de claves de la clave de firma en el certificado (véase también RFC 2459).

Las técnicas asimétricas que utilizan firmas digitales se pueden aplicar sobre una base salto por salto y/o también sobre una base de extremo a extremo.

Se describen los siguientes procedimientos para su utilización en este perfil:

El procedimiento II se basa en firmas digitales que utilizan una pareja de claves privada/pública para la provisión de la autenticación, la integridad y el no repudio de mensajes RAS, Q.931 y H.245. Los terminales pueden utilizar este método si se necesita el no repudio y una integridad sofisticada.

Dependiendo de cual sea la política de seguridad, la autenticación puede ser unilateral, o mutua (recíproca) en el caso en que también se aplica la autenticación/integridad en el sentido inverso y se proporciona por tanto una seguridad superior. La política de seguridad de un terminal puede permitir la autenticación solamente sin calcular la integridad criptográfica (véase E.7).

Los controladores de acceso que detectan el fallo de la validación de la autenticación y/o de la integridad en un mensaje RAS/de señalización de llamada recibido de un controlador de acceso par/terminal responden con un mensaje de rechazo correspondiente que indica un fallo de seguridad mediante la fijación de la causa de rechazo a **securityDenial**.

Hay una señalización H.235 implícita para indicar la utilización del procedimiento II y el mecanismo de seguridad aplicado que se basa en el valor de los identificadores de objeto (véase también E.18) y el relleno de los campos del mensaje. En este texto se hace referencia a los identificadores de objeto mediante letras (por ejemplo, "A").

Este perfil no utiliza los campos ICV H.235; en su lugar, los valores de comprobación de la integridad criptográfica son introducidos en el campo **signature** del **token** en el **cryptoSignedToken**.

---

[12] El reenvío generalmente cambia determinadas partes del mensaje; por ello no puede realizarse la seguridad de extremo a extremo.

E.5 Detalles de las firmas digitales con parejas de claves privada/clave pública (Procedimiento II)

Cuando se aplica el procedimiento II para la seguridad salto por salto, deberán adherirse al mismo los siguientes procedimientos:

- Deben utilizarse SHA1 o MD5 junto con el algoritmo RSA para generar la firma digital. La adhesión a PKCS #1 y PKCS #7 facilita el interfuncionamiento a este respecto.

  El campo **CryptoH323Token** de cada mensaje RAS/H.225.0 deberá contener los siguientes campos:

  – **nestedCryptoToken** conteniendo un **CryptoToken** que a su vez contiene el **cryptoSignedToken** con los siguientes campos:

    - **tokenOID** puesto a:
      – "A", que indica que el cálculo de la autenticación/integridad incluye todos los campos del mensaje RAS/H.225.0 (véase E.9);
      – "B", que indica que el cálculo de la autenticación/integridad incluye solamente un subconjunto de campos (véase E.8) del mensaje RAS/H.225.0 para autenticación solamente.
    - **token** conteniendo los campos:
      – **toBeSigned,** que contiene el **EncodedGeneralToken**, el cual es realmente un **ClearToken** con los siguientes campos fijados:
        - **tokenOID** fijado a "S", que indica que se esta utilizando **ClearToken** la autenticación/integridad/no repudio del mensaje.
        - **timeStamp,** que contiene la indicación de tiempo.
        - **random**, que contiene un número secuencial monotónicamente creciente.
        - **generalID,** que contiene el identificador del recipiente (sólo en caso de mensajes de unidifusión).
        - **sendersID,** que contiene el identificador del emisor.
        - **dhkey**, utilizado para transferir los parámetros Diffie-Hellman especificados en esta Recomendación durante **Setup** y **Connect**.
          – **halfkey,** que contiene la clave pública aleatoria de una parte.
          – **modsize**, que contiene el primo DH (véase el cuadro D.4).
          – **generator,** que contiene el grupo DH (véase el cuadro D.4).

          NOTA 1 – Cuando el perfil de seguridad de firmas se utiliza sin el perfil de seguridad de criptación vocal, no es necesario enviar los parámetros Diffie-Hellman; en su lugar, **halfkey**, **modsize** y **generator** pueden fijarse, por simplicidad, a la representación binaria de 0.

          – **certificate,** que contiene el certificado digital del emisor (véase E.12).
        - **algorithmOID** puesto a:
          – "V", que indica el empleo de la firma MD5-RSA;
          – "W", que indica el empleo de la firma SHA1 RSA.
        - **params** fijado a NULO.

- **signature**, que contiene la firma calculada utilizando SHA1o MD5 RSA en todos los campos (si **tokenOID** es "A", véase E.9) o en determinados campos críticos (si **tokenOID** es "B", véase E.8) del mensaje RAS/H.225.0.

Cuando se utiliza el **tokenOID** "A" para la protección de unidades H323-UU-PDUs tunelizadas que incluyen todos los contenidos de mensajes H.245, el cálculo de la firma se realizará sobre el mensaje **H323-UU-PDU** completo con todos los campos, de conformidad con el procedimiento descrito en la sección E.9. En el caso de que se utilice el **tokenOID** "B", la "autenticación solamente" del **CryptoToken** se alcanza cuando se aplica el procedimiento III (véase E.8).

- Una entidad (que puede estar alejada uno o más saltos de aplicación) para la que está destinada la firma, verifica dicha firma.

NOTA 2 – El recipiente es capaz de detectar la aplicación del procedimiento II mediante la evaluación del **algorithmOID** dentro del testigo del **cryptoSignedToken** (detectando la presencia de "V" o de "W").

E.6 Procedimientos para la conferencia multipunto

Las unidades de control multipunto (MCU) deberán soportar la distribución segura de certificados tras la petición efectuada desde los terminales mediante las instrucciones tunelizadas H.245 **ConferenceRequest** y **ConferenceResponse** descritas en 9.1. Esto permite a los terminales solicitar certificados desde otros terminales en un entorno de conferencia multipunto y por tanto obtener la certidumbre acerca de la identidad de los demás participantes en la conferencia.

ConferenceRequest transporta la **requestTerminalCertificate**, de la cual son fijados los siguientes campos:

- **terminalLabel**: utilizado como medio de direccionamiento del terminal distante a través de la MCU;
- **certSelectionCriteria**: el emisor sólo puede pedir certificados de tipos específicos;
- **sRandom**: pregunta aleatoria generada por el emisor de la petición.

ConferenceResponse transporta la **terminalCertificateResponse**, de la cual son fijados los siguientes campos:

- **terminalLabel**: permite asociar el certificado devuelto con el terminal.
- **CertificateResponse**: transporta la respuesta procedente de la MCU con los campos puestos a:
  - **terminalLabel**: identificación del terminal distante
  - **certificateResponse**: es de hecho una cadena de octetos codificada en ASN.1 a partir de la **EncodedReturnSig** como:
    - **generalID**: identificación del terminal de destino;
    - **responseRandom**: valor de la pregunta aleatoria generada por la MCU;
    - **requestRandom**: **sRandom** reproducida;
    - **certificate**: transporta el certificado devuelto donde **type** indica el tipo de certificado como OID y **certificate** cursa el certificado digital (véase E.12).

E.7 Autenticación de extremo a extremo (Procedimiento III)

En la figura E.1 se muestra un escenario con apoderados que separan los GK y los EP y donde se utilizan dos CryptoTokens diferentes para la autenticación salto por salto así como para la autenticación de extremo a extremo y/o la integridad salto por salto. El **CryptoToken** utilizado para autenticación salto por salto se aplica solamente a la rama entre dos entidades y debe ser recalculado en cada una de las demás ramas. Por otra parte, el **CryptoToken** utilizado para la autenticación de

extremo a extremo es generado una sola vez por el punto extremo de emisión y no es modificado en el tránsito por los nodos intermedios. Los nodos intermedios pueden validar firmas y certificados cursados en **CyptoTokens** de extremo a extremo y deben reenviar el **CryptoToken** en tránsito.

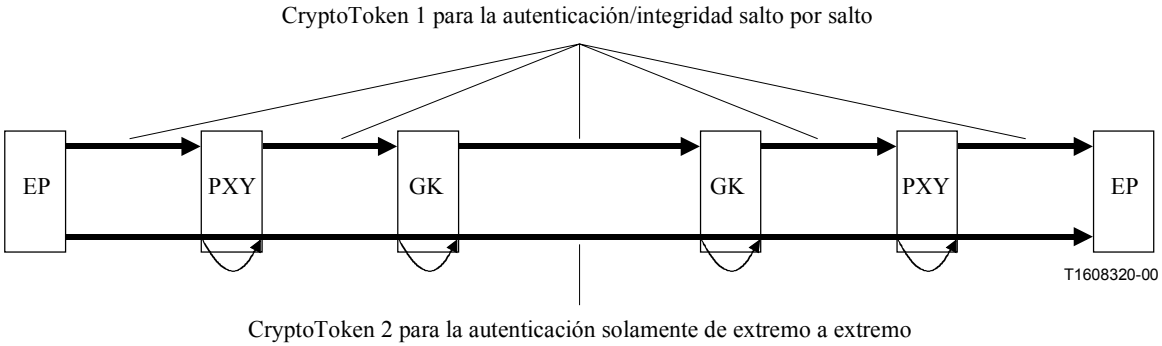


Figura E.1/H.235 – Utilización simultánea de la seguridad salto por salto y la autenticación de extremo a extremo

NOTA 1 – El apoderado puede ser un nodo de red independiente como muestra la figura E.1 o puede estar cosituado con la funcionalidad de una entidad H.323, por ejemplo, como parte del GK.

NOTA 2 – Dependiendo de cual sea el **tokenOID** señalado el apoderado será capaz de determinar si el **CryptoToken** recibido esta destinado al apoderado ("S") o a algún otro recipiente ("R").

NOTA 3 – Debido a que las entidades intermedias modifican el contenido del mensaje de señalización en cada rama, no es posible la integridad de extremo a extremo.

Para la autenticación verdadera de extremo a extremo a través de apoderados H.323 o elementos de red intermedios, el terminal/punto extremo emisor deberá calcular una firma digital como sigue:

El campo **CryptoH323Token** en cada mensaje RAS/H.225.0 deberá contener los siguientes campos:

- **nestedCryptoToken**, que contiene un **CryptoToken** que a su vez contiene el **cryptoSignedToken**, con los siguientes campos:
  - **tokenOID** puesto a:
    - "A", que indica que el cálculo de la autenticación/integridad salto por salto incluye todos los campos del mensaje RAS/H.225.0 (véase E.9).
    - "B", que indica que el cálculo de la autenticación incluye solamente un subconjunto de campos (véase E.8) del mensaje RAS/H.225.0 para autenticación solamente.
- **token**, que contiene los campos:
  - **toBeSigned**, conteniendo el campo **ClearToken** utilizado con los siguientes campos:
    - **tokenOID** puesto a "R", que indica que el **ClearToken** está siendo utilizado para autenticación solamente/no repudio[13] sobre una base de extremo a extremo.
    - **random**, que contiene un número secuencial monotónicamente creciente.
    - **timeStamp**, facultativamente, para una seguridad mejorada solamente cuando las entidades extremo de terminación están sincronizadas en el tiempo.

[13] El servicio de seguridad que se está realmente aplicando depende también de los bits de utilización de claves del certificado.

- **generalID**, que contiene el identificador de punto extremo del recipiente (sólo en el caso de unidifusión). En el caso salto por salto, éste es el identificador del salto siguiente; en el caso de extremo a extremo éste es el identificador de punto extremo del extremo lejano.
- **sendersID**, que contiene el emisor de punto extremo.
- **certificate**, que contiene el certificado digital del emisor, donde **type** indica el tipo de certificado ("V" para certificados MD5-RSA o "W" para certificados SHA1-RSA) y **certificate** transporta el certificado real (véase E.12).
- **dhkey**, utilizado para transferir los parámetros Diffie-Hellman especificados en esta Recomendación durante **Setup** y **Connect**.
  - **halfkey,** que contiene la clave pública aleatoria de una parte.
  - **modsize,** que contiene el primo DH (véase el cuadro D.4).
  - **generator,** que contiene el grupo DH (véase el cuadro D.4).

NOTA 4 – Cuando el perfil de seguridad de firmas se utiliza sin el perfil de seguridad de criptación vocal no es necesario enviar ningún parámetro Diffie-Hellman; en su lugar, **halfkey**, **modsize** y **generator** pueden ser fijados, por simplicidad, a la representación binaria de 0.

- **token**, con los campos:
  - **algoritmOID** puesto a:
    - "V", que indica la utilización de la firma MD5-RSA
    - "W", indicando la utilización de la firma SHA1-RSA
  - **params** puesto a NULO
  - **signature**, que contiene la firma calculada utilizando SHA1-RSA o MD5-RSA en todos los campos (si **tokenOID** es "A") o en determinados campos críticos (si **tokenOID** es "B") del mensaje RAS/H.225.0.

El apoderado puede verificar cualquier certificado y/o firma digital obtenidos, y puede descartar el mensaje si no los considera adecuados de acuerdo con la política local o reenviar más adelante el **CryptoToken** recibido. El apoderado deberá generar nuevos elementos de información de señalización H.235 para la seguridad salto por salto de conformidad con los procedimientos II o III.

La entidad que termina la rama – puede ser un terminal – debe verificar la información de seguridad recibida en el **CryptoToken** y, dependiendo de la presencia de elementos de seguridad de extremo a extremo, puede evaluar adicionalmente la información de **CryptoToken** de extremo a extremo. Los procedimientos de verificación exacta en un terminal o en una entidad H.323 intermedia pueden variar de acuerdo con la política local.

E.8 Autenticación solamente

Los terminales pueden decidir implementar la autenticación solamente (utilizando el OID "B"). En este caso, el autenticador es calculado solamente sobre un subconjunto (**ClearToken** dentro de **CryptoToken**) del mensaje RAS/H.225.0. La autenticación solamente puede se útil para la autenticación de extremo a extremo verdadera (véase E.7). Se utilizan como subconjunto los siguientes campos de la estructura **ClearToken**:

- **tokenOID**: Hay un identificador de objeto de testigo separado (tokenOID "B") para la implementación de la autenticación solamente.
- **random**: El número secuencial monotónicamente creciente.
- **timeStamp**: La indicación de tiempo.

- **generalID**: El identificador del recipiente (sólo en el caso de mensajes unidifusión). En el caso salto por salto, es el identificador del salto siguiente; en el caso de extremo a extremo es el identificador de punto extremo del extremo lejano.
- **sendersID**: El identificador del emisor.
- **dhkey**: Los parámetros Diffie-Hellman. Este campo y subcampos solamente se utilizan durante los mensajes **Setup** y **Connect**.

El autenticador se calcula sobre el **ClearToken** dentro del **EncodedGeneralToken** (es decir, el **ClearToken**) del **token** del **cryptoSignedToken**. La firma digital deberá calcularse sobre la cadena de bits codificada en ASN.1 de **ClearToken**. Antes del cálculo de la firma digital, el **tokenOID** del **ClearToken** deberá ponerse a {0 0}.

E.9 Autenticación e integridad

El procedimiento aplicado para la autenticación e integridad de mensajes sobre todos los campos del mensaje codificado en ASN.1 (utilizando el OID "A") es el siguiente.

El emisor de un mensaje deberá calcula la firma como sigue:

1) Fijará el valor de firma a un esquema por defecto específico de una longitud fija (por ejemplo 1024 bits). Este paso reservará espacio para la longitud máxima de una firma digital que es posible para un certificado determinado. El esquema exacto de bits no importa, pero constituye una buena elección un esquema de bits exclusivo que no ocurra en el resto del mensaje.
2) Codificará en ASN.1 el mensaje completo.
3) Localizará[14] el esquema por defecto en el mensaje codificado; sobrescribirá todo el esquema de bits construido con bits cero.
4) Calculará la firma digital después de la decodificación del mensaje en ASN.1 aplicando el método indicado por **algorithmOID** "V" o "W" (véase E.10).
5) Sustituirá el esquema por defecto en el mensaje codificado por el valor de firma digital calculado. Si la firma digital es más corta que el espacio reservado, deberán colocarse ceros delanteros antes de los bits más significativos del valor de firma.

El recipiente recibe el mensaje y procede como sigue:

1) Decodifica en ASN.1 el mensaje.
2) Extrae el valor de la firma digital recibida y lo guarda en un SV variable local.
3) Busca y localiza el valor de firma SV en el mensaje codificado recibido.

NOTA – En las ocasiones poco frecuentes en que la subcadena del valor de firma puede aparecer varias veces en el mensaje completo, se han de repetir sucesivamente los pasos 3-6 con una posición de arranque de búsqueda diferente.

4) Sobrescribe el esquema de bits en el mensaje codificado todo con ceros.
5) Calcula la firma digital tras el mensaje codificado aplicando el método indicado por el **algorithmOID** "V" o "W" (véase E.10).
6) Compara SV con el valor de firma calculado. El mensaje sólo es considerado incorrupto y auténtico si ambos valores de firma son iguales; en este caso la autenticación ha tenido éxito y el procedimiento se detiene.

[14] Esto puede implicar algunos pasos de prueba y error en el caso poco frecuente de que el esquema por defecto aparezca más de una vez en el mensaje.

7) En caso contrario, repite los pasos 3-7 restableciendo SV a la situación anterior y busca otra concordancia. Si ninguna de las concordancias arroja una comparación correcta de los valores de firma, la autenticación ha fracasado y el mensaje ha sido alterado (accidental o deliberadamente) durante el tránsito o por algún otro motivo.

E.10 Cálculo de la firma digital

La entrada al proceso de generación de firma digital es una cadena de bits codificada en ASN.1 que incluye el resultado del proceso de cálculo resumido del mensaje y la clave privada del firmante. Los detalles de la generación de la firma digital dependen del algoritmo de firma utilizado; el certificado determina el algoritmo de firma que ha de aplicarse; cuando la extensión de utilización de claves en el certificado está presente, el bit **digitalSignature** debe ser fijado para la clave deseable para la firma. El valor de firma generado por el firmante se codifica como una cadena de bits y es cursado en el campo **signature**.

Deberá utilizarse el método descrito en [PKCS #1, sección E.8.1.1] para el cálculo de una firma digital basada en RSA con apéndice (RSASSA-PKCS1-v1_5-SIGN) junto con los procedimientos OS2IP, RSASP1 y I2OSP y el método de codificación EMSA-PKCS1-v1_5.

E.11 Verificación de la firma digital

La entrada al proceso de verificación de firma incluye el resultado del proceso de cálculo resumido del mensaje y la clave pública del firmante. El recipiente puede obtener la clave pública correcta para el firmante por cualquier medio, pero el método preferido consiste en la obtención de un certificado a partir del campo **certificate** y la validación posterior utilizando el troceado del certificado del firmante. La validación de la clave pública del firmante puede basarse en el procesamiento del trayecto de certificación (RFC 2459). Los detalles de la verificación de firma dependen del algoritmo de firma empleado.

Deberá utilizarse el método descrito en [PKCS #1, sección E.8.1.2] para la verificación de una firma digital basada en RSA con apéndice (RSASSA-PKCS1-v1_5-VERIFY) junto con los procedimientos OS2IP, RSAVP1 y I2OSP y el método EMSA-PKCS1-v1_5-ENCODE.

E.12 Tratamiento de los certificados

Para la verificación de las firmas digitales, la entidad receptora debe tener acceso al certificado del emisor que está firmado por una autoridad de certificación (CA, *certification authority*) reconocida. El recipiente puede acceder al certificado del emisor de varias formas:

- El certificado está incluido en el intercambio de mensajes como se describe en los procedimientos II y III.
- El recipiente conoce el certificado; esté puede encontrarse almacenado en local procedente de un intercambio anterior.
- En vez de incluir el certificado propiamente dicho, el emisor proporciona una URL en la cual donde puede hallarse el certificado. A este fin, **certificate** contiene la URL y **type** es fijado al OID "P".
- El recipiente obtiene el certificado por otros medios distintos a los de la presente Recomendación (por ejemplo, por consulta al directorio LDAP).

Los procedimientos II y III proporcionan los medios de transportar un certificado digital. En aras de la eficacia, los certificados digitales de las entidades habrán de transmitirse a lo sumo una sola vez si no están ya disponibles en las entidades en virtud de la aplicación de otros medios distintos de los de esta Recomendación. El intercambio de certificados debería por tanto producirse solamente al principio del establecimiento de una comunicación: para RAS esto sucede durante el descubrimiento del controlador de acceso o, si esta fase se omite, durante el registro del controlador de acceso.

Ocurre de manera análoga en la conexión rápida, donde el certificado puede ser incluido en los mensajes de señalización de llamada iniciales pero ser omitido sin riesgo en los mensajes de señalización de llamada posteriores.

Para este perfil de seguridad, deberá utilizarse el certificado X.509v3 (1997). Quedan en estudio otros formatos de certificado.

E.13 Ilustración del empleo del procedimiento II

Consideremos el caso de la figura E.2, donde cada entidad tiene su propio certificado/pareja clave pública-clave privada. Una entidad puede también poseer múltiples parejas de claves. En la figura, un apoderado H.323 separa EP1 de GK1.

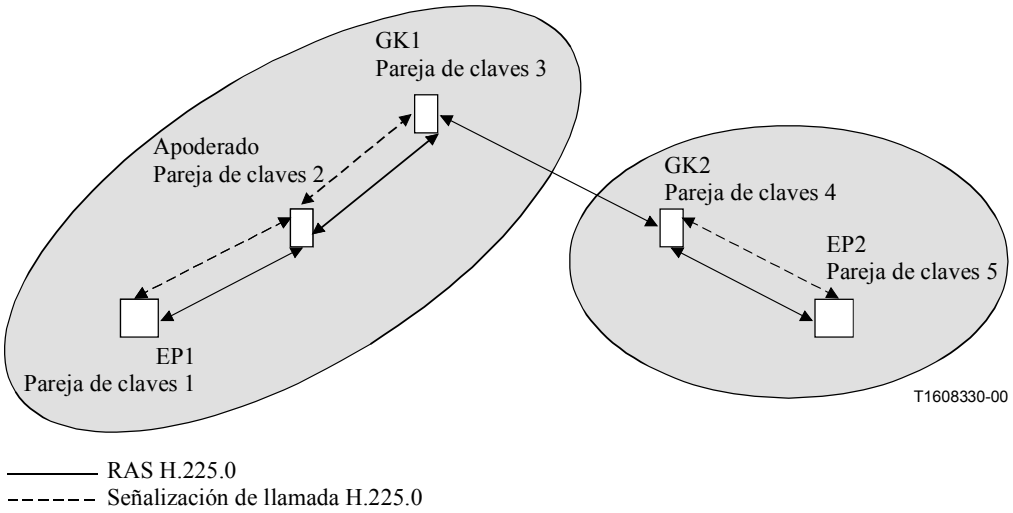


Figura E.2/H.235 – Ilustración de la utilización de claves públicas en un modelo encaminado por GK-GK

El apoderado H.323 actúa doblemente: Por un lado, el apoderado finaliza la autenticación e integridad de cada una de sus ramas. El apoderado incluye, en tiempo real, la información de autenticación/integridad calculada recientemente en los mensajes RAS de salida, de un modo análogo al descrito en el procedimiento I del anexo D. Por otro lado, el apoderado permite que la información de seguridad de extremo a extremo pase sin modificación. El apoderado puede, sin embargo, verificar los certificados recibidos y/o las firmas digitales en tránsito.

Más adelante se dan los detalles del procedimiento para la autenticación, integridad y no repudio de mensajes RAS, H.225.0 y H.245.

E.13.1 Autenticación, integridad y no repudio de mensajes RAS

Consideremos el caso de una comunicación salto por salto donde EP1 desea enviar un mensaje RAS – por ejemplo, un mensaje **ARQ** – a GK1. EP1 genera un indicación de tiempo y un número secuencial y los incluye en los campos **timeStamp** y **random** respectivamente, junto con el alias del apoderado en el campo **generalID** y el **sendersID** de EP1. Estos campos están presentes en el campo **ClearToken** del **EncodedGeneralTokens** presente en el **token** del **cryptoSignedToken** del campo **CryptoToken** del **cryptoH323Token** del mensaje **ARQ**. Este **cryptoH323Token** es uno de, por lo menos, varios testigos de la secuencia **cryptoTokens**. El **tokenOID** dentro del **cryptoSignedToken** se fija a "A", indicando con ello que todos los campos del mensaje **ARQ** están firmados. El **token** en **cryptoSignedToken** tiene el **algorithmOID** puesto a "V", indicando que se utiliza MD5-RSA, o el **algorithmOID** puesto a W", indicando que se utiliza SHA1-RSA, y **params** puesto a NULO. EP1

calcula entonces la firma basada en el algoritmo de firma dado utilizando su clave privada. La firma se calcula sobre todos los campos del mensaje **ARQ** cuando el **tokenOID** está puesto a "A". EP1 incluye la firma calculada dentro de **signature** en el campo **token** del campo **cryptoSignedToken** del **CryptoToken** presente en el **cryptoH323Token** del mensaje **ARQ**, e incluye su certificado en el campo **certificate**.

De manera análoga, en la comunicación de extremo a extremo a través de un apoderado, EP1 genera otro **CryptoToken** conteniendo una firma digital que cubre determinados campos críticos (véase E.7) en el **ClearToken** del mensaje **ARQ**. El **tokenOID** en el **CryptoSignedToken** se fija a "B", indicando la autenticación solamente de este **ClearToken**; fija **tokenOID** en el **ClearToken** a "R", indicando la autenticación de extremo a extremo. Asimismo **timeStamp**, **random**, **sendersID**, **generalID** y, en el caso de que éste sea un **SETUP/CONNECT**, también **dhkey**, fijan en **token** los siguientes campos: **algorithmOID** a "V" o "W", indicando el algoritmo de firma, **params** a NULO y **signature** a la firma digital calculada sobre los campos **ClearToken**. El **certificate** transporta el certificado digital de EP1. El mensaje **ARQ** es entonces enviado al apoderado.

Tras la recepción del mensaje **ARQ**, el apoderado verifica la firma de los testigos que están dirigidos a él (en este caso, por ejemplo, los que tienen un **tokenOID** "A"). Esta verificación se basa en varios criterios, que incluyen:

- Vida de la identificación de tiempo y unicidad de **random**.
- Identidad del **generalID** e identificador propio.
- Autorizaciones de acceso para los **sendersID**.
- Concordancia de la firma del mensaje **ARQ** con la firma calculada por GK1.
- Verificación de los parámetros Diffie-Hellman, por ejemplo, comprobando si el primo de 1024 bits y el generador son correctos. La comprobación de la seguridad de los parámetros DH se realiza al terminar el proceso, y sólo puede efectuarse cuando la política local lo requiere.
- Verificación del certificado recibido.

Si la verificación de la firma ha tenido éxito, el apoderado calcula una nueva firma para insertarla (sustituirla) en el mensaje **ARQ** antes de reenviar éste al GK1 como sigue. El apoderado sustituye los campos **timeStamp**, **random**, **sendersID** y **generalID** en el campo **ClearToken** (**toBeSigned**) utilizando valores pertinentes a la rama apoderado-GK1. El campo **timestamp** contiene la indicación de tiempo actual, el campo **random** contiene el siguiente número secuencial monotónicamente creciente para la rama apoderado-GK1, el **sendersID** del apoderado y el campo **generalID** contienen el alias de GK1. El apoderado calcula entonces una nueva firma para este mensaje **ARQ** utilizando su clave privada y el algoritmo de firma, la inserta en **signature** dentro de **token** y añade su **certificate**. El apoderado incluye también el **CryptoToken** de extremo a extremo recibido con su **ClearToken** en el nuevo mensaje saliente y pasa el mensaje **ARQ** al GK1. La firma calculada por EP1 basándose en campos seleccionados del mensaje **ARQ** (**tokenOID** de "B") y que no estaba destinada al apoderado, se envía también a GK1 sin modificación en el mensaje **ARQ**.

Tras la recepción del mensaje **ARQ**, GK1 verifica las firmas, calcula una nueva firma después de modificar adecuadamente los campos **ClearToken** en el **toBeSigned**, la inserta en el campos **signature**, añade su **certificate** y pasa el mensaje **Setup** al EP2. Nuevamente, GK1 debe enviar cualquier información de extremo a extremo recibida en el **CryptoTokens** separado al par GK2 mediante la inclusión de esta información en un **CryptoToken** separado sin modificar.

E.13.2 Autenticación solamente de mensajes RAS

Consideremos el caso de una comunicación salto por salto donde EP1 desea enviar un mensaje RAS – por ejemplo, un mensaje **ARQ** – a GK1. EP1 genera una indicación de tiempo y un número secuencial y los incluye en los campos **timeStamp** y **random** respectivamente, junto con el alias del apoderado en el campo **generalID** y el id de EP en el **sendersID**. Estos campos están presentes en el

campo **ClearToken** del **toBeSigned** presente en el **token** de **cryptoSignedToken** del campo **CryptoToken** del **cryptoH323Token** del mensaje **ARQ**. El **tokenOID** dentro del **cryptoSignedToken** es fijado a "B", indicando con ello que solamente los campos del subconjunto especificado en el **ClearToken** están firmados. El **token** en **cryptoSignedToken** tiene el **algorithmOID** puesto a "V", para indicar la utilización de MD5-RSA, o a "W", indicando la utilización del algoritmo de firma SHA1-RSA, y **params** puesto a NULO. EP1 calcula entonces la firma basada en el algoritmo de firma utilizando su clave privada. La firma se calcula sobre los campos **ClearToken** especificados del mensaje **ARQ**. EP1 incluye la firma calculada dentro de **signature** en el campo **token** del campo **cryptoSignedToken** del **CryptoToken** presente en el **cryptoH323Token** del mensaje **ARQ** y añade su **certificate**.

De manera análoga, EP1 genera otra firma digital para la autenticación de extremo a extremo que cubre determinados campos **ClearToken** en un **CryptoToken** separado en el mensaje **ARQ**. Es incluida esta firma digital (identificada por el **tokenOID** "V" o "W"). El mensaje **ARQ** es enviado entonces al apoderado.

Tras la recepción del mensaje **ARQ**, el apoderado verifica la firma de los testigos que están dirigidos a él (en este caso, por ejemplo, los que tienen un **tokenOID** "B"). Esta verificación se basa en varios criterios que incluyen:

- Vida de la identificación de tiempo y unicidad de **random**.
- Identidad del **generalID** e identificador propio.
- Autorizaciones de acceso para los **sendersID**.
- Concordancia de la firma del mensaje **ARQ** con la firma calculada por GK1.
- Verificación del certificado recibido.

Si la verificación de la firma ha tenido éxito, el apoderado calcula una nueva firma para insertarla (sustituirla) en el mensaje **ARQ** antes de reenviar éste al GK1 como sigue. El apoderado reemplaza los campos **timeStamp**, **random**, **sendersID** y **generalID** en el campo **ClearToken** de **toBeSigned** utilizando valores pertinentes a la rama apoderado-GK1. El campo **timestamp** contiene la indicación de tiempo actual, el campo **random** contiene el siguiente número secuencial monotónicamente creciente para la rama apoderado-GK1 y el campo **generalID** contiene el alias de GK1. El apoderado calcula entonces una nueva firma para este **ClearToken** utilizando su clave privada y el algoritmo de firma MD5-RSA o SHA1-RSA (**algorithmOID** = "V" o "W"), la inserta en **signature** dentro de **token** de **cryptoSignedToken**, añade su **certificate** y pasa el mensaje **ARQ** al GK1. La firma calculada por EP1 basándose en campos **ClearToken** seleccionados del mensaje **ARQ** (**tokenOID** de "B") y que no estaba destinada al apoderado, se envía también a GK1 sin modificación en el mensaje **ARQ**.

Tras la recepción del mensaje **ARQ**, GK1 verifica la firma, calcula una nueva firma después de la modificación adecuada de los campos **ClearToken** en **toBeSigned**, la inserta en el campo **signature** y pasa el mensaje **Setup** al EP2. La información de firma de extremo a extremo del EP1 es incluida sin modificación en el mensaje **Setup**.

E.13.3 Autenticación, integridad y no repudio de mensaje H.225.0

El procedimiento aplicable a los mensajes H.225.0 es idéntico al de los mensajes RAS. La única diferencia estriba en que el conjunto de campos que han de firmarse ha de ser identificado para cada mensaje H.225.0 cuando el **tokenOID** está puesto a "B".

E.13.4 Autenticación e integridad de los mensajes H.245

Consideremos el caso en el que EP1 desea enviar un mensaje H.245 – por ejemplo, un mensaje **TerminalCapabilitySet** – a EP2. EP1 comprueba si se necesita enviar un mensaje H.225.0 al apoderado. En caso afirmativo, el mensaje H.245 es tunelizado dentro de este mensaje H.225.0. Los campos en el mensaje H.225.0 son fijados del modo descrito anteriormente para la transmisión de un mensaje H.225.0. Puesto que el mensaje H.245 es tunelizado, **h323-uu-pdu** en el mensaje **h323-UserInformation** tiene sus campos fijados como sigue:

- el campo **h323-message-body** es puesto al tipo de mensaje H.225.0 que se está transmitiendo.
- **h245Tunneling** se pone a VERDADERO (TRUE).
- **h245Control** contiene la cadena de octetos PDU H.245.

Sin embargo, si no hay pendiente ninguna transmisión de mensaje H.225.0, el mensaje H.245 es entonces tunelizado dentro del mensaje **facility** H.225 ad-hoc. La **h323-uu-pdu** en el mensaje **h323-UserInformation** tiene sus campos fijados como sigue:

- el mensaje **h323-message-body** es puesto a **facility**, que contiene:
  - **reason** puesto a **undefinedReason**;
  - **tokens** y **cryptoTokens** fijados como para cualquier mensaje H.225.0.
- **h245Tunneling** puesto a VERDADERO (TRUE).
- **h245Control** contiene la cadena de octetos PDU H.245.

El mensaje **facility** es a continuación transmitido por EP1 al apoderado.

En cualquiera de los dos casos (si está pendiente la transmisión de un mensaje H.225.0 ó si se utiliza un mensaje **facility** H.225.0 ad hoc), el apoderado verifica la firma destinada para él (representada en este caso por el **tokenOID** "A") tras la recepción del mensaje. A continuación, si está pendiente la transmisión de un mensaje H.225.0 para la rama apoderado-GK, el mensaje H.245 es tunelizado dentro de este mensaje; en caso contrario, es tunelizado dentro de un mensaje **facility** H.225.0 ad hoc. Como en el caso de la transmisión de un mensaje H.225.0, se calcula una nueva firma para el mensaje H.225.0 antes de su transmisión desde el apoderado al GK1. La firma que fue enviada desde el EP1 al apoderado y que no estaba destinada a este último es transferida del apoderado al GK1 sin modificación.

Esta cláusula proporciona un resumen de cómo, y mediante qué métodos, el perfil de firmas asegura los distintos mensajes de señalización H.323.

E.14 Compatibilidad con la Versión 1 de H.235

Si bien estos perfiles de seguridad se han desarrollado pensando en la H.235 versión 2 [H.235 (2000)], se pueden también aplicar a la H.235 versión 1 [H.235 (1998)] con algunas modificaciones menores. Un recipiente es capaz de detectar la presencia de la versión de protocolo H.235 mediante la evaluación de los identificadores de objeto del perfil de seguridad (véase la sección E.18).

Implementaciones de H.235 version 1 [H.235 (1998)]:

- no fijar o evaluar el **sendersID** en el **ClearToken**.

E.15 Comportamiento multidifusión

Los mensajes multidifusión H.225.0, tales como **GRQ** o **LRQ** deberán incluir un **CryptoToken** de conformidad con los procedimientos II y III, donde el **generalID** no está fijado. Cuando tales mensajes son enviados en unidifusión, el mensaje incluirá un **CryptoToken**.

E.16 Lista de mensajes de señalización seguros

E.16.1 RAS H.225

Mensaje RAS H.225.0	Campos de señalización H.235	Autenticación solamente	Autenticación e integridad	No repudio
Cualquiera	cryptoTokens	Procedimiento II/III	Procedimiento II/III	Procedimiento II/III

NOTA – Para los mensajes de unidifusión, se deberán aplicar los procedimientos II y III con los campos de seguridad en el **CryptoToken** utilizado.

E.16.2 Señalización de llamada H.225.0

Mensaje de señalización de llamada H.225.0	Campos de señalización H.235	Autenticación solamente	Autenticación e integridad	No repudio
UUIE Aviso, UUIE Llamada en curso, UUIE Conexión, UUIE Establecimiento, UUIE Facilidad, UUIE Progresión, UUIE Información, UUIE Liberación completa	cryptoTokens	Procedimiento II/III	Procedimiento II/III	Procedimiento II/III

E.17 Utilización de sendersID y generalID

El **ClearToken** guarda los campos **sendersID** y **generalID**. Cuando se dispone de información de identificación, el **sendersID** debe igualarse al identificador del controlador de acceso (GKID) para los mensajes iniciados por el controlador de acceso y al identificador de punto extremo (EPID) para los mensajes iniciados por el punto extremo. Cuando se dispone de información de identificación, el **generalID** debe igualarse al GKID para los mensajes iniciados por el punto extremo y al EPID para los mensajes iniciados por el controlador de acceso. Cuando no se dispone de información de identificación o cuando la radiodifusión/multidifusión es ambigua es porque falta el campo o porque éste debería contener una cadena nula. El cuadro E.2 resume la situación:

Cuadro E.2/H.235 – Identificadores de objeto usados por el anexo E

Mensaje	sendersID	generalID
GRQ unidifusión	**EPID** si está disponible, en su defecto **NULL**	**GKID**
GRQ multidifusión	**EPID** si está disponible, en su defecto **NULL**	
GCF, GRJ	**GKID**	**EPID** si está disponible, en su defecto **NULL**
RRQ inicial		**GKID**
RCF	**GKID**	**EPID**
RRJ	**GKID**	
URQ, UCF, URJ, BRQ, BCF, BRJ, DRQ, DCF, DRJ, NSM, RIP, SCI, SCR, XRS (EP a GK)	**EPID**	**GKID**
URQ, UCF, URJ, BRQ, BCF, BRJ, DRQ, DCF, DRJ, NSM, RIP, SCI, SCR, XRS (GK a EP)	**GKID**	**EPID**
ARQ, IRQ, RAI	**EPID**	**GKID**
ACF, ARJ, BCF, LCF, LRJ, IRR, IRQ, RAC, LCF, LRJ, IACK, INAK	**GKID**	**EPID**
LRQ unidifusión (EP a GK)	**EPID**	**GKID**
LRQ unidifusión (GK a GK)	**GKID**	**GKID**
LRQ multidifusión	**EPID**	
NOTA – GKID es el identificador del controlador de acceso, EPID es el identificador de punto extremo. Un espacio en blanco equivale una cadena de identificación faltante o nula.		

E.18 Lista de identificadores de objeto

En el cuadro E.3 se presenta una lista de todos los OID referenciados (véase también [OIW] y [WEBOIDs]). Hay identificadores de objeto para H.235v1 [H.235 (1998)] y para H.235v2 [H.235 (2000)].

Cuadro E.3/H.235 – Identificadores de objeto utilizados por el anexo E

Referencia de identificador de objeto	Valor(es) del identificador de objeto	Descripción
"A"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 1} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 1}	Utilizado en el procedimiento II para el CryptoToken-tokenOID indicando que la firma incluye **todos** los campos del mensaje RAS/H.225.0 (autenticación e integridad).
"B"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 2} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 2}	Utilizado en el procedimiento II para el CryptoToken-tokenOID indicando que la firma incluye un **subconjunto** de campos del mensaje RAS/H.225.0 (ClearToken) para terminales de autenticación solamente sin integridad.
"P"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 4} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 4}	Utilizado en los procedimientos II o III para indicar que el campo **certificate** transporta una URL.
"R"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 3} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 3}	Utilizado en el procedimiento II para el ClearToken-tokenOID indicando que el ClearToken está siendo utilizado para la autenticación/integridad de extremo a extremo.
"S"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 7} {itu-t (0) recommendation (0) h (8) 235 version (0) 1 7}	Utilizado en el procedimiento II, este OID de testigo indica la autenticación, integridad y no repudio del mensaje.
"V"	{iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 4}	Utilizado en el procedimiento II como OID de algoritmo indicando el empleo de la firma digital MD5 RSA.
"W"	{iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 5}	Utilizado en el procedimiento II como OID de algoritmo indicando el empleo de la firma digital SHA1 RSA.

APÉNDICE I

Detalles de las implementaciones H.323

I.1 Métodos de relleno de texto cifrado

En **[Schneier]**, páginas 191 y 196, hay una descripción de apropiación de texto cifrado. Las figuras I.1 a I.5 ilustran la técnica.

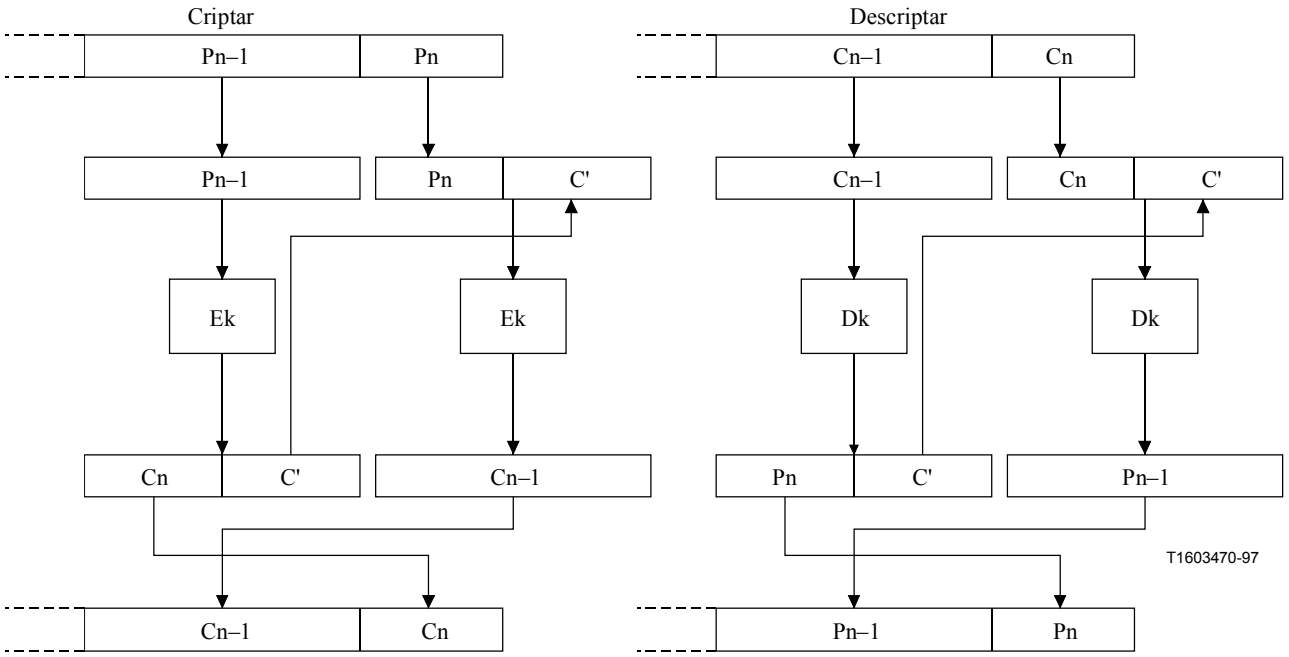


Figura I.1/H.235 – Apropiación de texto cifrado en modo ECB

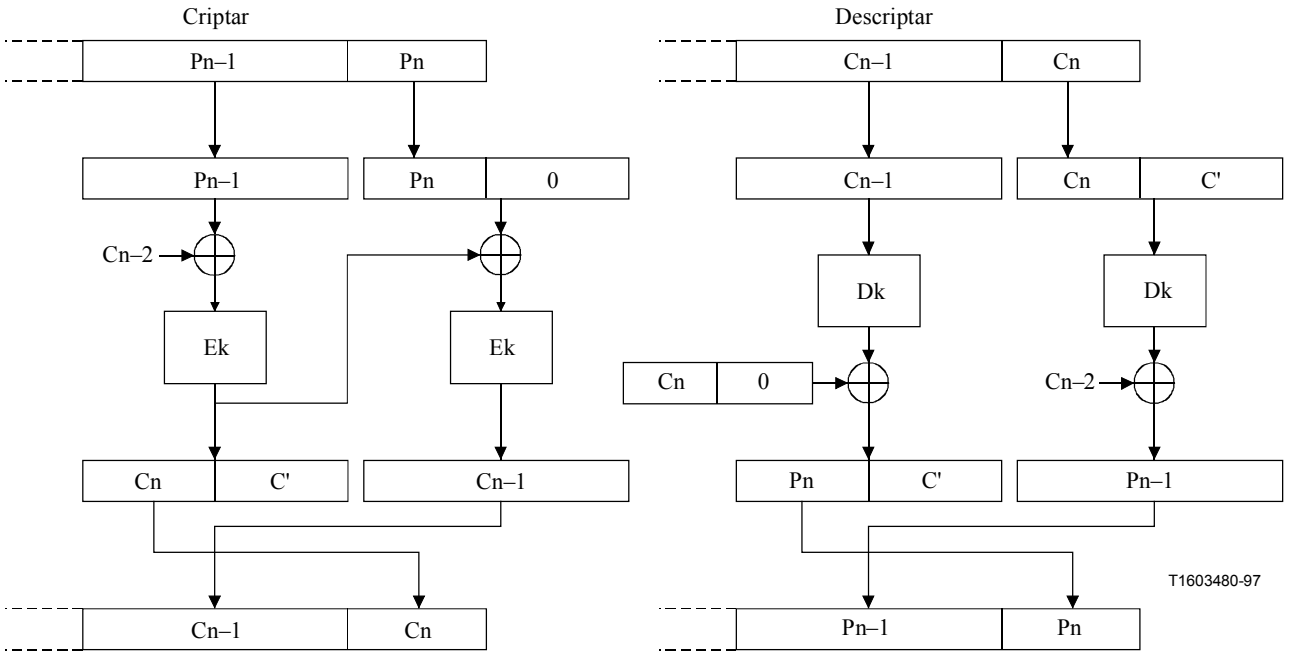


Figura I.2/H.235 – Apropiación de texto cifrado en modo CBC

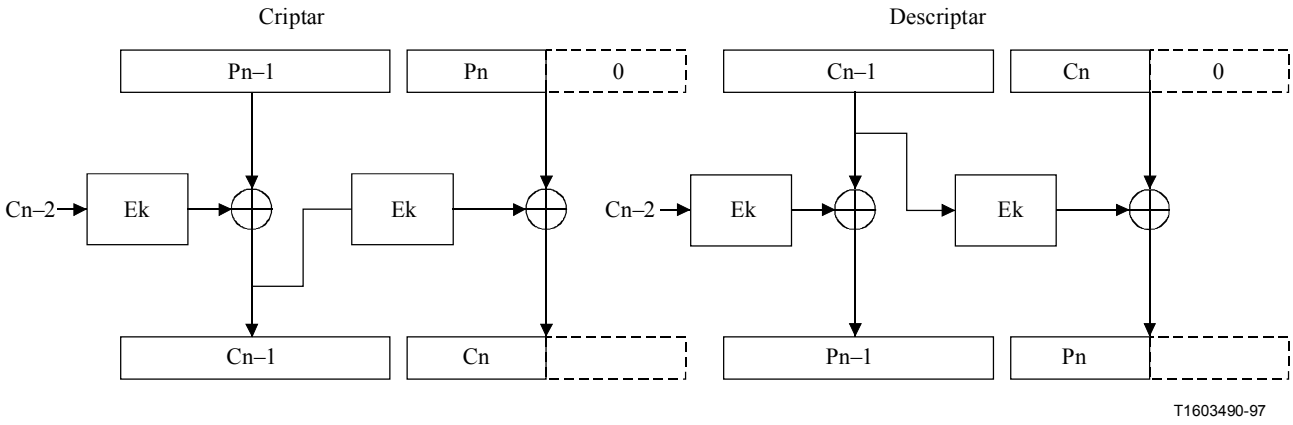
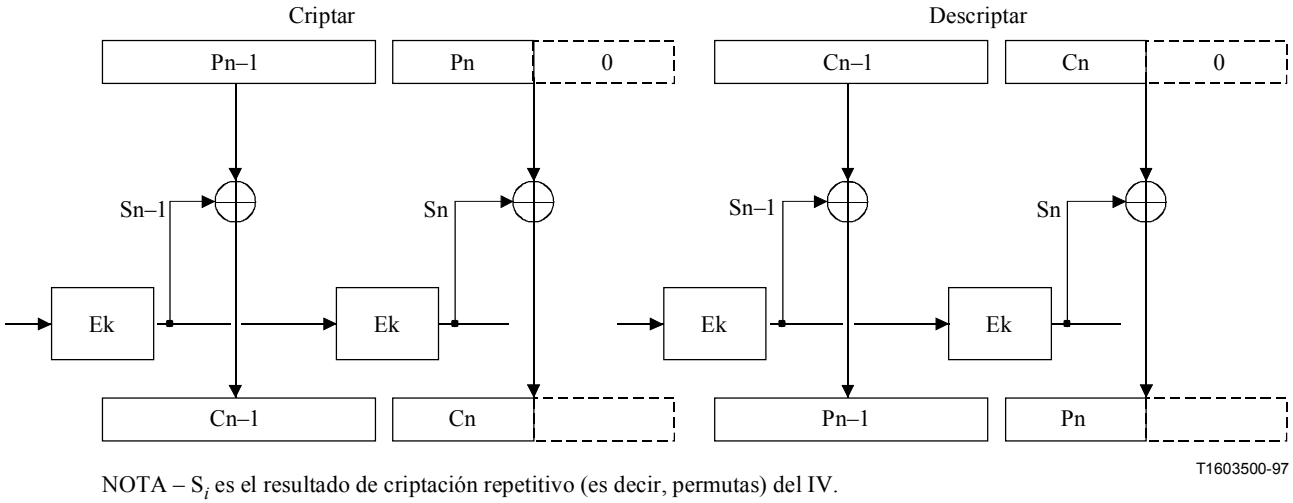


Figura I.3/H.235 – Relleno de ceros en modo CFB



NOTA – S_i es el resultado de criptación repetitivo (es decir, permutas) del IV.

Figura I.4/H.235 – Relleno de ceros en modo OFB

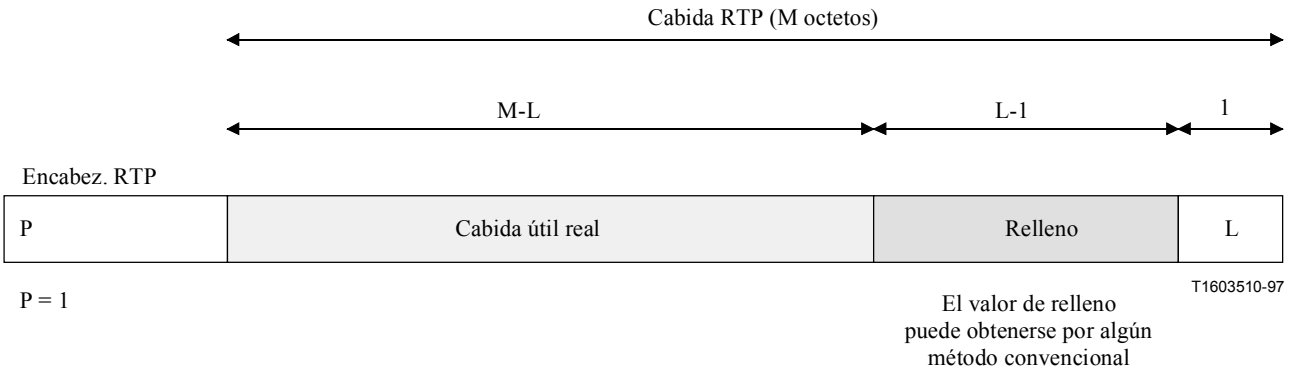


Figura I.5/H.235 – Relleno prescrito por RTP

I.2 Nuevas claves

Los procedimientos indicados en 8.5/H.323 son completados por un MC para sacar a un participante de la conferencia. El terminal director puede generar nuevas claves de criptación para los canales lógicos (y no distribuirlas a la parte eliminada); esto se puede utilizar para evitar que la parte eliminada supervise los trenes de medios.

I.3 Elementos de confianza H.323

En general, las MC(U), las pasarelas y los controladores de acceso (si se aplica el modelo con encaminamiento por controlador de acceso) son fiables con respecto a la privacidad del canal de control. Si el canal de establecimiento de la conexión (H.225.0) es seguro *y* es encaminado a través del controlador de acceso, se debe considerar también de confianza. Si algunos de estos componentes H.323 deben funcionar en los trenes de medios (es decir, mezcla, transcodificación), por definición, serán considerados también de confianza para la privacidad de los medios.

Se puede confiar también en los apoderados/cortafuegos (aunque no son elementos específicos de H.323), porque terminan conexiones, y pueden tener que manipular los mensajes y los trenes de medios.

I.4 Ejemplos de implementaciones

A continuación se describen ejemplos de implementaciones que pudieran ser desarrolladas dentro del protocolo H.235. No se pretende restringir las muchas otras posibilidades disponibles dentro de esta Recomendación, sino más bien dar ejemplos más concretos de utilización dentro de UIT-T H.323.

I.4.1 Testigos

Esta cláusula describe un ejemplo de utilización de testigos de seguridad para oscurecer u ocultar la información de direccionamiento de destino. El caso de ejemplo es un punto extremo que desea hacer una llamada a otro punto extremo utilizando su alias conocido. Más concretamente, esto comprende un punto extremo H.323, un controlador de acceso, una pasarela POTS y un teléfono como se ilustra en la figura I.6.

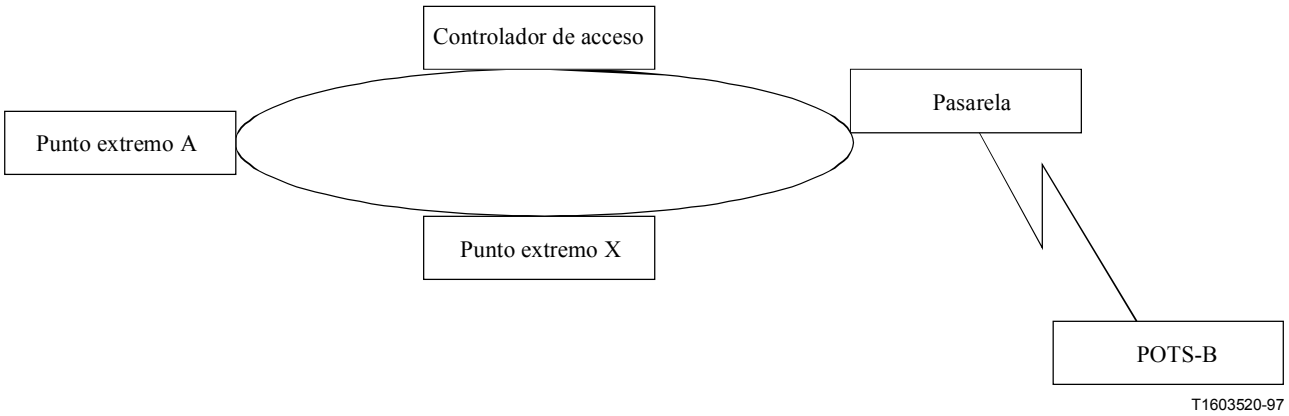


Figura I.6/H.235 – Testigos

Actualmente, el protocolo H.323 puede funcionar de manera similar a una red telefónica con el ID del llamante. Este caso ilustra una situación en la cual la parte *llamada* no desea exponer su dirección física, a la vez que permite que se complete la llamada. Esto puede ser importante en pasarelas POTS-H.323, cuando el número telefónico deseado puede tener que permanecer privado.

Se supone que EPA está tratando de llamar a POTS-B y POTS-B no desea exponer su número telefónico E.164 a EPA. (La manera en que se establece esta política está fuera del alcance de este ejemplo.)

- EPA enviará ARQ a su controlador de acceso para resolver la dirección del teléfono POTS representada por su alias/pasarela. El controlador de acceso reconocerá esto como un alias "privado" sabiendo que para completar la conexión debe devolver la dirección de pasarela de POTS (de manera similar a la devolución de la dirección H.320 si un punto extremo H.320 es llamado por un punto extremo H.323).
- En el ACF devuelto, el controlador de acceso devuelve la dirección de pasarela de POTS según lo previsto. La información de direccionamiento requerida para marcar el teléfono del extremo (es decir el número telefónico) es devuelta en un testigo criptado incluido en ACF. Este testigo criptado contiene el número telefónico E.164 real del teléfono que no puede ser descifrado ni comprendido por el llamante (es decir, EPA).
- El punto extremo emite el mensaje ESTABLECIMIENTO al dispositivo de pasarela (cuya dirección de señalización de llamada fue devuelta en ACF) incluidos los testigos opacos que recibió con ACF.
- La pasarela, al recibir el mensaje ESTABLECIMIENTO, emite su ARQ a su controlador de acceso incluidos cualesquiera testigos que fueron recibidos en el mensaje ESTABLECIMIENTO.
- El controlador de acceso puede descifrar el testigo o testigos y devolver el número telefónico en ACF.

A continuación se muestra la ASN.1 parcial de la estructura de un testigo de ejemplo, describiendo el contenido de campo. Se supone que se utiliza **testigo general codificado en cifra (cryptoEncodedGeneralToken)** para contener el número telefónico criptado.

Una implementación pudiera elegir un **OID de testigo (tokenOID)** que indica que este testigo contiene el número telefónico E.164. El método particular que se utiliza para cifrar este número telefónico (por ejemplo, DES de 56 bits) se incluiría en el **OID de algoritmo (algorithmOID)** de la definición de "CRIPTAR".

```
CryptoToken::= CHOICE
{
        cryptoEncodedGeneralToken SEQUENCE   -- General purpose/application specific token
        {
                tokenOID  OBJECT IDENTIFIER,
                ENCRYPTED { EncodedGeneralToken }
        },
.
.
. [abbreviated text]
.

}
```

El **testigo cifrado (CryptoToken)** se transferiría en los mensajes ESTABLECIMIENTO (del EPA a la pasarela) y **ARQ** (de la pasarela al controlador de acceso) como se indica anteriormente. Una vez que el controlador de acceso decriptó el testigo (el número telefónico) transferirá la versión clara en el **testigo claro (ClearToken)**.

I.4.2 Utilización de testigos en los sistemas H.323

Ha habido alguna confusión en la utilización de **CryptoH323Tokens** individuales pasados en mensajes RAS. Existen dos categorías principales de **CryptoH323Tokens**: los utilizados para los procedimientos H.235 y los utilizados en un modo específico de la aplicación. El uso de estos testigos debe adecuarse a las siguientes reglas:

- Todos los definidos de H.235 (por ejemplo, **cryptoEPPwdHash**, **cryptoGKPwdHash**, **cryptoEPPwdEncr**, **cryptoGKPwdEncr**, **cryptoGKCert**, y **cryptoFastStart**), se deberán utilizar con los procedimientos y algoritmos descritos en esta Recomendación.
- El uso propietario o específico de la aplicación de los testigos deberá utilizar el **nestedcryptoToken** para sus intercambios.
- Cada **nestedcryptoToken** utilizado debe tener una **tokenOID** que lo identifique inequívocamente.

I.4.3 Utilización del valor aleatorio H.235 en sistemas H.323

El valor aleatorio que se pasa en la secuencia xRQ/xCF entre puntos extremos y controladores de acceso puede ser actualizado por el controlador de acceso. Como se describe en B.4.2, este valor aleatorio puede ser renovado en cualquier mensaje xCF para ser utilizado por un mensaje xRQ subsiguientes procedente del punto extremo. Como pueden perderse mensajes RAS (incluidos xCF/xRJ), el valor aleatorio actualizado también puede perderse. La recuperación desde esta situación puede consistir en la reinicialización del contexto de seguridad, pero se deja a la implementación local.

Las implementaciones que requieren la utilización de múltiples peticiones RAS pendientes estarán limitadas por la actualización de los valores aleatorio utilizadas en cualquier autenticación. Si la actualización de este valor se produce con cada respuesta a una petición, no están permitidas las peticiones en paralelo. Una solución posible a esta situación es disponer de una "ventana" lógica durante la cual un valor aleatorio permanece constante. Este tema es incumbencia de la implementación local.

I.4.4 Contraseña

En este ejemplo, se supone que el usuario está abonado al controlador de acceso (es decir, el usuario estará en su zona) y tiene un ID de abono y una contraseña asociada. El usuario se registrará con el controlador de acceso utilizando el ID de abono (transferido en un alias – H323ID) y criptando una cadena de preguntas presentada por el controlador de acceso. Esto supone que el controlador de acceso conoce también la contraseña asociada con el ID de abono. El controlador de acceso autenticará al usuario verificando que la cadena de preguntas está criptada correctamente.

El procedimiento de registro de ejemplo con autenticación de controlador de acceso es el siguiente:

1) Si el punto extremo utiliza **GRQ** para descubrir un controlador de acceso, uno de los alias del mensaje sería el ID de suscripción (como un **H323ID**). La **capacidad de autenticación (authenticationcapability)** contendría un **Mecanismo de autenticación (AuthenticationMechanism)** de **criptación simétrica de contraseña (pwdSymEnc)** y los **OID de algoritmo (algorithmOIDs)** se fijarían para indicar el conjunto completo de algoritmos de criptación soportados por el punto extremo. (Por ejemplo, uno de estos sería DES de 56 bits en modo EBC.)

2) El controlador de acceso respondería con **GCF** (suponiendo que reconoce el alias) que transporta un elemento **testigos (tokens)** que contiene un **testigo claro (ClearToken)**. Este **Testigo claro** contendría una **pregunta** y un elemento **de indicación de tiempo**. La **pregunta** contendría 16 octetos. (Para impedir ataques de reproducción, el **Testigo claro** contendría un **sello de hora**.) El **modo de autenticación** se pondría a **criptación simétrica de contraseña** y el **OID de algoritmo** se fijaría para indicar el algoritmo de criptación requerido por el controlador de acceso (por ejemplo, DES de 56 bits en modo EBC).

   Si el controlador de acceso no soporta algunos de los **algorithmOIDs** indicado en el **GRQ**, respondería con un mensaje **GRJ** que contiene un **Motivo de rechazo de controlador de acceso (GatekeeperRejectReason)** de **recurso no disponible (resourceUnavailable)**.

3) La aplicación de punto extremo trataría de registrarse con (uno de) los controladores de acceso que respondieron con un **GCF** enviando un **RRQ** que contiene una **contraseña de EP cifrada (cryptoEPPwdEncr)** en los **testigos cifrados**. La **contraseña de EP criptada** tendría el **OID del algoritmo** de criptación acordado en el intercambio **GRQ/GCF**, y la pregunta criptada.

   La clave de criptación se construye a partir de la contraseña del usuario utilizando el procedimiento descrito en 10.3.2. La "cadena" de octetos resultante se utiliza como la clave DES para criptar la **pregunta**.

4) Cuando el controlador de acceso recibe la pregunta criptada en el **RRQ**, la comparará con una pregunta criptada generada idénticamente para autenticar al usuario que registra. Si las dos cadenas criptadas no concuerdan, el controlador de acceso responderá con un **RRJ** con el **Motivo de rechazo de registro (RegistrationRejectReason)** puesto a **denegación de seguridad**. Si concuerdan, el guardián de puerta envía un **RCF** al punto extremo.

5) Si el controlador de acceso recibe un **RRQ** que no contiene un elemento Testigos criptados aceptable, debe responder con un **RRJ** con un **Motivo de rechazo de controlador de acceso de descubrimiento requerido (discoveryRequired)**. El punto extremo, al recibir este **RRJ** puede efectuar un descubrimiento que le permitirá al controlador de acceso/punto extremo intercambiar una nueva pregunta. Obsérvese que el mensaje **GRQ** puede ser unidifundido al controlador de acceso.

I.4.5 IPSEC

En general IPSEC **[13/IPSEC**] se puede utilizar para proporcionar autenticación y, facultativamente, confidencialidad (es decir, criptación) en la capa IP transparente a cualquier protocolo (aplicación) que funcione por encima de ella. El protocolo de aplicación no tiene que ser actualizado para permitir esto; sólo la política de seguridad en cada extremo.

Por ejemplo, para utilizar al máximo IPSEC para una llamada simple punto a punto, se puede aplicar lo que sigue:

1) El punto extremo llamante y su controlador de acceso fijarían la política para requerir la utilización de IPSEC (autenticación y, facultativamente confidencialidad) en el protocolo RAS. De este modo, antes de que el primer mensaje RAS sea enviado desde el punto extremo al controlador de acceso, el protocolo ISAKMP/Oakley en el punto extremo negociará los servicios de seguridad que se han de utilizar en paquetes a y desde el puerto bien conocido del canal RAS. Una vez completada la negociación, el canal RAS funcionará exactamente como si no fuese seguro. Al utilizar este canal de seguridad, el controlador de acceso informará al punto extremo la dirección y el número de puerto del canal de señalización de la llamada en el punto extremo llamado.

2) Después de obtener la dirección y el número de puerto del canal de señalización de llamada, el punto extremo llamante actualizaría dinámicamente su política de seguridad para requerir la seguridad IPSEC deseada en esa dirección y par de protocolo/puerto. En ese momento, cuando el punto extremo llamante intenta ponerse en contacto con esta dirección/puerto, los paquetes se pondrían en cola mientras se realiza una negociación ISAKMP/Oakley entre los puntos extremos. Al completar esta negociación, existirá una asociación de seguridad (SA, *security association*) IPSEC para la dirección/puerto y se puede pasar a la señalización Q.931.

3) En el intercambio de los mensajes ESTABLECIMIENTO y CONEXIÓN Q.931, los puntos extremos pueden negociar la utilización de IPSEC para el canal H.245. Esto permitiría a los puntos extremos actualizar de nuevo dinámicamente sus bases de datos de política IPSEC para forzar el uso de IPSEC en esa conexión.

4) Al igual que en el caso del canal de señalización de llamada, se producirá una negociación ISAKMP/Oakley transparente antes de que se transmitan paquetes H.245. La autenticación realizada por esta negociación ISAKMP/Oakley será el intento inicial de la autenticación de usuario a usuario, y establecerá entre los dos usuarios un canal (probablemente) seguro por el cual negociar las características del canal de audio. Si después de Q y A de persona a persona, uno de los dos usuarios no está satisfecho con la autenticación, se pueden elegir diferentes certificados y repetir el intercambio ISAKMP/Oakley.

5) Después de cada autenticación ISAKMP/Oakley H.245, se intercambia nuevo material de claves para el canal de audio RTP. Este material de claves es distribuido por el terminal director por el canal H.245 seguro. Como el protocolo H.245 está definido para que el director distribuya el material de clave de los medios por el canal H.245 (para la comunicación multipunto), no se recomienda utilizar IPSEC para el canal RTP.

Un canal H.245 criptado es un posible problema para apoderados o cortafuegos NAT, porque los números de puerto asignados dinámicamente son transportados en el protocolo H.245. Estos cortafuegos tendrían que descifrar, modificar y cifrar de nuevo el protocolo para funcionar correctamente. Por este motivo, se introdujo el canal lógico de "seguridad" en UIT-T H.245. Si este canal se utiliza, el canal H.245 puede permanecer inseguro; la autenticación y la generación de claves se haría con el canal lógico de "seguridad". La señalización de canal lógico permitiría que este canal estuviese protegido con IPSEC, y la clave secreta utilizada en el canal lógico de "seguridad" se emplearía para proteger el campo **sincronización criptada** distribuido por el terminal director por el canal H.245.

I.4.6 Soporte de servicios fuera del terminal

Los servidores fuera del terminal son una función suplementaria importante en un entorno multimedios basado en H.323 global. Por ejemplo, los BES proporcionan servicios para la autenticación del usuario y la autorización del servicio, así como la facturación, tarificación, contabilidad y otros servicios. En un modelo simple el controlador de acceso puede proporcionar tales servicios. En una arquitectura descompuesta el controlador de acceso no siempre puede proveer tales servicios; bien porque no tiene acceso a las bases de datos BES o bien porque estas pueden ser parte de un dominio administrativo diferente. Del mismo modo, el terminal o usuario no conoce normalmente sus BES.

En la figura I.7 se muestra un escenario con un terminal multimedios (por ejemplo, un SASET), un controlador de acceso y un BES enlazado. No cae en el ámbito de H.323 el modo en los BES comunican exactamente con el GK. Se pueden aplicar varios métodos y protocolos: RADIUS (véase RFC 2138) se considera uno de los más importantes, y es desplegado ampliamente por los proveedores del servicio.

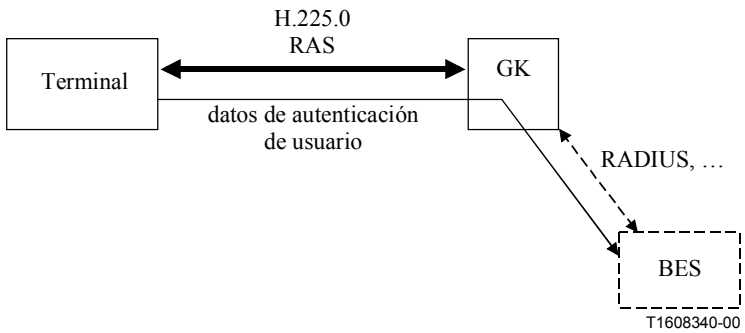


Figura I.7/H.235 – Escenario con servidor fuera del terminal

Un GK que ofrece el soporte BES debería soportar al menos los dos modos siguientes:

1) **modo por defecto**, en el cual el terminal no conoce el BES y necesita una relación de confianza con el GK. El terminal envía al GK los datos de autenticación de usuario en forma criptada (**cryptoEncryptedToken**), y el GK decripta estos datos, extrae la información de autenticación de usuario y la envía hacia el BES. La criptación basada en contraseñas del **ClearToken** se realiza aplicando un secreto distinto del compartido entre el terminal y el GK al **CryptoToken**. La clave de criptación puede obtenerse a partir de la contraseña con la cual el terminal se registra de modo seguro en el GK.

   CryptoToken cursa **cryptoEncryptedToken** en el cual **tokenOID** se pone a "M", indicando el modo por defecto de BES; y el **token** contiene:

   - **algorithmOID**, que indica el algoritmo de criptación: "Y" (DES56-CBC), "Z" (3DES-ocbc); véase D.11,
   - **params**, no utilizado,
   - **encryptedData,** fijado a la representación de octetos del **ClearToken** criptado.

   El **ClearToken** contiene como **password** los datos de autenticación de usuario. La información **ClearToken** protegida puede ser contraseña/PIN, identificación de usuario, número de tarjeta de llamadas de previo pago y número de tarjeta de crédito. El campo **timestamp** se fija al tiempo real del terminal; **random** contiene un número secuencial monotónicamente creciente, **sendersID** se fija al valor del ID de terminal y **generalID** al valor del identificador de GK. El valor inicial (IV) del algoritmo de criptación deberá mantenerse constante; este valor puede formar parte del secreto del abono del terminal.

   NOTA – El **ClearToken** no se transmite.

2) ***modo RADIUS***, donde el BES y el usuario terminal comparten un secreto común y el GK no ha de ser apoderado para la autenticación RADIUS de BES. El GK simplemente reenvía una consulta (challenge) RADIUS recibida del BES dentro de *Access-Challenge* hacia el terminal y envía la respuesta del usuario como una respuesta RADIUS dentro de *Access-Request* en la dirección inversa. El terminal y el GK negocian la capacidad consulta/respuesta **radius** en **AuthenticationBES** dentro del **AuthenticationMechanism** durante el descubrimiento del controlador de acceso.

   Tras la recepción de un mensaje *Access-Challenge* RADIUS que transporta una consulta, el GK coloca la consulta de 16 octetos en el campo **challenge** del **ClearToken** cuando se pregunta al terminal con un **GCF** o cualquier otro mensaje RAS. El **tokenOID** 'K' en el **ClearToken** indica una consulta RADIUS.

   El terminal puede entonces presentar la consulta al usuario y esperar la respuesta. El terminal deberá contestar con un mensaje RAS en el cual se ha introducido la respuesta en el campo **challenge** del **ClearToken.** El **tokenOID** 'L' en el **ClearToken** indica una respuesta RADIUS.

En el cuadro I.1 se da una lista de los OIDs referidos.

Cuadro I.1/H.235 – Identificadores de objeto utilizados en I.4.6

Referencia del identificador de objeto	Valor del identificador de objeto	Descripción
"K"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 1}	indica una consulta RADIUS en el ClearToken
"L"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 2}	indica una respuesta RADIUS (cursada en el campo consulta) en el ClearToken
"M"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 3}	indica el modo por defecto BES con una contraseña protegida en el ClearToken

APÉNDICE II

Detalles de implementaciones del protocolo H.324

Queda en estudio.

APÉNDICE III

Otros detalles de implementaciones de la serie H

Queda en estudio.

APÉNDICE IV

Bibliografía

[Daemon]

– DAEMON (J.): Cipher and Hash function design, Ph.D. Thesis, Katholieke Universiteit Leuven, marzo de 1995.

[IPSEC]

– MAUGHAN (D.), SCHERTLER (M.), SCHNEIDER (M.), TURNER (J.): Internet Security Association and Key Management Protocol (ISAKMP), draft-ietf-ipsec-isakmp-08.text, *Internet Engineering Task Force*, 1997.

[ISO | IEC 14888-3]

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

[PKCS]

– PKCS #1 v2.0: RSA Cryptography Standard; RSA Laboratories; 1 de octubre de 1998; http://www.rsa.com/rsalabs/pubs/PKCS/index.html.

– PKCS #7: Cryptographic Message Syntax Standard, An RSA Laboratories Technical Note, Version 1.5, revisada el 1 de noviembre de 1993; http://www.rsa.com/rsalabs/pubs/PKCS/index.html

[RTP]

– SCHULZRINNE (H.), CASNER (S.), FREDERICK (R.), JACOBSON (V.): RTP: A transport Protocol for Real-Time Applications, RFC 1889, *Internet Engineering Task Force*, 1996.

[Schneier]

– SCHNEIER (B.): Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition, John Wiley & Sons, Inc., 1995.

[TLS]

– DIEKS (T.), ALLEN (C.): The TLS Protocol Version 1.0, RFC 2246, *Internet Engineering Task Force*, 1999.

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

Ginebra, 2001