UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T H.235

(08/2003)

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 UIT-T H.323 y H.245)

Recomendación UIT-T H.235

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
PROCEDIMIENTOS DE MOVILIDAD Y DE COLABORACIÓN	
Visión de conjunto de la movilidad y de la colaboración, definiciones, protocolos y procedimientos	H.500–H.509
Movilidad para los sistemas y servicios multimedios de la serie H	H.510–H.519
Aplicaciones y servicios de colaboración en móviles multimedios	H.520–H.529
Seguridad para los sistemas y servicios móviles multimedios	H.530–H.539
Seguridad para las aplicaciones y los servicios de colaboración en móviles multimedios	H.540–H.549
Procedimientos de interfuncionamiento de la movilidad	H.550–H.559
Procedimientos de interfuncionamiento de colaboración en móviles multimedios	H.560–H.569
SERVICIOS DE BANDA ANCHA Y DE TRÍADA MULTIMEDIOS	
Servicios multimedios de banda ancha sobre VDSL	H.610–H.619

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 UIT-T 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 la Rec. 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 la Rec. 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.

La versión 2 de la Rec. 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 la Rec. UIT-T H.323.

La versión 3 de la Rec. UIT-T H.235 sustituye a la versión 2 con las siguientes mejoras: un procedimiento para señales DTMF criptadas, unos identificadores de objeto para el algoritmo de criptación AES a efectos de criptación de cabida útil de medios, el modo de criptación para el cifrado de trenes OFB mejorado (EOFB, *enhanced OFB*) para la criptación de trenes de medios, una opción de sólo autenticación para el paso sin problemas a través de un NAT/cortafuegos, presentada en el anexo D, un procedimiento de distribución de claves en el canal RAS, algunos procedimientos para el transporte más seguro de claves de sesión y una distribución y actualización de claves más robustas, unos procedimientos para proporcionar seguridad a trenes de cabida útil múltiple, un mejor soporte de seguridad para las llamadas con encaminamiento directo en un nuevo anexo I, unos medios de señalización que permitan informes de error más flexibles, algunas aclaraciones y mejoras de la eficacia con el fin de lograr seguridad en el arranque rápido y para la señalización Diffie-Hellman, junto con parámetros Diffie-Hellman más largos y ciertos cambios provenientes de la guía del implementador de la Rec. UIT-T H.323.

Orígenes

La Recomendación UIT-T H.235 fue aprobada el 6 de agosto de 2003 por la Comisión de Estudio 16 (2001-2004) del UIT-T por el procedimiento de la Rec. A.8.

Palabras clave

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

Historia

Versión	Aprobación
H.235v1	1998-02-06
H.235v2	2000-11-17
H.235v3	2003-08-06

PREFACIO

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

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

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

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

NOTA

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

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

PROPIEDAD INTELECTUAL

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

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

ÍNDICE

Página

Recomendación UIT-T H.235

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

1 Alcance

La finalidad primaria de esta Recomendación es proporcionar autenticación, privacidad e integridad dentro del marco de los protocolos vigentes de la serie H. El texto actual de esta Recomendación (2003) proporciona detalles sobre la implementación con la Rec. UIT-T H.323. Se prevé que este marco funcione junto con otros protocolos de la serie H que utilizan como protocolo de control el protocolo H.245.

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) Esta 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 seguros 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) Esta 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 la compatibilidad. 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 intromisión en la red o algún otro método de desviar los trenes de medios.

La Rec. UIT-T H.323 proporciona los medios para conducir una conferencia de audio, vídeo y datos entre dos o más partes, pero no el mecanismo para que cada participante pueda autenticar la identidad de los otros participantes, ni los medios para salvaguardar la privacidad de comunicaciones (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 la Rec. 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 lo que pueden decodificar los receptores, 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. En esta Recomendación, la referencia a un documento, en tanto que autónomo, no le otorga el rango de una Recomendación.

- Recomendación UIT-T H.225.0 (2003), *Protocolos de señalización de llamadas y paquetización de trenes de medios para sistemas de comunicaciones multimedios por paquetes*.
- Recomendación UIT-T H.235 (1998), *Seguridad y criptado para terminales multimedios de la serie H (basados en las Recomendaciones H.323 y H.245)*.
- Recomendación UIT-T H.235 (2000), *Seguridad y criptado para terminales multimedios de la serie H (basados en las Recomendaciones H.323 y H.245)*.
- Recomendación UIT-T H.530 (2002), *Procedimientos de seguridad simétricos para movilidad de sistemas H.323 según la Recomendación H.510*.
- Recomendación UIT-T H.530 corrigendum 1 (2003), *Procedimientos de seguridad simétricos para movilidad de sistemas H.323 según la Recomendación H.510*.
- Recomendación UIT-T H.245 (2003), *Protocolo de control para comunicación multimedios*.
- Recomendación UIT-T H.323 (2003), *Sistemas de comunicación multimedios basados en paquetes*.
- Recomendación 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*.
- Recomendación 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*.
- Recomendación 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*.

- Recomendación UIT-T X.803 (1994) | ISO/CEI 10745:1995, *Tecnología de la información – Interconexión de sistemas abiertos – Modelo de seguridad de capas superiores*.
- Recomendación 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*.
- Recomendación 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 – 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 10116:1997, *Information technology – Security techniques – Modes of operation for an n-bit block cipher*.
- ISO/CEI 15946-1:2002, *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 1: General*.
- ISO/CEI 15946-2:2002, *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 1.1*.
- IETF RFC 1321 (1992), *The MD5 Message-Digest Algorithm*.
- IETF RFC 2104 (1997), *HMAC: Keyed-Hashing for Message Authentication*.
- IETF RFC 2865 (2000), *Remote Authentication Dial In User Service (RADIUS)*.
- IETF RFC 2198 (1997), *RTP Payload for Redundant Audio Data*.
- 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 Cryptography Specifications Version 2.0*.
- IETF RFC 2833 (2000), *RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals*.
- IETF RFC 3280 (2002), *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*.

3 Términos y definiciones

En 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 las Recomendaciones UIT-T X.800 | ISO 7498-2 y X.803, X.810 y 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 (Rec. UIT-T X.800).

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

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

3.4 ataque: Actividades realizadas para obviar 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 obvian 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 (Rec. UIT-T 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, archivado y aplicación de claves de acuerdo con una política de seguridad (Rec. UIT-T 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 el cifrado.

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 la Rec. UIT-T H.235 y que son útiles para proporcionar seguridad a 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 (Rec. UIT-T X.810).

3.20 amenaza: Posible violación de la seguridad (Rec. UIT-T X.800).

4 Símbolos y abreviaturas

En esta Recomendación se utilizan las siguientes siglas.

3DES	DES triple (*triple DES*)
AES	Algoritmo de criptación avanzado (*advanced encryption algorithm*)
ASN.1	Notación de sintaxis abstracta N.º 1 (*abstract syntax notation No. 1*)
BES	Servidor fuera del terminal (*back-end server*)
CA	Autoridad de certificación (*certificate authority*)
CBC	Concatenación de bloques cifrados (*cipher block chaining*)
CFB	Modo de retroalimentación cifrado (*cipher feedback mode*)
CRL	Lista de revocación de certificados (*certificate revocation list*)
DES	Norma de criptación de datos (*data encryption standard*)
DH	Diffie-Hellman
DNS	Sistema de nombres de dominio (*domain name system*)
DSS	Norma sobre firmas digitales (*digital signature standard*)
DTMF	Multifrecuencia bitono (*dual tone multi-frequency*)
ECB	Libro de código electrónico (*electronic code book*)
ECC y EC	Criptosistema de curva elíptica (*elliptic curve cryptosystem*) (véase la sección 8.7 *ATM Forum Security Specification Versión 1.1*). Un criptosistema de claves públicas (*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 (*elliptic curve key agreement scheme – Diffie-Hellman*) – El esquema de convenio de claves Diffie-Hellman que utiliza criptografía de curva elíptica. *The Diffie-Hellman key agreement scheme using elliptic curve cryptography*)

EOFB	Modo OFB mejorado (*enhanced OFB mode*)
EP	Punto extremo (*endpoint*)
GK	Controlador de acceso (*gatekeeper*)
GW	Pasarela (*gateway*)
ICV	Valor de comprobación de integridad (*integrity check value*)
ID	Identificador (*identifier*)
IPSEC	Seguridad del protocolo Internet (*Internet protocol security*)
ISAKMP	Protocolo de gestión de clave con asociación de seguridad en Internet (*Internet security association key management protocol)*
IV	Vector de inicialización (*initialization vector)*
LDAP	Protocolo ligero de acceso al directorio (*lightweight directory access protocol*)
MAC	Código de autenticación de mensaje (*message authentication code*)
MCU	Unidad de control multipunto (*multipoint control unit*)
MD5	Message Digest No. 5
MPS	Tren de cabida útil múltiple (*multiple payload stream*)
NAT	Traducción de dirección de red (*network address translation*)
OCSP	Protocolo en línea del estado del certificado (*online certificate status protocol*)
OFB	Modo realimentación de salida (*output feedback mode*)
OID	Identificador de objeto (*object identifier*)
PDU	Unidad de datos de protocolo (*protocol data unit*)
PKCS	Criptosistema de claves públicas (*public-key crypto system*)
PKI	Infraestructura de claves públicas (*public key infrastructure*)
PRF	Función seudoaleatoria (*pseudo-random function*)
QOS	Calidad de servicio (*quality of service*)
RSA	Rivest, Shamir y Adleman (algoritmo de clave pública)
RTCP	Protocolo de control de transporte en tiempo real (*real-time transport control protocol*)
RTP	Protocolo de transporte en tiempo real (*real-time transport protocol*)
SDU	Unidad de datos de servicio (*service data unit*)
SHA1	Algoritmo de generación numérica seguro N.º 1 (*secure hash algorithm No. 1*)
SRTP	Protocolo de transporte en tiempo real seguro (*secure real-time transport protocol*)
SSL	Capa de zócalo segura (*secure socket layer*)
TLS	Seguridad de nivel de transporte (*transport level security*)
TSAP	Punto de acceso al servicio de transporte (*transport service access point*)
X || Y	Concatenación de X e I
XOR, ⊕	O exclusivo (*exclusive OR*)

5 Convenios

En la presente Recomendación se utilizan los siguientes convenios:

– El tiempo futuro de mandato 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 refieren a 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 Rec. UIT-T 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 unidas y en negritas (**maximumDelayJitter)**; 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 aparecen en mayúsculas **(Call Proceeding)**.

En esta Recomendación se definen diversos identificadores de objeto (OID) para la señalización de capacidades de seguridad, procedimientos o algoritmos de seguridad. Estos identificadores están relacionados con un árbol jerárquico de valores atribuidos que puede provenir de una fuente externa o ser parte del árbol de OID mantenido por el UIT-T. En particular, aquellos OID relativos a la Rec. UIT-T H.235 se presentan en el texto de la siguiente manera:

"OID" = {itu-t (0) recommendation (0) h (8) 235 version (0) **V N**}, donde **V** representa simbólicamente una única cifra decimal que indica la versión correspondiente de la Rec. UIT-T H.235, por ejemplo, 1, 2 ó 3, **N** representa simbólicamente una cifra decimal que identifica unívocamente el ejemplar del OID y, por tanto, el procedimiento, algoritmo o capacidad de seguridad.

Es decir, el OID codificado ASN.1 consta de una secuencia de números. Por comodidad, se utiliza una notación de abreviaturas nemotécnicas de texto para cada OID, por ejemplo "OID". Se suministra una correspondencia entre cada cadena OID y una secuencia de números ASN.1. Las implementaciones conformes a la Rec. UIT-T H.235 utilizarán únicamente los números codificados ASN.1.

Cuando se utiliza la criptación de medios junto con el relleno de cabida útil, a veces se dice que: "el valor del relleno debería establerse utilizando el convenio normal del algoritmo cifrado"; véanse por ejemplo las cláusulas 8.6.1 y B.2.4, y la figura I.5. Con esto se pretende decir que algunos algoritmos cifrados (por ejemplo, DES) permite saber más acerca de cómo el remitente puede escoger el valor del (los) byte(s) de relleno. Esto ocurre, por ejemplo, en los valores que se rellenan al azar, los valores estáticos u otros patrones generados. Aunque el método escogido no afecta la compatibilidad, la calidad de seguridad puede muy bien depender de él. Esto se considera como un aspecto de implementación y no se trata más en esta Recomendación.

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 (Rec. UIT-T H.225.0).

2) Los usuarios pueden ser autenticados durante la conexión de llamada inicial, en el proceso de proporcionar seguridad el canal H.245 y/o mediante el intercambio de 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 H.245 **OpenLogicalChannel** (**Apertura canal lógico**) u **OpenLogicalChannelAck** (**Acuse apertura canal lógico**).

5) El recifrado se puede realizar mediante las instrucciones H.245: **EncryptionUpdateCommand** (**Instrucción actualización criptación**), **EncryptionUpdateRequest** (**Petición actualización criptación**), **EncryptionUpdate** (**Actualización criptación**) y **EncryptionUpdateAck** (**Acuse 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 presentación/admisión. 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 pueden ser soportadas 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 proporcionar seguridad al 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) debería estar seguro 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 características diferentes de las 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 **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 criptado o cualesquiera canales lógicos del tipo **encryptedData** (**Datos criptados**) será considerada 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 debería 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.

6.8 Seguridad en entorno de movilidad

Es posible utilizar los sistemas basados en la H.323 en un entorno de movilidad conforme a la Rec. UIT-T H.510. En la Rec. UIT-T H.530 se describen los procedimientos y protocolos de seguridad para dichos sistemas, y se presentan protocolos y procedimientos de esta Recomendación.

6.9 Perfiles de seguridad

Esta Recomendación tiene varios anexos (por ejemplo, anexos D, E y F) y, cada uno de ellos mantiene perfiles de seguridad de H.235. En un perfil de seguridad se especifica la utilización particular de H.235 o un subconjunto de funcionalidades de esa Recomendación para entornos bien definidos, con un alcance de aplicabilidad preciso.

Dependiendo del entorno y de la aplicación, se pueden implementar los perfiles de seguridad bien sea de una manera selectiva o todos al tiempo. Con frecuencia, en los sistemas en que se ha habilitado la H.235 se indica dentro de los identificadores de objeto, como parte de los mensajes de señalización, qué perfiles de seguridad utilizan. En estos sistemas se debería escoger el perfil de seguridad conforme a sus propias necesidades.

Por otra parte, los puntos extremos pueden también ofrecer inicialmente múltiples perfiles de seguridad simultáneamente, en mensajes RRQ/GRQ, y después esperar a que el controlador de acceso escoja el más adecuado a través de una respuesta a ellos en un mensaje RCF/GCF. Las transacciones LRQ/LCF entre controladores de acceso también pueden transportar varios perfiles de seguridad. Al calcular firmas digitales o números generados para proporcionar integridad de mensaje, en primer lugar se deberían calcular los números generadores y signaturas digitales que no proporcionen dicha integridad en el subconjunto de campos y ponerlos en el mensaje, poner a cero en la memoria intermedia de mensaje todos aquellos que sí lo hagan, y sólo entonces se deberían calcular las firmas digitales y los números generadores utilizando esta memoria, para después ponerlos en el mensaje.

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 debería transferir información sobre cualquier discordancia de seguridad y el terminal llamante tendrá que determinar el problema por otros medios. Cuando el terminal llamante reciba un mensaje 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 debería considerarse un error de protocolo y terminarse la conexión.

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 de control 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, en cuyo caso 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 proteja 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 5.2/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 puede 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 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**.

8.6 Seguridad de conexión rápida

Es posible que los puntos extremos utilicen el procedimiento de conexión rápida (véanse 8.1.7 y 8.1.7.1/H.323) utilizando el elemento de arranque rápido para intercambiar con seguridad material de claves (clave maestra y claves de criptación de sesión). Los procedimientos presentados en 8.6.1. describen el arranque rápido "básico" en que no se utilizan los diversos algoritmos de criptación ofrecidos, mientras que en 8.6.1.1 se describe el caso particular de un arranque rápido con diversos algoritmos de criptación ofrecidos, lo que facilita una codificación más compacta de mensaje.

8.6.1 Seguridad de arranque rápido unidireccional

Este procedimiento describe cómo establecer un canal lógico de seguridad unidireccional (semiduplex) desde el emisor hasta el receptor de la llamada.

Procedimientos del llamante (emisor)

El llamante (fuente del **Setup)** presenta tanto su testigo DH como las estructuras FastStart soportadas. El testigo DH se transportará dentro de un ClearToken incorporado como parte de un CryptoToken, o como un ClearToken separado, véase también 8.8. Durante la secuencia **Setup**-to-**Connect**, se efectuará un intercambio Diffie-Hellman (DH), de manera que se establezca en ambos puntos extremos un secreto compartido. El campo **ClearToken** de los campos **CryptoToken** incluirá una **dhkey**, utilizada para pasar los parámetros conforme a esta Recomendación. **halfkey** contiene la clave pública aleatoria de una parte, **modsize** el número primo DH y **generator** el grupo DH. En el cuadro D.4 se indican los parámetros DH que se han de utilizar. Para mayor información véase [RFC 2412, apéndice E2].

NOTA 1 – Puesto que los mensajes H.225.0 son autentificados (como se describió en el procedimiento I), el intercambio DH es autenticado.

En cualquier sentido, con un mensaje de señalización de llamada H.225.0 que transporte media clave Diffie-Hellman, si se dispone de información de identificación, el llamante o el llamado "cuando estén registrados" incluirán también un **ClearToken** extremo a extremo separado, en el que se haya puesto **sendersID** al identificador de punto extremo del remitente y **tokenOID** a "E". Toda entidad de señalización H.323 intermedia reenviará este testigo extremo a extremo sin modificación.

Las estructuras FastStart tienen los canales lógicos abiertos ofrecidos con las capacidades de seguridad propuestas. Se debería ofrecer tanto el canal H235Cap como nonH235Cap. Durante el intercambio de capacidades H.245, los puntos extremos presentan entradas **H235SecurityCapability** para los códecs que soportan. Cada códec se asocia con una capacidad de seguridad H.235 independiente. Conforme al anexo D, estas capacidades deberían indicar el soporte de AES-CBC de 128 bits (OID – "Z3"), RC2 compatible CBC de 56 bits (OID – "X") y DES-CBC de 56 bits (OID – "Y"), y podrían indicar el soporte de DES triple-CBC de 168 bits (OID – "Z"), o de DES triple EOFB de 168 bits (OID – "Z1"), RC2 compatible con EOFB (OID – "X1"), DES-EOFB (OID – "Y1") o de AES-EOFB (OID – "Z2"). Véase también el cuadro D.6.

El **OpenLogicalChannel** transporta tanto **forwardLogicalChannelParameters** como **reverseLogicalChannelParameters** con **dataType**, lo que proporciona **encryptionAuthenticationAndIntegrity**, a **h235Media** y al menos un **MediaEncryptionAlgorithm** en la **encryptionCapability**.

A efectos de la relación de seguridad, el destinatario será en principio el terminal director, véase también 8.4.

El llamante debería poner **mediaWaitForConnect** a verdadero, con el fin de afirmar que se dispone de material de clave de sesión y que se pueden descifrar los medios criptados recibidos. Siempre que se desee un establecimiento de canal "temprano", por ejemplo cuando el llamado transmita simultáneamente medios criptados o no criptados con respuestas a mensajes y material de clave de criptación, el llamante debería estar preparado para no poder descifrar los contenidos a menos que disponga de material clave.

NOTA 2 – En este caso, si el llamado envía medios criptados al llamante (algo que en teoría puede hacer, puesto que tiene su dirección RTP/RTCP), éste no podrá descifrarlo sin la ayuda del secreto compartido proporcionado en el mensaje de Conexión (Aviso o Llamada en curso).

Procedimientos del llamado

Durante el FastStart, el llamado presenta su testigo DH (véase también 8.8) y las estructuras FastStart aceptadas. Cuando se utiliza el procedimiento Diffie-Hellman, se recomienda que el llamado retorne su testigo DH como parte del mensaje respuesta tan pronto como pueda. Es decir, en el mensaje respuesta que viene inmediatamente después del SETUP. De esta manera, el llamante podrá calcular la clave maestra a partir del secreto compartido DH y estará preparado para recibir la clave de sesión y los medios criptados.

NOTA 3 – De no haber algoritmo de criptación disponible en ambos lados, se puede dejar el tren de medios sin criptar o se puede abandonar la conexión dependiendo de la política de seguridad.

Cada entidad tomará los bits menos significativos adecuados a partir del secreto Diffie-Hellman compartido común para la clave de criptación clave (clave maestra), es decir una cantidad de bits menos significativos del secreto Diffie-Hellman correspondiente a: 56 para OID "X", OID "X1", OID "Y1" u OID "Y", 168 para OID "Z", OID "Z1" u OID "Z2", y 128 para OID "Z3" u OID "Z2". Véase también el cuadro D.6.

Se emiten respuestas **OpenLogicalChannel(Ack)** con la clave de sesión creada (maestro) incluida en el campo **encryptionSync**. Este campo incluye la clave de sesión para el canal lógico dirigido desde el llamante hasta el llamado. El transporte de clave se hará conforme al procedimiento descrito en B.2.4, ya sea utilizando **KeySyncMaterial** o **V3KeySyncMaterial** (véase B.2.4.1). La clave de sesión se criptará utilizando el secreto compartido DH, como se describe a continuación.

NOTA 4 – No existe ningún método preestablecido para generar las claves de sesión, utilizadas en la criptación de medios. La generación de estos valores depende de la implementación que, a su vez, se ve afectada por los recursos locales, las políticas, y el algoritmo de criptación que se vaya a utilizar. Conviene tener cuidado de no generar claves débiles.

Utilizando el procedimiento de B.2.4, se transportará la sesión criptada en el **H.235Key/sharedSecret** dentro del campo **encryptionSync**. La clave de sesión se transportará en el campo **keyMaterial** del **KeySyncMaterial**, y cuando no sea un múltiplo del tamaño del bloque se aplicará un relleno para completarla antes de la criptación. El valor del relleno se debería estimar mediante el convenio normal del algoritmo de cifrado. El **KeySyncMaterial** (de relleno) se codificará conforme a:

– 56 bits del secreto compartido, empezando por los menos significativos del secreto Diffie-Hellman para OID "X", OID "X1" , OID "Y1" u OID "Y".

– Todos los bits del secreto compartido para OID "Z2", OID "Z" u OID "Z1", comenzando con los bits menos significativos del secreto DH.

Por otra parte, y siempre que se pueda, se debería utilizar el transporte de clave mejorado conforme a B.2.4.1, debido al resultado del procedimiento indicativo de la versión 3 (véase B.2.3).

Cuando se deba establecer un canal de medios seguro dúplex utilizando un arranque rápido, de entre dos canales unidireccionales, el llamado abrirá un segundo canal lógico hacia el llamante. Este canal se señalará en un elemento fastStart separado. Utilizando el secreto compartido DH disponible como clave maestra, el recipiente incluirá otra clave de sesión para este canal lógico en el **encryptionSync**.

8.6.1.1 Utilización de algoritmos de criptación múltiple en la conexión rápida

La negociación de la criptación de medios como parte de los procedimientos de conexión rápida conduce a un incremento ineficaz de la cantidad de elementos **OpenLogicalChannel** en el elemento **fastConnect** de un mensaje SETUP. Esto ocurre puesto que se necesita un **OLC** independiente para cada combinación de códecs (**dataType**) y algoritmo de criptación (incluido "none").

Se especifica el algoritmo de criptación que se ha de aplicar a un tren de medios mediante la inclusión del **dataType.h235Media.encryptionAuthenticationAndIntegrity.encryptionCapability dataType** en el **OLC**. En H.235v2 se recomienda incluir solamente un único **MediaEncryptionAlgorithm** en la **encryptionCapability**, aunque este último elemento se defina como una secuencia de los elementos anteriores. Así pues, se puede incluir una secuencia ordenada por preferencias de capacidades de criptación en cada **OLC** ofrecido. El receptor de **OLC** escogerá entonces un algoritmo único de entre aquellos que se ofrecen, y retornará el **OLC** con únicamente el algoritmo escogido (junto con las direcciones de transporte e información clave de criptación apropiadas).

Para garantizar una eficacia máxima, el identificador de objeto "NULL-ENCR" (véase el cuadro 1) representa el algoritmo de criptación "null", o lo que es lo mismo indica que no tiene lugar ninguna operación de criptación. De esta manera, se necesita solamente un **OLC** por códec ofrecido y para cada sentido.

Procedimientos para el llamante (véase 8.1.7.1/H.323)

Si se especifica en un elemento **dataType** ofrecido la criptación a través de la selección de **h235Media**, es posible que el elemento **encryptionAuthenticationAndIntegrity** allí presente incluya un elemento **encryptionCapability** que contenga diversos algoritmos de criptación (incluido el algoritmo NULL). Esta construcción estará destinada a ofrecer la posibilidad de escoger entre los diversos algoritmos especificados para la criptación de las capacidades de medios correspondientes.

Procedimientos para el llamado (véase 8.1.7.1/H.323)

Si se ofrecen diversos algoritmos de criptación para un canal, el punto extremo llamado deberá seleccionar uno y modificar el **OpenLogicalChannel** a fin de suprimir los otros.

Cuadro 1/H.235 – Identificador de objeto para la criptación NULL

Referencia de identificador de objeto	Valor de identificador de objeto	Descripción
"NULL-ENCR"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 26}	Indica el "algoritmo de criptación NULL"

8.6.2 Seguridad de canales bidireccionales durante el arranque rápido

La seguridad de los canales de datos T.120 bidireccionales queda en estudio.

8.7 DTMF H.245 criptadas

Los puntos extremos pueden enviar señales DTMF criptadas para lograr confidencialidad. Usando la clave de criptación de sesión, estos puntos pueden criptar las señales DTMF en **UserInputIndication** de la siguiente manera:

- Cadena básica criptada: **encryptedAlphanumeric**.
- Cadena iA5 criptada: **encryptedSignalType** en **signal**.
- Cadena general criptada: **encryptedAlphanumeric** en **extendedAlphanumeric**.

NOTA 1 – No se criptan los parámetros adicionales para el RTP en la cadena iA5, con indicaciones de tiempo y números de canal lógico o la actualización de señal con la duración de tono, al no considerarlos adecuados para transportar información confidencial.

La capacidad negociada **secureDTMF** está relacionada con una cadena iA5 criptada.

Como se explica en la cláusula D.7, la gestión de clave debería aplicarse para obtener una clave de criptación de sesión. Dicha clave se utilizará para criptar las señales DTMF H.245.

NOTA 2 – Esto no significa necesariamente que se deba aplicar la clave de sesión también para el criptado de cabida útil RTP.

No obstante, cuando se use también la DTMF a través del RTP y fijando la bandera **rtpPayloadIndication**, se recomienda enfáticamente que se asegure la cabida útil RTP mediante el perfil de criptación de voz de la cláusula D.7.

En el cuadro 2 se presentan los algoritmos de criptación disponibles (DES, 3DES o AES) que deberían utilizar el modo EOFB (incluyendo el modo OFB como un caso especial; véase B.2.5). Para evitar un posible relleno de caracteres DTMF, se recomienda no utilizar para la criptación de señales DTMF los modos CBC, CFB u otros modos de encadenamiento de bloques que puedan requerir el relleno.

8.7.1 Cadena básica criptada

Si se ha seleccionado **encryptedBasicString** en **UserInputCapability**, el **encryptedAlphanumeric** indicará qué algoritmo descriptado se aplica en el **algorithmOID**, y **paramS** tiene el valor inicial para la operación de criptación. Se colocará la cadena alfanumérica criptada en **encrypted**.

8.7.2 Cadena iA5 criptada

Si se selecciona **encryptedBasicString** en **UserInputCapability**, el **encryptedSignalType** tendrá el **ClearSignalType** criptado, donde **sig** transporta el caracter **signalType** de texto claro. **signalType** tendrá un valor ficticio "!" que será descartado por el recipiente.

algorithmOID indicará cuál algoritmo de criptación se aplica, y **paramS** tiene el valor inicial para la operación de encriptación.

8.7.3 Cadena general cryptada

Si se escoge una **encryptedGeneralString** en **UserInputCapability**, el **encryptedAlphanumeric** en el **extendedAlphanumeric** indicará el algoritmo de criptación aplicado dentro del **algorithmOID**, mientras que **alphanumeric** mantendrá una cadena vacía y **paramS** el valor inicial para la operación de criptación.

8.7.4 Lista de identificadores de objeto

Cuadro 2/H.235 – Identificadores de objeto para la criptación de DTMF H.245

Referencia de identificador de objeto	Valor(es) de identificador de objeto	Descripción
"DES-EOFB-DTMF"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 12}	Criptación de DTMF H.245 con DES-56 en modo EOFB
"3DES-EOFB-DTMF"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 13}	Criptación de DTMF H.245 con 3DES-168 en modo EOFB
"AES-EOFB-DTMF"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 14}	Criptación de DTMF H.245 con AES-128 en modo EOFB

8.8 Operación Diffie-Hellman

En esta Recomendación se soporta el protocolo Diffie-Hellman para el acuerdo de clave de extremo a extremo. Dependiendo de la situación, la clave Diffie-Hellman negociada puede funcionar como clave maestra (cláusula D.7) o como clave dinámica de sesión (anexo F y Rec. UIT-T H.530).

El sistema Diffie-Hellman se caracteriza por los parámetros de sistema g y p, donde p es un número primo grande y g indica el generador del grupo multiplicativo módulo p o de un subgrupo fuerte módulo p. g^x mod p indica la media clave Diffie-Hellman (pública) del llamante, mientras que g^y mod p la del llamado. En RFC 2412 se presenta más información acerca de ello y se aconseja cómo escoger parámetros Diffie-Hellman seguros.

La Rec. UIT-T H.235 transporta un ejemplar de Diffie-Hellman (g, p, g^x) codificado con un **ClearToken**, donde **dhkey** mantiene la **halfkey** g^x mod p (o g^y mod p, cuando sea el caso) para alguna x (o y) aleatoria secreta, el p primo en **modsize** y el g en **generator**. Un caso especial ocurre con la **dhkey** vacía, o lo que es lo mismo el trio (0, 0, 0), que no representa un ejemplar DH, pero que se utilizará para señalizar que no está utilizando el perfil de criptación de voz.

Los parámetros de sistema DH p y g se suelen fijar para un conjunto de aplicaciones con valores bien definidos, aunque es posible también que los sistemas extremos escojan su propio conjunto de parámetros. Conviene que el recipiente sepa que los parámetros DH no estándar pueden proporcionar menos seguridad de lo que podría parecer a primera vista; es decir, el llamante pudo haber escogido un número no primo, o es posible que g genere simplemente un pequeño subgrupo. Aunque en la práctica es imposible efectuar una prueba exhaustiva de los parámetros, depende de la política de seguridad del recipiente si se aceptan o rechazan dichas ofertas.

Para los parámetros de sistema DH fijo, es posible obtener mensajes codificados más compactos utilizando abreviaturas que incluyendo valores literales. Un **ClearToken** que transporte un ejemplar DH con parámetros DH fijos y normalizados, puede hacer referencia a este ejemplar mediante un DH OID en el campo **tokenOID**; al menos que **tokenOID** se utilice para otros efectos (por ejemplo para un **CryptoToken** particular, en D.6.3.2). Asimismo, el remitente puede incluir los valores literales DH, aunque no está obligado hacerlo.

Cuando se deban indicar varios ejemplares DH, cada uno mediante un DH-OID, se omitirán los parámetros DH en un **CryptoToken** particular (se trata en el anexo D), sin **dhkey** y cada ejemplar DH se transportará entonces en un **ClearTokens** independiente, donde el **tokenOID** mantiene el DH-OID y es posible que no haya **dhkey**; no se utilizará ningún otro campo en el **ClearToken**.

NOTA 1 – Esto no excluye la posibilidad de transportar un ejemplar DH en un **CryptoToken** particular u otros **ClearTokens** disponibles incluyendo literalmente los valores de los parámetros DH.

Cuando se deba indicar un ejemplar DH no estándar, se utilizará el DH-OID "DHdummy" y se proporcionarán explícitamente los parámetros de grupo DH no estándar en el **ClearToken**.

El llamante puede presentar uno o varios **ClearTokens** que transporta cada uno un ejemplar diferente Diffie-Hellman. Conviene que el llamante suministre el mayor número posible de ejemplares DH permitido por su política de seguridad. De esta manera, el recipiente puede escoger el ejemplar adecuado para la respuesta, incrementando así la probabilidad de encontrar un buen conjunto común de parámetros.

El recipiente escogerá y aceptará una única instancia DH (si se decide hacerlo) a partir del conjunto desordenado suministrado por el llamante en el mensaje SETUP. Cuando el recipiente pueda escoger un ejemplar DH conforme a sus propias necesidades de seguridad, no necesitará modificar un ejemplar DH propuesto o retornar uno que no haya sido enviado por el llamante. La solidez de los algoritmos de criptación de que disponen ambos puntos extremos durante la llamada debería corresponder a la solidez del ejemplar DH escogido entre los proporcionados que retorna el recipiente; véase el cuadro D.4. El recipiente indicará el ejemplar DH escogido en el mensaje de respuesta.

Cuando el llamado rechace cualquiera de las propuestas por razones de seguridad o debido a falta de capacidades de procesamiento, no incluirá **dhkey** en el mensaje de respuesta.

El recipiente incluirá su testigo DH en la respuesta **Setup**-a-**Connect**. Podrá también incluirlo en el mensaje de respuesta inmediatamente después del SETUP o después, pero en el peor de los casos en el mensaje CONNECT.

NOTA 2 – Es necesario tener en cuenta diversos aspectos al considerar cuándo el llamado puede incluir el (los) testigo(s) DH durante la respuesta **Setup**-to-**Connect**: el tiempo de respuesta, la carga de procesamiento en el recipiente, la capacidad de establecimiento de los canales de medios temprano, y otros más. Todos estos aspectos dependen de la implementación.

Es posible, sin embargo, que ciertos GK de encaminamiento no entreguen la respuesta **Setup**-to-**Connect** al llamante. Es decir, se pueden perder uno o varios mensajes de respuestas señalización de llamada H.225.0, incluido un posible testigo DH, y que por tanto no llegarían al llamante. En ese caso, éste no podría calcular la clave maestra y la(s) clave(s) de sesión de medios DH. Para evitar que esto ocurra, el llamado debería incluir siempre el mismo testigo DH en cada mensaje de respuesta **Setup**-to-**Connect**.

Cuando el DH-OID indique un ejemplar DH diferente del que se está transportando en **modsize** y **generator**, los valores literales transportados en estos dos parámetros tendrán prioridad sobre el DH-OID en el testigo. Para la respuesta, el llamado debería reemplazar el DH-OID que provoca conflicto por el DH-OID estático, es decir "DH1024", que corresponde al **modsize** y **generator** o "DHdummy" cuando no exista un DH-OID correspondiente.

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.

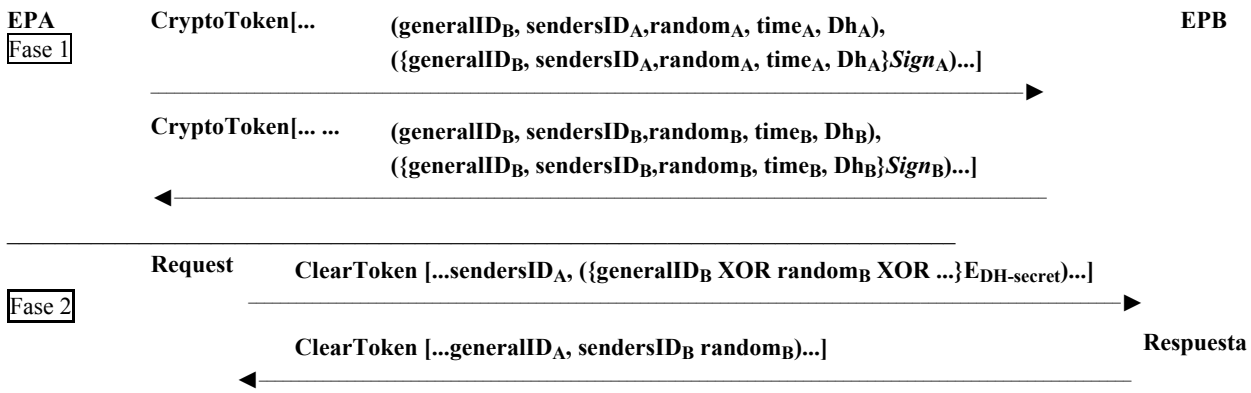
10.2 Intercambio Diffie-Hellman con autenticación facultativa

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 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 deriva 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 generación numérica;

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 seguro.

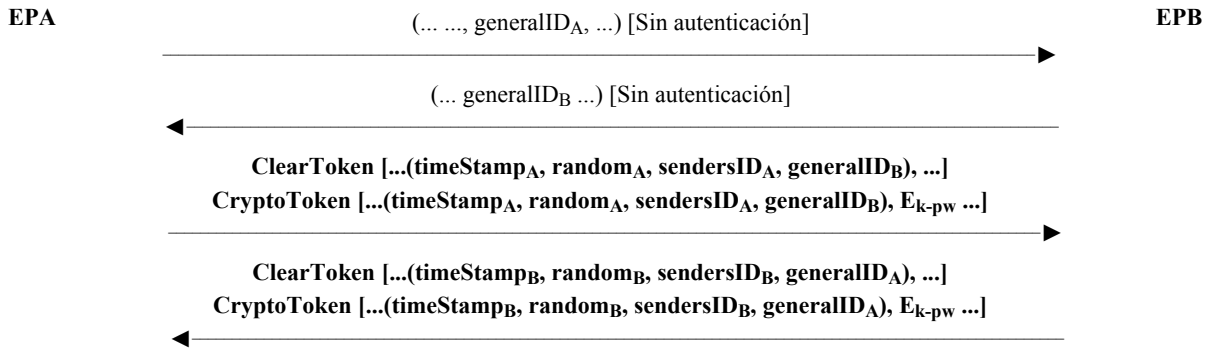
10.3.2 Contraseña con criptación simétrica

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).



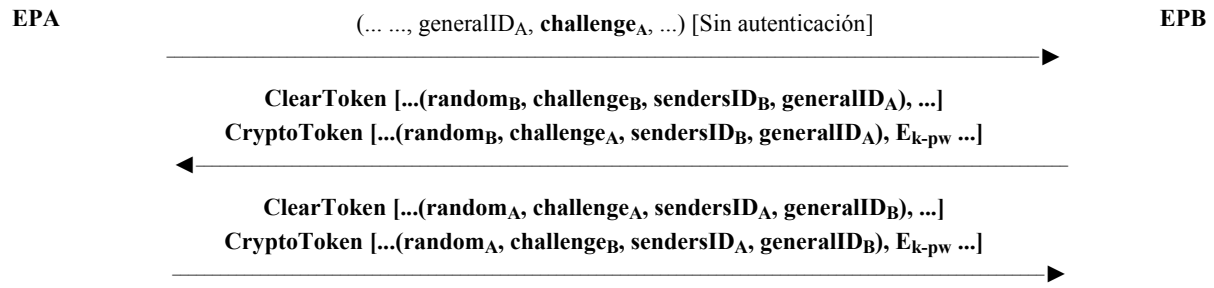
NOTA 1 – La devolución de testigo del EPB es facultativa; si se omite, sólo se logra una autenticación unidireccional.

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 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".

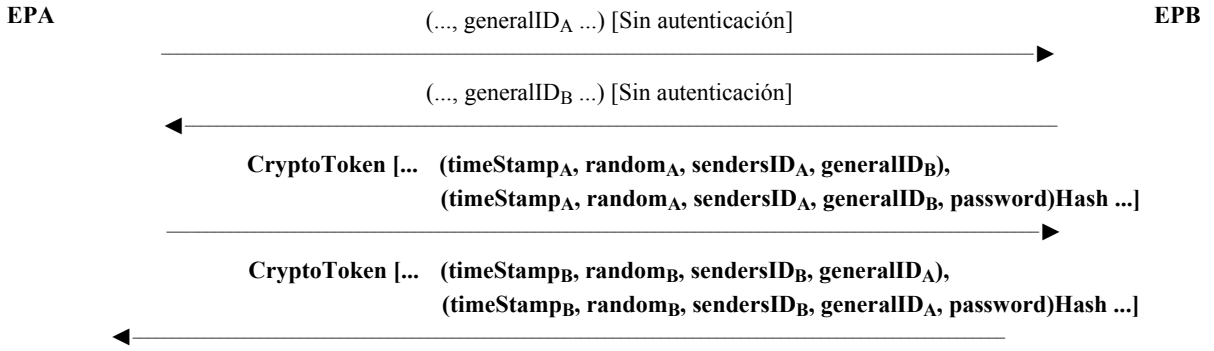
NOTA 3 – En el tercer mensaje el EPA proporciona una nueva **challenge$_A$** en texto claro en un **ClearToken** independiente, que es identificada por el mismo OID que el OID del **CryptoToken**. EPA también devuelve la **challenge$_B$** criptada como respuesta; y viceversa, sucede de manera similar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

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 generación numérica de dos pasos.

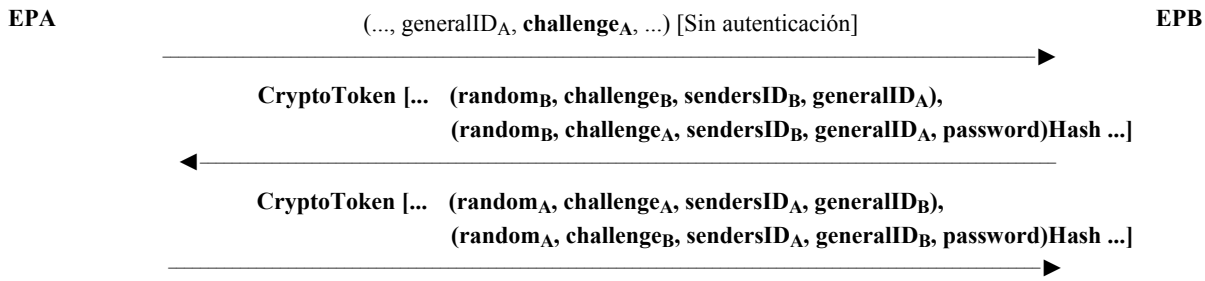


NOTA 1 – La devolución de testigo del EPB es facultativa; si se omite, sólo se logra una autenticación unidireccional.

NOTA 2 – **Hash** indica una función generadora 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 generación numérica; 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 generadora que opera sobre los valores contenidos.

NOTA 3 – En el tercer mensaje el EPA proporciona una nueva **challenge$_A$** en texto claro dentro del **ClearToken** insertado en **cryptoHashedToken**. EPA también devuelve la **challenge$_B$** 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 generación numérica; 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 número generador. Los implementadores deberán incluir la indicación de tiempo en el **hashedVals** y *no* deberán incluir la contraseña. (Por ejemplo, la contraseña y el '**generalID**' deben ser conocidos por el recipiente previamente; los primeros pueden omitirse.)

NOTA 2 – La función generadora 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.**

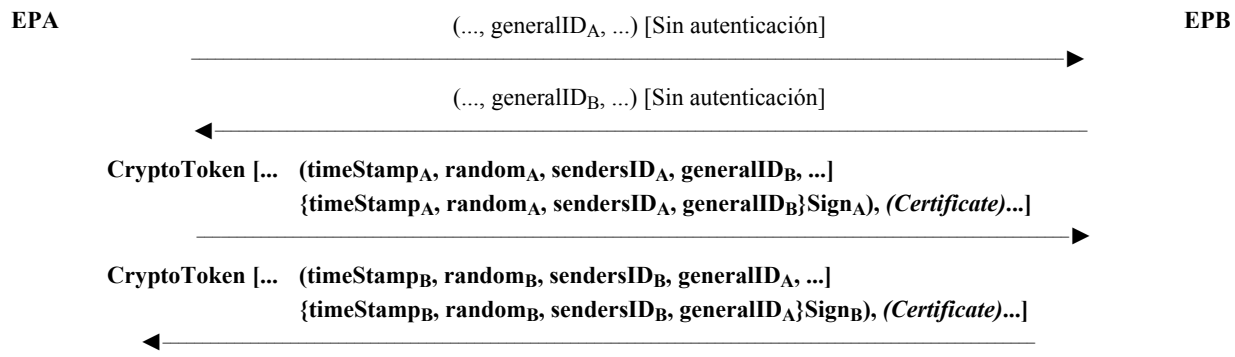
NOTA 3 – Las implementaciones deben garantizar que las contraseñas introducidas por el usuario transportan suficiente entropía. Las contraseñas que son demasiado cortas o que son vulnerables a los ataques de diccionario deben ser rechazadas. En determinados casos puede ser ventajosa la aplicación de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



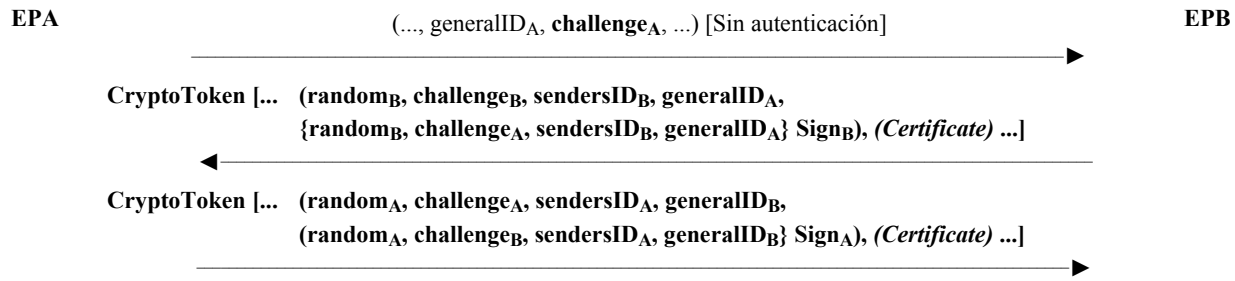
NOTA 1 – La devolución de testigo del EPB es facultativa; si se omite, sólo se logra una autenticación unidireccional.

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 generadora unidireccional criptográficamente fuerte.

Ejemplo recomendado, cuando se usa el perfil de seguridad del anexo D, es la aplicación de troceado SHA1 a la cadena contraseña, con lo que se obtiene un secreto compartido de 20 bytes. La ventaja es que el valor generador resultante 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 la Rec. UIT-T H.323, el encabezamiento RTP (SDU) cambiará su tipo de cabida útil para indicar la conmutación a la nueva clave.

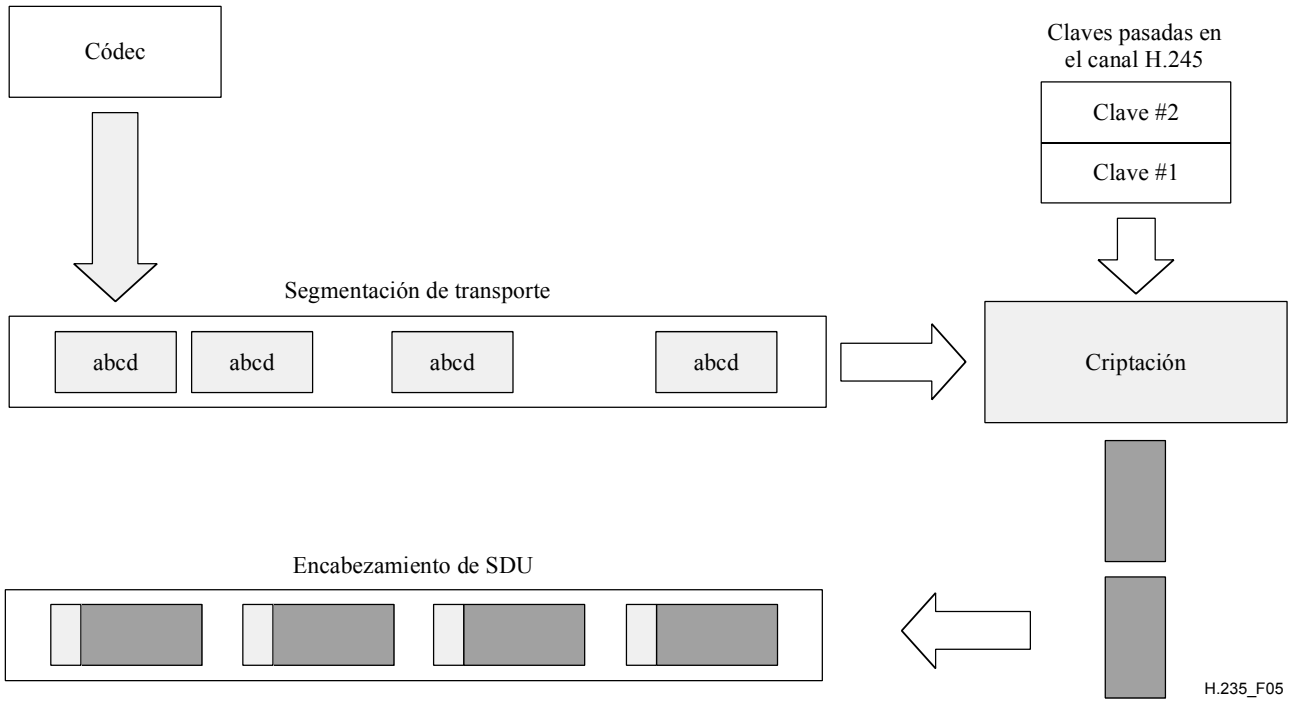


Figura 5/H.235 – Criptación de trenes de medios

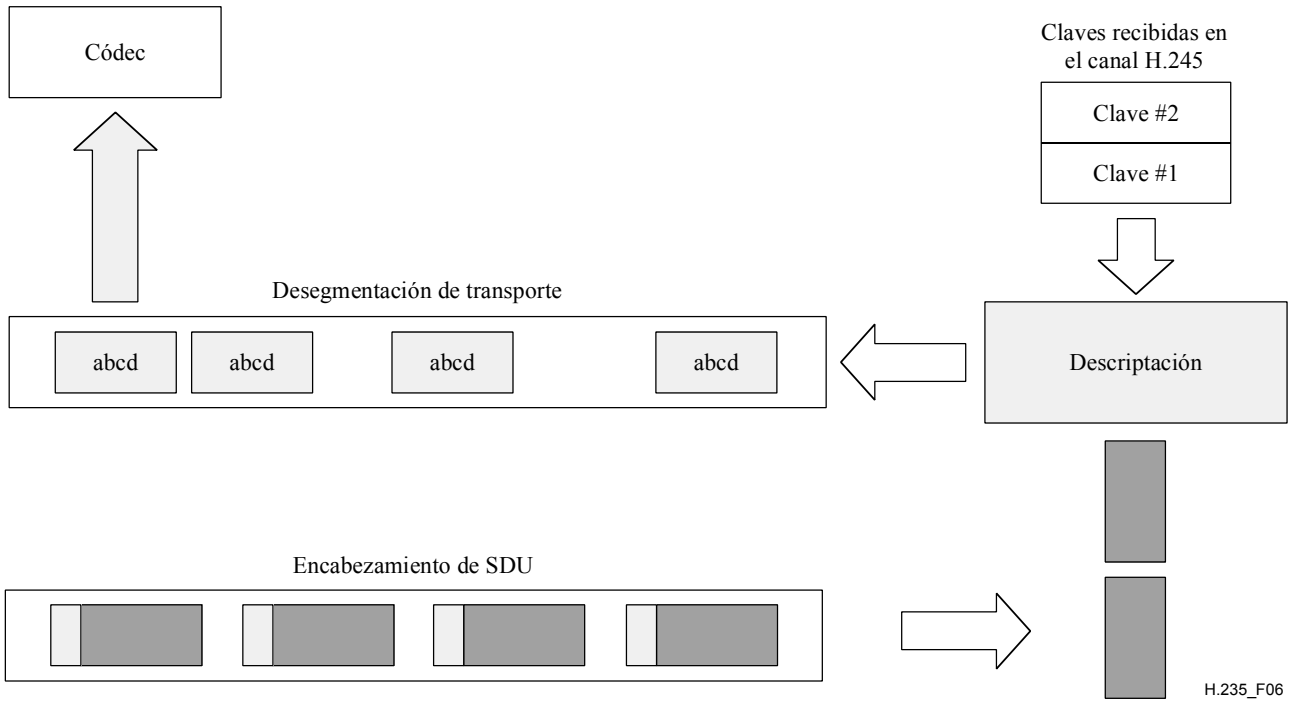


Figura 6/H.235 – Descriptación de trenes de medios

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 desear 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 "N", 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:

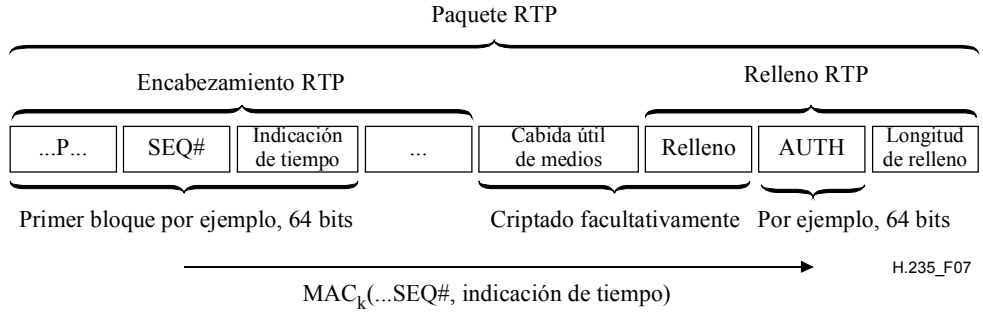


Figura 7/H.235 – Formato de paquete RTP para la antiinundación de medios

NOTA 1 – Si no se utiliza la antiinundación, tampoco se utilizan los campos AUTH 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 la Rec. 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 AUTH del relleno RTP. El emisor y el receptor conocen el tamaño del campo AUTH 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 generadora unidireccional SHA1;

k = SHA1(*K*); deberán tomarse suficientes bits del número generador resultante 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 3 se listan todas las referencias de los OID.

Cuadro 3/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 B-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 B-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 seguros (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,
     h235Key        H235Key OPTIONAL -- central distributed key in V3
}

--   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
ECGDSA Signature
} ( 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
     iv             OCTET STRING OPTIONAL, -- arbitrary length initialization
     vector
     clearSalt      OCTET STRING OPTIONAL -- unencrypted salting key for
     encryption
}

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 or ClearToken "h235Key"
field
{
     secureChannel           KeyMaterial,
     sharedSecret            ENCRYPTED {EncodedKeySyncMaterial},
     certProtectedKey        SIGNED {EncodedKeySignedMaterial },
     ...,
     secureSharedSecret      V3KeySyncMaterial -- for H.235 V3 endpoints
}
```

```
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)



V3KeySyncMaterial  ::= SEQUENCE
{
     generalID               Identifier OPTIONAL, -- peer terminal ID
     algorithmOID            OBJECT IDENTIFIER OPTIONAL, -- encryption algorithm
     paramS                  Params, -- IV
     encryptedSessionKey     OCTET STRING OPTIONAL, -- encrypted session key
     encryptedSaltingKey     OCTET STRING OPTIONAL, -- encrypted media salting
                                                    -- key
     clearSaltingKey         OCTET STRING OPTIONAL, -- unencrypted media salting
                                                    -- key
     paramSsalt              Params OPTIONAL, -- IV (and clear salt) for salting
                                                    -- key encryption
     keyDerivationOID        OBJECT IDENTIFIER OPTIONAL, -- key derivation
                                                    -- method
     ...
}


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 la Rec. UIT-T H.323.

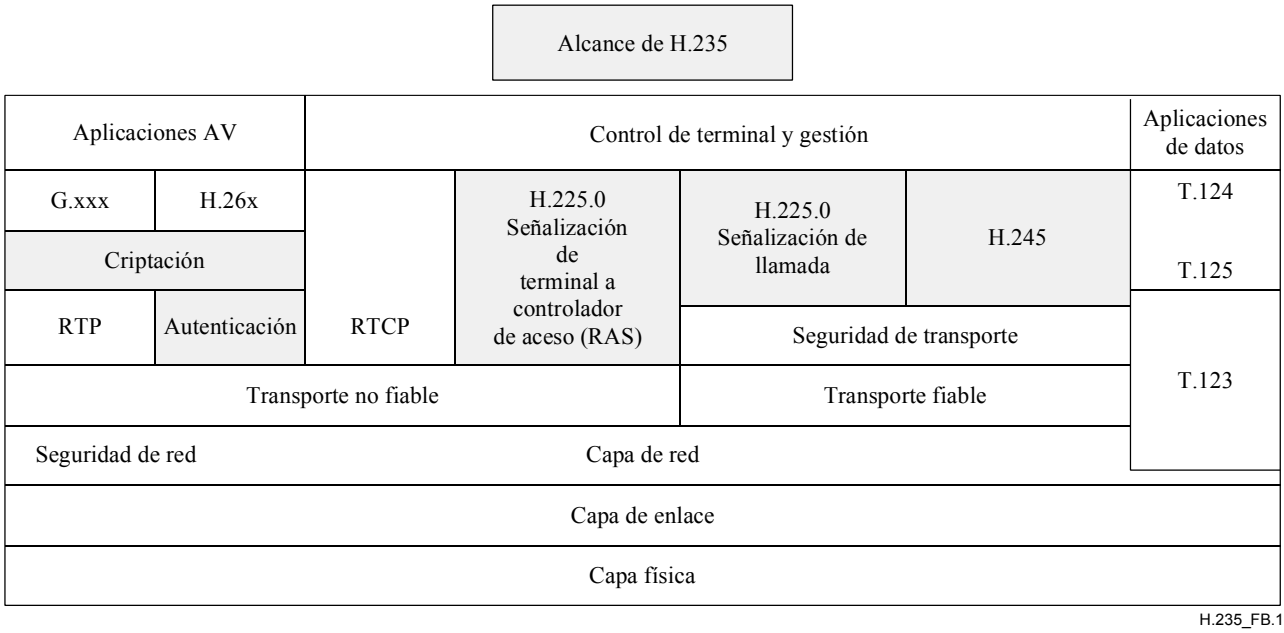


Figura B.1/H.235 – Visión general

Para el protocolo de la Rec. UIT-T 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 seguro, 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/o generación numérica (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 **Release Complete** 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** 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 **Release Complete** con un código ***undefinedReason***.

B.2.2 Señalización de error

Un controlador de acceso capaz de ofrecer seguridad, u otra entidad H.225.0 con seguridad mejorada, proporcionará indicaciones de error. Un error de seguridad indica que la entidad no pudo procesar correctamente el mensaje recibido. Siempre que sea posible, se proporcionará un código de error detallado.

- **securityWrongSyncTime** indicará que el remitente encontró un problema de seguridad relativo a indicaciones de tiempo inadecuadas. Esto podría deberse a un problema con el servidor de tiempo, una pérdida de sincronización o un retraso excesivo de red.
- **securityReplay** indicará que se ha encontrado un ataque de reproducción. Esto ocurre cuando se presenta más de una vez el mismo número de secuencia para una indicación de tiempo determinada.
- **securityWrongGeneralID** indicará una discordancia del identificador general en el mensaje. Esto podría deberse a un direccionamiento errado.
- **securityWrongSendersID** indicará una discordancia del identificador del remitente en el mensaje. Podría deberse a una entrada errónea del usuario.
- **securityIntegrityFailed** indicará que fracasó el test de integridad/firma. En el caso del anexo D, podría deberse a una contraseña errónea o mal escrita durante la petición inicial o a haberse encontrado un ataque activo. Para los anexos E/F, indicará que falló la prueba de firma digital en el mensaje. Podría deberse a la aplicación de una clave privada/pública errónea o a que se ha encontrado un ataque activo.

- **securityWrongOID** indicará cualquier discordancia en los OID de testigo (testigo despejado o criptado) o en los OID de algoritmo de criptación. Esto indica que se han implementado diversos algoritmos/perfiles de seguridad.
- **securityDHmismatch** indicará cualquier discordancia en los parámetros Diffie-Hellman intercambiados. Esto podría indicar que se han implementado diversos conjuntos de parámetros DH e incluso diversos algoritmos de criptación de voz.
- **securityCertificateExpired** indicará que ha expirado un certificado.
- **securityCertificateDateInvalid** indicará que aún no es válido un certificado.
- **securityCertificateRevoked** indicará que se encontró un certificado revocado.
- **securityCertificateNotReadable** indicará que no se pudo decodificar un certificado mediante ANS.1 o que está en otra forma inadecuada.
- **securityCertificateSignatureInvalid** indicará que la firma en el certificado es incorrecta.
- **securityCertificateMissing** indicará que no se encontró certificado donde se esperaba uno o que no pudo ser localizado de otra manera.
- **securityCertificateIncomplete** indicará que no estaban presentes algunas extensiones de certificados esperadas.
- **securityUnsupportedCertificateAlgOID** indicará que no se entendieron o no se soportan algunos algoritmos de criptación tales como generación numérica (hash) o las firmas digitales, utilizados en certificados. El remitente puede proporcionar, como parte de la respuesta, una lista de los certificados aceptables en testigos separados, para facilitar al recipiente escoger uno adecuado.
- **securityUnknownCA** indicará que no se pudo encontrar el certificado CA/root o que no fue posible hacerlo corresponder con un CA de confianza.

Cualquier otro fallo de una operación de seguridad H.235 implicará el retorno de un **securityDenial** para RAS H.225.0 (o **securityDenied** en el caso de la señalización de llamada H.225.0).

NOTA 1 – En los perfiles de seguridad de los anexos D, E o F pueden aparecer securityWrongSyncTime, securityReplay, securityWrongGeneralID, securityWrongSendersID, SecurityIntegrityFailed, securityDHmismatch**,** y securityWrongOID.

NOTA 2 – En los perfiles de seguridad de los anexos E o F pueden aparecer securityCertificateExpired, securityCertificateDateInvalid, securityCertificateRevoked, securityCertificateNotReadable, securityCertificateSignatureInvalid, securityCertificateMissing, securityCertificateIncomplete, securityUnsupportedCertificateAlgOID y securityUnknownCA.

B.2.3 Indicación de característica de la versión 3

Los puntos extremos conformes a la versión 3 y versiones superiores de la Rec. UIT-T H.235 proporcionan procedimientos mejorados de seguridad en el trayecto de medios que no soportan las versiones 1 y 2, a saber:

- el transporte de clave mejorado (**V3KeySyncMaterial**, véase B.2.4.1),
- la actualización de clave mejorada, véase B.2.6.2.

Puesto que suele ocurrir que los puntos extremos no sepan que soportan mutuamente la versión 3 de H.235, se añade durante el establecimiento de la comunicación una indicación explícita de la versión utilizada.

Los puntos extremos conformes a la versión 3 y versiones superiores de la Rec. UIT-T H.235 deberían utilizar siempre el procedimiento descrito en esta cláusula para determinar las capacidades de la versión 3 (transporte de clave mejorado, sincronización de criptación mejorada). Dependiendo del resultado del proceso de señalización lógica, los puntos extremos pueden utilizar los procedimientos (véase B.2.4) para la compatibilidad con los puntos extremos de las versiones 1 ó 2 de dicha Recomendación.

Con el fin de indicar si se utilizan los procedimientos mejorados de la versión 3 de la Rec. UIT-T H.235, los puntos extremos llamante y llamado incluirán un **ClearToken** adicional que indique la capacidad versión 3 durante la señalización de llamada (SETUP, CONNECT, etc.). La ausencia de dicho **ClearToken** indicaría que se soporta solamente la versión 1 o la versión 2. En este caso, el punto extremo utilizará el procedimiento B.2.4. De lo contrario, puede utilizar los procedimientos mejorados que se describen en B.2.4.1, o el procedimiento B.2.4 de la versión 1 o de la versión 2.

Dicho **ClearToken** utilizará **tokenOID** puesto a "V3", y se le asigna el siguiente valor.

"V3"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 24}	Indicación de capacidad de versión 3 en ClearToken durante la señalización de llamada

Todos los demás campos en dicho **ClearToken** permanecerán inutilizados, a menos que se usen para transportar parámetros DH.

B.2.4 Transporte de clave

El terminal director generará material de clave de sesión y lo distribuirá al(los) par(es). Existen dos procedimientos para el transporte de clave:

- Un procedimiento destinado en principio a los puntos extremos de las versiones 1 ó 2 de la Rec. UIT-T H.235; descrito en esta cláusula.
- Un procedimiento mejorado para los puntos extremos conformes a la versión 3 y versiones superiores de la Rec. UIT-T H.235, descrito en B.2.4.1.

Los puntos extremos de la versión 1 o versión 2 de la Rec. UIT-T H.235 aplican el siguiente procedimiento para el transporte de clave de sesión:

KeySyncMaterial mantiene el identificador de punto extremo del maestro en **generalID** y transporta el material clave de sesión en **keyMaterial**. Debería incluirse el valor **generalID** para proporcionar un nivel mínimo de autenticación a la fuente de la clave de sesión (véase también D.7.2). El recipiente debería verificar si el **generalID** recibido es correcto.

NOTA – En esta Recomendación se supone que cada punto extremo se ha registrado con un controlador de acceso y ha obtenido un identificador de punto extremo que se puede transportar en **generalID**. En esta Recomendación no se soporta la opción sin controladores de acceso; que queda en estudio.

KeySyncMaterial será criptado utilizando la clave maestra negociada. El **KeySyncMaterial** se rellenará siempre hasta un múltiplo de bloques antes de criptarlo, donde el último octeto se fijará al número de octetos de relleno (incluido el último). El valor del relleno debería determinarse utilizando la convención normal del algoritmo de cifrado. Se almacenará el resultado de la criptación en **sharedSecret** de **H235Key**.

B.2.4.1 Transporte de clave mejorado en la versión 3 de H.235

Se ha observado que la definición de sintaxis ASN.1 de **KeySyncMaterial** y la manera como se aplica la operación ENCRYPTED{} a los datos de las versiones 1 y 2 de la Rec. UIT-T H.235 dejan bastante texto claro conocido: en primer lugar el **generalID** del terminal director, aunque también algunos bits de codificación conocidos para la estructura. Incluso durante la criptación, el **generalID** se distingue de otras partes no criptadas del mensaje de señalización (por ejemplo **senderID**). Se cree que la presencia de dicho texto claro conocido debilita significativamente el esquema de seguridad, puesto que un atacante podría violar con más facilitad la clave de sesión, especialmente si se trata de un cifrado de bloques cuyo tamaño de bloque sea menor que DES-56 o compatible con RC2.

Además, la versión 3 de la Rec. UIT-T H.235 será capaz de transportar otro material de clave:

- Transporte seguro de clave adicional al (los) par(es). Dicha clave se introduce para el modo OFB mejorado; véase B.2.5.

La versión 3 de la Rec. UIT-T H.235 amplía **H235Key** con un **secureSharedSecret** que contiene **V3KeySyncMaterial**, que a su vez mantiene los siguientes parámetros:

generalID mantiene el identificador de punto extremo del remitente de origen, si lo hay, o de lo contrario no se usa.

algorithmOID indica el algoritmo de criptación aplicado y el modo de funcionamiento.

paramS mantiene el valor de inicialización, que se aplica para la criptación de la(s) clave(s) transportada(s).

NOTA 1 – El IV en **paramS** no debería confundirse con el IV de paquete RTP que no está siendo señalado. Como opción, **ClearSalt** mantiene una clave adicional sin codificar para la criptación de clave de sesión (por ejemplo, para EOFB).

encryptedSessionKey mantiene el texto cifrado de la clave de sesión raw criptada.

encryptedSaltingKey mantiene el texto cifrado de la clave adicional, de medios raw cifrada, si la hubiese. La clave adicional es necesaria para el modo OFB mejorado.

clearSaltingKey puede mantener la clave adicional de medios raw sin criptar. En las implementaciones habrá que asegurarse de que no se utilicen simultáneamente **encryptedSaltingKey** y **clearSaltingKey**.

paramSsalt mantiene el valor inicial para la criptación de la clave adicional. Como opción, **ClearSalt** mantiene una clave adicional sin criptar para la criptación de clave adicional (por ejemplo, para EOFB).

NOTA 2 – **generalID**, **algorithmOID** y **paramS** se transmiten siempre en texto claro, mientras que **encryptedSessionKey** y **encryptedSaltingKey** mantienen el texto cifrado del material clave criptado.

El terminal director genera la(s) clave(s) conforme a las capacidades de terminal negociadas y la(s) envía al (los) punto(s) extremo(s) par(es) utilizando **V3KeySyncMaterial**. Los controladores de acceso intermedios, si los hay, reenviarán el **V3KeySyncMaterial** sin modificación.

Los puntos extremos de la versión 3 o versiones superiores de H.235 deberían utilizar siempre **secureSharedSecret** en **H235Key**, pero también pueden según el resultado del procedimiento de señalización lógico de B.2.3 y utilizando la indicación **ClearToken** de versión 3, utilizar **sharedSecret** para la compatibilidad con los puntos extremos de las versiones 1 ó 2.

B.2.5 Modo OFB mejorado

El modo OFB (ISO/CEI 10116) define un modo de funcionamiento que utiliza un cifrado de trenes con algoritmos de criptación de bloque. Este modo proporciona:

- calidad de funcionamiento mejorada gracias a un retraso reducido del procesamiento de criptación,
- un manejo más fácil y menos complejo de los bloques incompletos,
- buena resistencia contra los errores de bits.

El modo OFB mejorado es una ligera modificación del modo OFB que se llamará de aquí en adelante modo de "retroalimentación de salida mejorado" (EOFB), y que además de las características o el OFB tiene las siguientes:

1) utiliza una clave adicional KS (*salting key*) además de la clave de criptación KE (*encryption key*) y
2) introduce un índice de paquete implícito.

La utilización de una clave adicional KS secreta, a la que se aplica una operación XOR con el resultado de la retroalimentación, produce más seguridad contra el análisis del texto claro conocido. Éste es un beneficio de seguridad importante que no puede ser proporcionado por otros modos de funcionamiento estándar (tales como CBC, OFB etc.). El uso del modo EOFB conllevaría entonces a un incremento de seguridad contra los textos claros de alta redundancia y contra el análisis de los textos claros conocidos.

El método EOFB se define como $C_i = P_i \oplus S_i$, con $S_i = E_{KE}(KS \oplus S_{i-1})$ para $i = 1 \ldots n$, y $S_0 = IV$, donde C_i es el i-ésimo bloque de texto cifrado, P_i el i-ésimo bloque de texto claro, S_i el i-ésimo bloque de tren de clave, KE la clave de criptación y $\oplus$ el XOR basado en bit. En la figura I.4.1 se ilustra el EOFB.

Es posible también que el EOFB funcione en el modo OFB normal, de tal manera que sea compatible con éste. Siempre que se desee dicha compatibilidad, se fijará la clave adicional KS bien a todo cero o bien se dejará vacío **encryptedSaltingKey** en **V3KeySyncMaterial**. No obstante, es altamente recomendable utilizar una clave adicional real para aquellos casos en que se cripten cabidas útiles RTP cuyo cifrado de bloque tenga un tamaño de bloque menor, como por ejemplo DES-56 o compatible con RC2.

Tras haber procesado al menos 2^{48} paquetes, se utilizará una nueva clave de criptación de sesión KE y una nueva clave adicional KS, o de lo contrario podría ocurrir una reutilización de trenes de clave lo que pondría en riesgo la seguridad.

En el anexo D se definen los identificadores de objeto para DES56-EOFB, compatible RC2, EOFB, 3-DES-EOFB y AES-EOFB.

B.2.6 Actualización y sincronización de clave

Las claves de sesión de medios tienen una vida útil limitada. En algún momento, toda clave expira. Se debería entonces utilizar una clave nueva para proteger la sesión de seguridad en curso. En los entornos de conferencia, se debería definir y distribuir una nueva clave de sesión de grupo cuando los miembros del grupo se unan o se retiren de la conferencia, evitando así que puedan acceder datos pasados o futuros, respectivamente.

- La actualización y sincronización de clave basada en el tipo de cabida útil define un nuevo tipo de cabida útil dinámica para la nueva clave de sesión; véanse B.2.6.1, B.2.6.2 y B.2.6.3.

A efectos de la actualización de clave, en esta Recomendación se ofrece un procedimiento de toma de contacto sin acuse, que se aplica también para los puntos extremos de las versiones 1 y 2 de la Rec. UIT-T H.235 y también uno robusto y con acuse para la versión 3 y versiones superiores.

B.2.6.1 Actualización de clave sin acuse

En la figura B.1.1 se muestra el procedimiento de toma de contacto sin acuse para la distribución o actualización de claves de sesión. Si el terminal subordinado desea una clave de sesión actualizada, puede pedir una nueva clave de sesión al terminal director enviándole una **encryptionUpdateRequest** a éste. El terminal director enviará una nueva clave de sesión (con una **encryptionUpdateRequest** anterior o sin ella del subordinado) al subordinado dentro de un mensaje **EncryptionUpdate**.

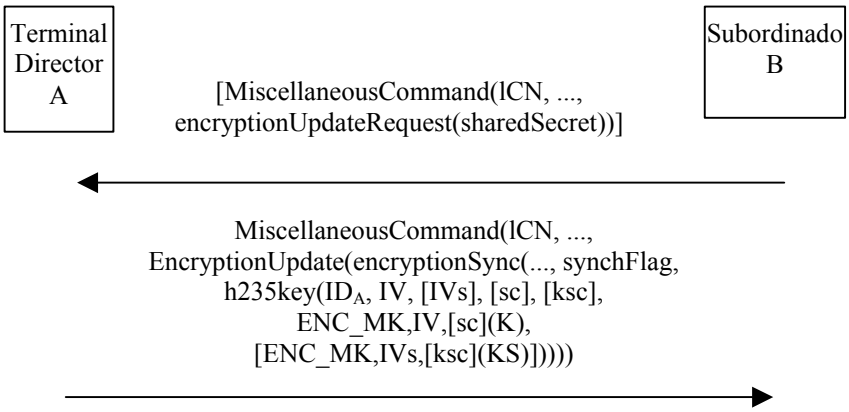


Figura B.1.1/H.235 – Distribución o actualización de clave de sesión sin acuse del terminal director a(los) subordinado(s)

donde:

lCN es el número de canal lógico;

synchFlag es el nuevo número de cabida útil RTP dinámica;

ID_A es el **generalID** del origen;

IV es el valor o vector inicial para la criptación de la sessionkey;

IVs es el valor o vector inicial para la criptación de la clave adicional;

ENC_MK,IV,sc(K) indica la criptación del texto claro *K* utilizando clave *M*, *IV* vector inicial [y una clave adicional *sc*, solamente para EOFB];

KS es la clave adicional para el medio (solamente para el modo EOFB);

K es la clave de sesión de texto claro;

sc es la clave adicional sin criptar, cuando se ha venido utilizando el modo EOFB para la criptación de la clave sesión;

ksc es la clave adicional sin criptar cuando se ha venido utilizando el modo EOFB para la criptación de la clave adicional;

s2M/m2S es la bandera **direction** (solamente para la versión 3 de H.235) (s2m = subordinado a director, m2s = director a subordinado);

[] representa algo facultativo.

En los métodos de actualización de clave descritos en las cláusulas siguientes se puede desplegar el modo de criptación EOFB para proteger el material clave transmitido. Para ello, de la misma manera que para la protección de la cabida útil de medios, se debe utilizar una clave adicional (sc o ksc).

B.2.6.2 Actualización de clave mejorada

Los puntos extremos conformes a la versión 3 de la Rec. UIT-T H.235 y a versiones superiores ejecutarán un procedimiento de actualización de clave con acuse explícito o implícito. De esta manera, se proporcionan métodos fiables de actualización de clave, que se basan en el método de actualización de claves sin acuse suministrado por las versiones anteriores a la 3. La capacidad de dicho procedimiento se negociará utilizando la indicación de característica de la versión 3, según la cláusula B.2.3

En la figura B.1.2 se muestran los procedimientos de actualización de clave para un canal lógico que pertenece al subordinado. Cuando éste inicie la actualización de clave y solicite una nueva clave de sesión al terminal director, el subordinado enviará una **MiscellaneousCommand** al

director, donde **logicalChannelNumber** mantendrá el número de canal lógico (definido por el subordinado), **sharedSecret** se fijará a verdadero, la bandera **direction** se fijará a **slaveToMaster** y se solicitará el nuevo número de cabida útil dinámica en **synchFlag** dentro de **EncryptionUpdateRequest**. De lo contrario, si el director inicia la actualización de clave, no se enviará este mensaje **EncryptionUpdateRequest**.

El director emitirá, bien como respuesta a una petición del subordinado o bien en nombre propio, una **EncryptionUpdateCommand** en la que **logicalChannelNumber** mantendrá el número de canal lógico, **direction** se fijará a **slaveToMaster** en **MiscellaneousCommand**, y **synchFlag** dentro de **encryptionSync** refleja el nuevo número de cabida útil dinámica. **h235key** transportará la nueva clave de sesión, y mantendrá la identidad del director en **generalID** y el vector inicial aplicado *IV* en **paramS**. La clave de sesión de medios criptada se transportará en **encryptedSessionKey**, para el que la función de criptación aplicará la clave de sesión maestra y el valor inicial en **paramS** a la clave de sesión *K*. Para EOFB, se transporta una clave adicional sin criptar en **ClearSalt** dentro de **paramS** (*sc*). **encryptedSaltingKey** transportará la clave adicional de medios criptada, con la función de criptación aplicando la clave de sesión maestra y el valor inicial **paramSsaltIV** a la clave adicional de medios *KS*. Para EOFB, se transporta una clave adicional no criptada (*ksc*) en **ClearSalt** dentro de **paramSsalt**. **clearSaltingKey** puede mantener una clave adicional de medios sin criptar, en cuyo caso **encryptedSaltingKey** permanecerá vacía y viceversa. La transmisión de una clave adicional sin criptar se logrará solamente si no afecta la seguridad, de lo contrario se recomienda criptar la clave adicional de medios.

El terminal director estará preparado para recibir medios criptados con la nueva clave de sesión tras presentar el **EncryptionUpdateCommand**, pero debería seguir utilizando la clave antigua hasta la recepción del **EncryptionUpdateAck**. El director puede aplicar la nueva sesión a partir de la recepción del **encryptionUpdateAck**, mientras que el subordinado puede hacerlo a partir de la recepción del **EncryptionUpdateCommand.**

NOTA 1 – El director puede escoger cualquier valor de tipo de cabida útil dinámico para el subordinado, puesto que el tipo de cabida útil depende solamente del puerto de canal de medios.

NOTA 2 – No es necesario que el subordinado acuse explícitamente recepción de la nueva clave. El director puede deducir que aquél ya la recibió cuando le lleguen medios criptados mediante el nuevo tipo de cabida útil.

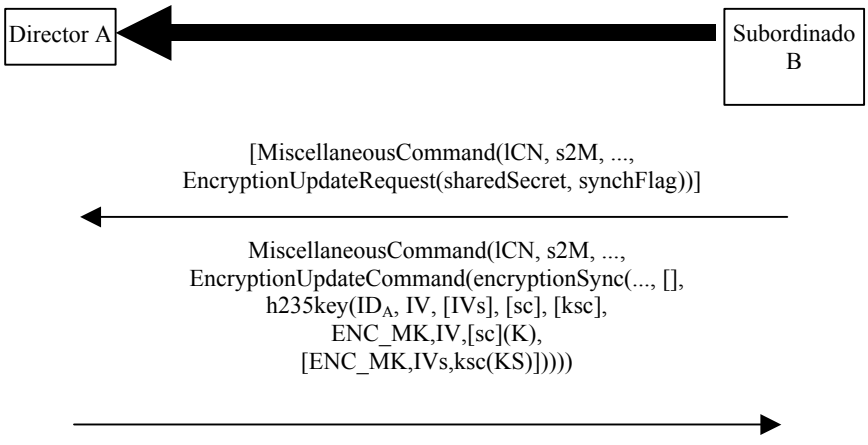


Figura B.1.2/H.235 – Actualización de clave de sesión en el canal lógico del subordinado

En la figura B.1.3 se muestran los procedimientos de actualización de clave para un canal lógico propiedad del director. Si el subordinado inicia la actualización de clave y solicita una nueva clave de sesión al director, aquél enviará una **MiscellaneousCommand** a éste, donde **logicalChannelNumber** mantendrá el número de canal lógico (definido por el director),

sharedSecret se fijará a verdadero, y la bandera **direction** se fijará a **masterToSlave**. De lo contrario, si el director inicia la actualización de clave, no se enviará este mensaje **EncryptionUpdateRequest**.

El director emitirá, como respuesta una petición del subordinado o en su propio nombre, una **EncryptionUpdateCommand** donde **logicalChannelNumber** mantendrá el número de canal lógico, **direction** se fijará a **masterToSlave**, **encryptionSync** proveerá la **synchFlag** con el nuevo número de cabida útil dinámica. **h235key** transportará la nueva clave de sesión y mantendrá la identidad del director en **generalID** y el vector inicial applied *IV* en **paramS**. La clave de sesión de medios criptada será transportada en **encryptedSessionKey**, donde la función de criptación aplicará la clave maestra y el valor inicial en **paramS** a la clave de sesión *K*. Para EOFB, se transporta una clave adicional sin criptar en **ClearSalt** dentro **paramS** (*sc*). Para EOFB, **encryptedSaltingKey** transportará la clave adicional de medios criptada, donde la función de criptación aplicará la clave de sesión maestra y el valor inicial **paramSsaltIV** a la clave adicional *KS*. Para EOFB, se transporta una clave adicional sin criptar (*ksc*) en **ClearSalt** dentro de **paramSsalt**. **clearSaltingKey** puede mantener una clave adicional de medios sin criptar, en cuyo caso **encryptedSaltingKey** permanecerá vacía y viceversa. La transmisión de una clave adicional sin criptar se logrará solamente si no se afecta la seguridad, de lo contrario se recomienda criptar la clave adicional de medios.

El subordinado acusará recibo de recepción de la nueva clave de sesión respondiendo con una **MiscellaneousCommand**, donde el **logicalChannelNumber** mantendrá el número de canal lógico y **encryptionUpdateAck** indicará el nuevo número de cabida útil dinámica en la **synchFlag**.

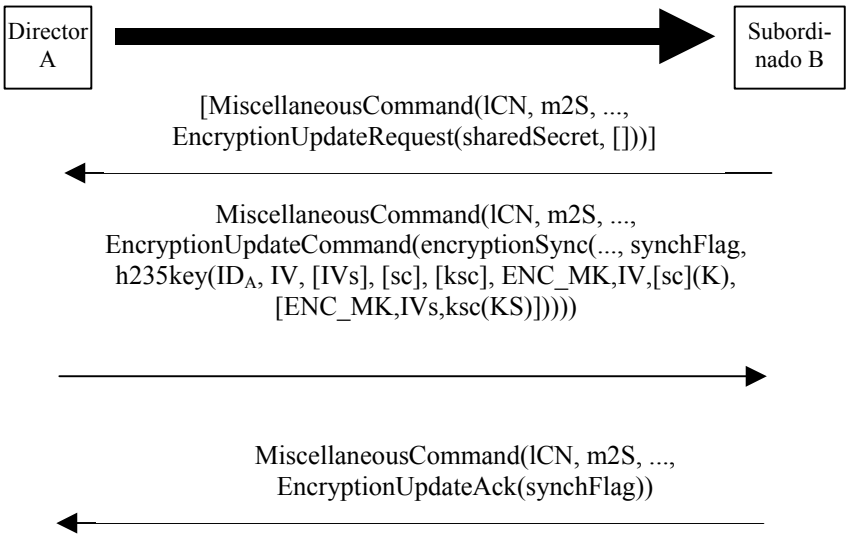


Figura B.1.3/H.235 – Actualización de clave de sesión en el canal lógico de terminal director

B.2.6.3 Actualización y sincronización de clave basada en el tipo de cabida útil

El terminal director presenta la clave de criptación inicial junto con el número de cabida útil dinámica en **synchFlag** (a través de **EncryptionSync** en la Rec. UIT-T H.245). El(los) receptor(es) del tren de medios empezará(n) a utilizar la clave tras recibir este número de cabida útil en el encabezamiento RTP.

Si el canal lógico negociado transporta sólo un tipo de cabida útil, el valor de la **synchFlag** puede reemplazar el tipo de cabida útil negociado en el encabezamiento RTP. Por otra parte, si el canal lógico negociado puede transportar más de un tipo de cabida útil (incluso si lo hace solamente en paquetes RTP separados), los paquetes RTP tendrán el formato descrito en RFC 2198, el valor **synchFlag** será el tipo de cabida útil encapsulamiento, y el tipo o tipos de cabida útil real estarán en el bloque o bloques adicionales de encabezamiento, como se especifica en RFC 2198.

El punto extremo director puede distribuir en cualquier momento una o varias nuevas claves. La sincronización de la clave más reciente con el tren de medios se indicará cambiando el tipo de cabida útil a un nuevo valor dinámico.

NOTA – Los valores concretos pueden ser cualesquiera, siempre que cambien para cada nueva clave que se distribuya.

B.3 Aspectos relativos a RTP/RTCP

La utilización de criptación en el tren RTP seguirá la metodología general recomendada en el documento al que se hace referencia en [RTP]. La criptación de los medios se producirá de manera independiente, paquete por paquete[1]. El encabezamiento RTP no será criptado. Para los códecs de audio/vídeo, se criptará toda la cabida útil de códec de audio/vídeo, incluido(s) todos los encabezamientos de cabida útil de audio/vídeo. La sincronización de nuevas claves y textos criptados se basa en el tipo de cabida útil dinámica (véase B.2.6.3).

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 debería descifrarse independientemente. Dos requisitos del modo algoritmo de bloque funcionarán como sigue:

B.3.1 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.

B.3.1.1 Vector de inicialización CBC

Se requiere un IV cuando se utilice un cifrado de bloque en el modo CBC para criptar cabidas útiles de paquetes RTP. El tamaño de un IV es igual al tamaño de bloque para el cifrado de bloque correspondiente. Por ejemplo, el tamaño IV para DES y 3-DES es 64 bits, mientras que para AES es 128 bits.

Para el caso CBC, el IV se construirá a partir de los primeros *B* octetos (donde *B* es el tamaño de bloques) de: Seq# concatenado con + Timestamp. Esto forma el patrón, *SSTTTT*, donde *SS* es el Seq# RTP de 2 octetos y *TTTT* es la indicación de tiempo RTP de 4 octetos. Este esquema se repetirá hasta que se hayan generado *B* octetos, truncando siempre que sea necesario. Por ejemplo, los IV de 64 y 128 bits podrían contener *SSTTTTSS* y *SSTTTTSSTTTTSSTT*, respectivamente. Nótese que el IV generado de esta manera puede producir un esquema de clave considerado "débil" en ciertos algoritmos.

---

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

B.3.1.2 Vector de inicialización EOFB

El vector inicial IV único para cada paquete RTP en el modo EOFB se calculará de la siguiente manera:

Se asocia cada paquete RTP con un índice *i* de paquete de 48 bits implícito, como se define en [SRTP], donde $i = 2^{16} \times \text{ROC} + \text{SEQ}$, y para el que SEQ es el número de secuencia tomado del encabezamiento RTP y ROC el contador de incremento de 32 bits (cuántas veces el número de secuencia SEQ ha vuelto a 65535).

Para comenzar, el contador de incremento ROC se fijará a cero. Cada vez que el SEQ llega al módulo 2^{16}, el remitente incrementará ROC en un módulo 2^{32}.

El vector inicial IV se calcula como ($i \parallel T$ [$\parallel i \parallel T \parallel \ldots$]) con el índice *i* de 48 bit y el indicador de tiempo *T* de 32 bit tomados del encabezamiento RTP concatenado varias veces hasta que se llena completamente el tamaño de bloques. El símbolo || indica concatenación.

NOTA – El contador de recomienzo y el IV se mantienen y calculan localmente en cada extremo par, y no se transmiten.

Cuando el receptor tenga que hacer frente a paquetes perdidos o reordenados debería calcular un índice *i* mediante:

$i = 2^{16} \times v + \text{SEQ}$, donde *v* se escoge del conjunto {ROC-1, ROC, ROC+1} modulo 2^{32}, de tal manera que sea el más cercano respecto al valor $2^{16} \times \text{ROC} + s_l$ ("en el sentido de" 2^{48}) donde s_l es el número de secuencia mantenido en el receptor. Tras haber procesado el paquete utilizando el índice así calculado, el receptor decidirá si hay que actualizar s_l y ROC. Por ejemplo, un método simple (pero no muy resistente a los errores) consiste en simplemente fijar s_l a SEQ (si $\text{SEQ} > s_l$) y, si el valor $v = \text{ROC} + 1$ ha sido utilizado, actualizar ROC a *v*; en [SRTP, sección 3.2.1] se puede encontrar más información al respecto.

B.3.2 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, $N (\leq B)$, 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 bloques incompletos ECB y CBC; sin relleno para CFB y OFB.

2) Relleno de la manera prescrita por [RTP, sección 5.1].

[RTP, sección 5.1] describe un método de relleno en el cual la cabida útil se rellenará hasta un múltiplo de bloque. El último octeto se fijará 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.

B.3.3 Protección RTCP

La aplicación de técnicas criptográficas a los elementos RTCP queda en estudio.

B.3.4 Tren de cabida útil seguro

Las redes basadas en H.323 suelen utilizar, por ejemplo para la transmisión mediante módem en el IP, señalización H.245 para establecer y negociar un canal de datos de banda local y RTP para la paquetización de un tren de cabida útil múltiple (MPS, *multiple payload stream*).

En el caso de un tren de medios único con un solo tipo de cabida útil o FEC para otro canal, el tipo de cabida útil dinámica en **encryptionSync** reemplazará el tipo de cabida útil por defecto.

Para los trenes encapsulados (es decir, con codificación de redundancia o con FEC la codificada según RFC 2198) el tipo de cabida útil dinámico en **encryptionSync** reemplazará el tipo de cabida útil de encapsulamiento.

Para los trenes de cabida útil múltiple se ignorará el tipo de cabida útil dinámico en la **syncFlag** de **encryptionSync**, y se utilizarán en su lugar los tipos de cabida útil (facultativos) en el (los) **multiplePayloadStreamElement**(s).

En el procedimiento mejorado de actualización de clave, se utilizará la **EncryptionUpdateCommand** para distribuir nuevo material clave de sesión (véase B.2.6.2). **multiplePayloadStream** se utiliza solamente cuando se debe reotorgar una clave a un tren de cabida útil múltiple, en cuyo caso se ignorará el tipo de cabida útil dinámico en **EncryptionSync**.

B.3.5 Interfuncionamiento con J.170

Queda en 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, 10.3.2, 10.3.3 y 10.3.4. 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 se utilizará en todo **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**) u otro código de error de seguridad adecuado, conforme a B.2.2 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 la Rec. 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 generación numérica;
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 u otro código de error de seguridad adecuado, conforme a B.2.2. 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**.

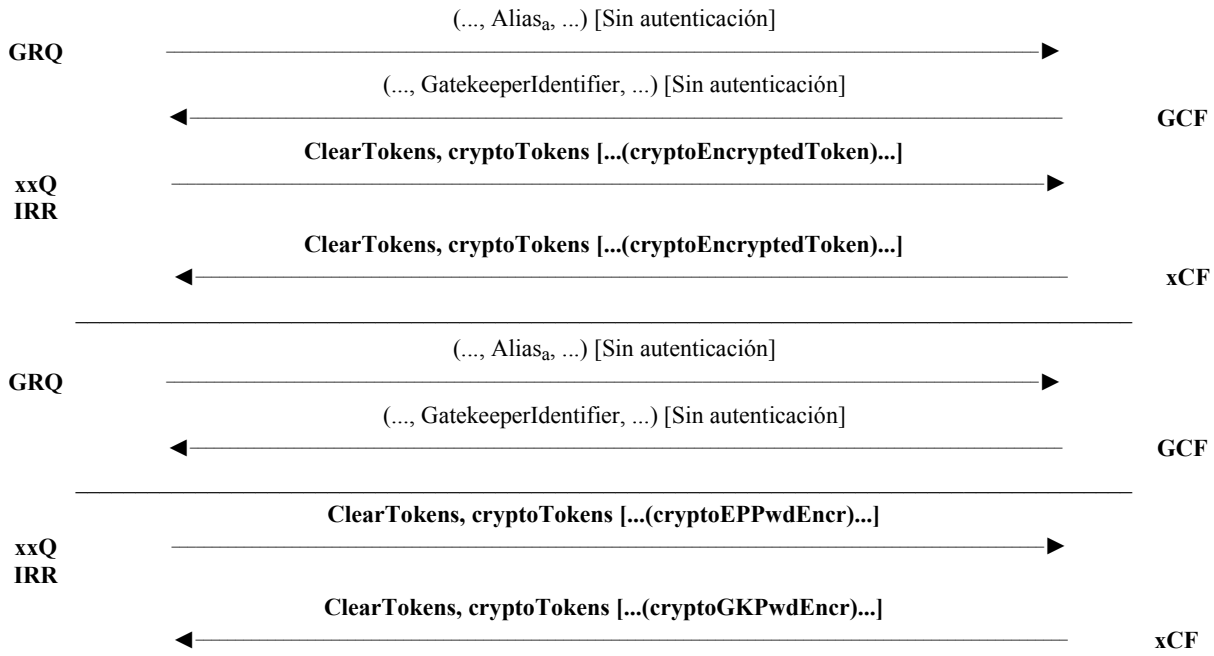


Figura B.2/H.235 – Contraseña con criptación simétrica

B.4.3.2 Contraseña con generación numérica

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**.

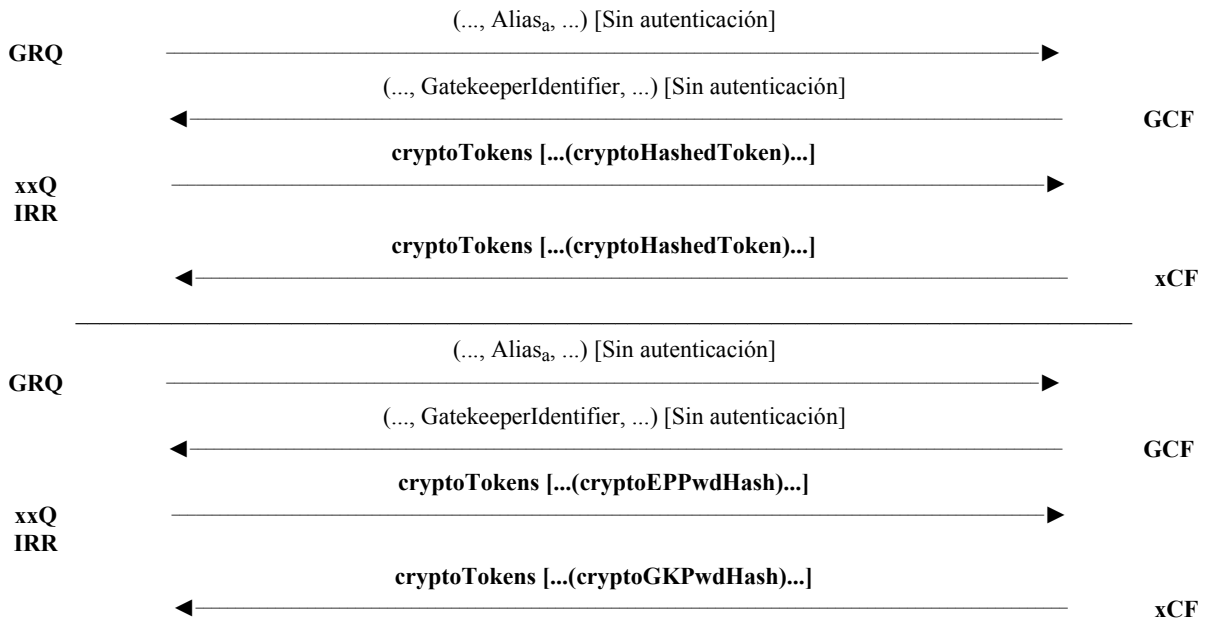


Figura B.3/H.235 – Contraseña con generación numérica

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**.

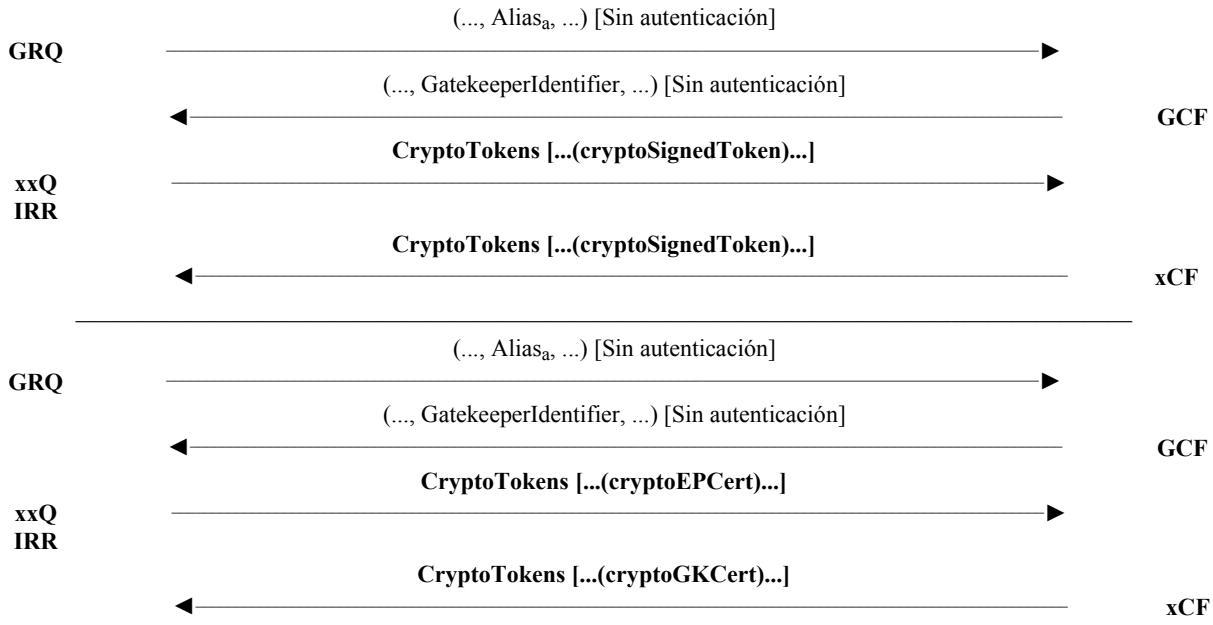


Figura B.4/H.235 – Certificado con firmas

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 pasarelas de seguridad (servidores intermedios/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.

B.6 Gestión de clave en el canal RAS

En algunos casos, conviene distribuir las claves de sesión (RAS) desde un controlador de acceso hacia uno o varios puntos extremos bajo su control, o desde un punto extremo hacia otro. En el mecanismo propuesto se supone que el controlador de acceso y el punto extremo comparten una clave secreta fuerte o conocen cada uno la clave pública del otro. Esto ocurre, por ejemplo, cuando un controlador de acceso de encaminamiento emite una clave de sesión a un punto extremo en un mensaje RAS, por ejemplo **RCF** o **ACF**, para que se utilice en la criptación de un canal de señalización encaminado por controlador de acceso. Otro ejemplo ocurre cuando el controlador de acceso emite una clave de sesión para ser utilizada en la criptación que viene después de las comunicaciones RAS (por ejemplo **RRQ** o **ARQ**).

Este mecanismo es similar al que se utiliza para la distribución de claves de sesión de medios. Es posible utilizarlo en algunos casos para evitar la tara de la negociación de clave.

Para el transporte de clave, el campo **h235Key** facultativo del **ClearToken** debería utilizarse en H.235v3. La flexibilidad del elemento **H235Key** permitirá el transporte de material clave de criptación utilizando:

- un canal seguro (la opción **secureChannel**) suponiendo que el RAS o un canal de señalización de llamada ha sido seguro por otros medios (por ejemplo IPSEC/SSL);
- un secreto de criptación compartido en un canal despejado (la opción **sharedSecret**), o de la misma manera (aunque preferiblemente) la opción **secureSharedSecret**;
- una criptación y un certificado de clave pública en un canal despejado (la opción **certProtectedKey**).

La utilización de la clave de sesión RAS intercambiada y su aplicación a RAS, mensajes de señalización de llamada y/o canal transporte queda en estudio.

B.7 Función seudoaleatoria (PRF, *pseudo-random function*)

En esta cláusula se define una función seudoaleatoria para calcular claves dinámicas a partir de material de clave estática y un valor aleatorio.

NOTA – Esta PRF es idéntica a la PRF MIKEY (véase [MIKEY]/RFC xxxx).

El método de cálculo de clave tiene los siguientes parámetros de entrada:

- *inkey*: la clave de entrada para la función de cálculo.
- *inkey_len*: la longitud en bits de la clave de entrada.
- *label*: una etiqueta específica, que depende del tipo de clave que se debe calcular y del valor aleatorio **challenge**.
- *outkey_len*: longitud deseada en bits de la clave de salida.

La función seudoaleatoria tiene el siguiente resultado:

- *outkey*: la clave de salida de longitud deseada.

Sea HMAC (véase [RFC 2104]) la función de autenticación de mensaje basada en SHA1 (véase [ISO/CEI 10118-3]). Como en RFC 2246 se define:

$$P(s, label, m) = \text{HMAC}(s, A_1 \,||\, label) \,||\, \text{HMAC}(s, A_2 \,||\, label) \,||\, \ldots \,||\, \text{HMAC}(s, A_m \,||\, label)$$

donde:

A_0 = label,

A_i = HMAC (s, A_{i-1}).

Mientras que se suele utilizar SHA1 [ISO/CEI 10118-3], se puede también utilizar HMAC con otras funciones generadoras; esto queda en estudio.

A continuación se describe un procedimiento para obtener una función seudoaleatoria, denominada *PRF(inkey, label)*, que se aplica para computar la clave de salida, outkey:

- sea s_n = *inkey_len/512*, aproximada al entero más cercano
- sepárese *inkey* en *n* bloques, *inkey* = s_1 || ... || s_n, donde todos los s_i, salvo probablemente s_n, tienen 512 bits
- sea *m* = *outkey_len / 160*, aproximada al entero más cercano.

Se obtiene entonces la clave de salida, *outkey*, como los *outkey_len* bits más significativos de

PRF(inkey, label) = P(s_1, label, m) XOR P(s_2, label, m) XOR ... XOR P(s_n, label, m).

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 la Rec. UIT-T H.235, los perfiles ETSI y los perfiles IMTC. Los perfiles de seguridad seleccionan las características de seguridad apropiadas a partir de la Rec. 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 (generación numérica codificada). A lo largo del texto se hace referencia a los identificadores de objeto mediante un símbolo (por ejemplo, "A"), véase también la cláusula 5.

Si bien el servicio de integridad de mensajes además siempre proporciona 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 generación numérica) 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 utilizado 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 generadora cifrada 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 número generador 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.

---

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

Este perfil define "fijar el **generalID** en el **ClearToken** al identificador del recipiente". Esto de hecho significa que, para los mensajes RAS destinados al controlador de acceso, este identificador es el identificador del GK; para los mensajes RAS destinados al punto extremo, es el identificador de punto extremo, y para los mensajes de señalización de llamada H.225.0 destinados al controlador de acceso, éste es el identificador de GK, y para los mensajes de señalización de llamada H.225.0 destinados para el punto extremo, es el identificador de punto extremo llamado, véase también la cláusula D.10.

El **sendersID** deberá ser fijado a la cadena de identificación del emisor. Esto quiere decir que, para los mensajes RAS destinados al controlador de acceso, éste es el identificador de punto extremo; para los mensajes RAS destinados al punto extremo, éste es el identificador de controlador de acceso; para los mensajes de señalización de llamada H.225.0 destinados al controlador de acceso, éste es el identificador GK y para los mensajes de señalización de llamada H.225.0 destinados al punto extremo, es el identificador de punto extremo llamado, véase también la cláusula D.10.

Un **block** (**bloque**) se refiere a la unidad básica de bits empaquetados que el descifrador de bloques es capaz de criptar/decriptar 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 la AES es 128 bits.

Este anexo puede aplicar protección de integridad de mensaje que cubra el mensaje completo. Para los RAS H.225.0, la protección de integridad cubre todo el mensaje RAS; para la señalización de llamada, cubre el mensaje completo de señalización de llamada H.225.0, incluidos los encabezamientos Q.931.

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

Este anexo utiliza las siguientes siglas.

AES Algoritmo de criptación avanzado (*advanced encryption algorithm*)

BES Servidor fuera del terminal (*back-end server*)

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 Telecomunicación (*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*)

KS Clave adicional de seguridad en modo EOFB (*salting key in EOFB mode*)

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

MD5 Message Digest 5

NAT Traducción de dirección de red (*network address translation*)

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 de transporte en tiempo real (*real-time transport 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

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. En esta Recomendación, la referencia a un documento, en tanto que autónomo, no le otorga el rango de una Recomendación.

AES [FIPS-197] US National Institute of Standards, "*Advanced Encryption Algorithm (AES)*", Federal Information Processing Standard, (FIPS) Publication 197, November 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

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

DES [FIPS-74] USNational Institute of Standards, "*Guidelines for Implementing and Using the Data Encryption Standard*", Federal Information Processing Standard (FIPS) Publication 74, April 1981, http://www.itl.nist.gov/div897/pubs/fip74.htm.

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

[ISO/CEI 10118-3] ISO/CEI 10118-3:2004, *Information Technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions*.

[H.225.0] Recomendación UIT-T H.225.0 versión 5 (2003), *Protocolos de señalización de llamadas y paquetización de trenes de medios para sistemas de comunicación multimedios por paquetes*.

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

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

[H.245] Recomendación UIT-T H.245, versión 10 (2003), *Protocolo de control para comunicación multimedios*.

[H.323] Recomendación UIT-T H.323, versión 5 (2005), *Sistemas de comunicación multimedios basados en paquetes*.

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

[RFC 2268] RFC 2268 (1998), *A Description of the RC2® Encryption Algorithm*.

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[3] 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.

[3] Salto tiene aquí el sentido de un elemento de red H.235 de confianza (por ejemplo, GK, GW, MCU, servidor intermedio, 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.

- 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 números generadores criptográficos 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 intromisión en el 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.
- Su aplicabilidad en distintos escenarios de despliegue, tales como los grupos cerrados, los entornos escalables y las conferencia multipunto.
- El perfil de seguridad de sólo autenticación se aplica cuando se proporciona alguna seguridad para el paso a través de un NAT/cortafuegos.

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). El perfil facultativo de seguridad de sólo autenticación se muestra como una diagonal sombreada de color azul en la copia electrónica.

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

<table>
<tr><th rowspan="2">Servicios de seguridad</th><th colspan="4">Funciones de llamada</th></tr>
<tr><th>RAS</th><th>H.225.0</th><th>H.245 (nota)</th><th>RTP</th></tr>
<tr><td>Autenticación</td><td>Contraseña HMAC-SHA1-96</td><td>Contraseña HMAC-SHA1-96</td><td>Contraseña HMAC-SHA1-96</td><td></td></tr>
<tr><td>Sólo autenticación</td><td>Contraseña HMAC-SHA1-96</td><td>Contraseña HMAC-SHA1-96</td><td>Contraseña HMAC-SHA1-96</td><td></td></tr>
<tr><td>No repudio</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Integridad</td><td>Contraseña HMAC-SHA1-96</td><td>Contraseña HMAC-SHA1-96</td><td>Contraseña HMAC-SHA1-96</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td>DES de 56 bits | Compatible con RC2 de 56 bits | DES triple de 168 bits | AES de 128 bits</td></tr>
<tr><td>Confidencialidad</td><td></td><td></td><td></td><td>Modo CBC o modo EOFB</td></tr>
<tr><td>Control de acceso</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Gestión de claves</td><td>Asignación de contraseña basada en abono</td><td>Asignación de contraseña basada en abono | Intercambio de claves Diffie-Hellman autenticadas</td><td>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), AES de 128 bits</td><td></td></tr>
<tr><td colspan="5">NOTA – H.245 tunelizado o H.245 insertada en una conexión rápida H.225.0.</td></tr>
</table>

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. La generación numérica de todos los campos en los mensajes RAS y de señalización de llamada 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 AES de 128 bits 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 utilizan 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 AES de 128 bits y la DES triple de 168 bits, o 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 darse fuera del alcance 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, servidores intermedios). El perfil proporciona la autenticación e integridad, o la autenticación solamente, del mensaje RAS y de señalización de llamada H.225.0 y H.245 tunelizado utilizando la generación numérica 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[4]	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 cuatro algoritmos de criptación: los esquemas propuestos consisten en la criptación que utiliza AES, la norma compatible con RC2, la

[4] El perfil de seguridad de sólo autenticación no proporciona integridad de mensaje.

DES o la DES triple basada en el modelo comercial y el requisito de exportabilidad. Además del modo de criptación CBC, las entidades H.323 pueden implementar el modo de criptación de cifrado de trenes EOFB. 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 utilizan 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 AES de 128 bits o la DES triple de 168 bits, o pueden implementar la criptación exportable utilizando la compatible con RC2 de 56 bits.

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

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

<table>
<tr><th rowspan="2">Servicios de seguridad</th><th colspan="4">Funciones de llamada</th></tr>
<tr><th>RAS</th><th></th><th>RAS</th><th>RTP</th></tr>
<tr><td>Autenticación e integridad</td><td></td><td></td><td></td><td></td></tr>
<tr><td>No repudio</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td>DES de 56 bits | Compatible con RC2 de 56 bits | DES triple de 168 bits | AES de 28 bits</td></tr>
<tr><td>Confidencialidad</td><td></td><td></td><td></td><td>Modo CBC o modo EOFB</td></tr>
<tr><td>Control de acceso</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Gestión de claves</td><td></td><td>Intercambio de claves Diffie-Hellman autenticadas</td><td>Gestión de claves de sesión H.235 integrada (distribución de claves, actualización de claves)</td><td></td></tr>
</table>

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 puede asegurar autenticación aplicando correctamente un clave secreta compartida. Ambos servicios de seguridad son proporcionados por el mismo mecanismo de seguridad.
- **Sólo autenticación**: Este servicio, parte opcional del perfil básico de seguridad, soporta solamente la autenticación de campos escogidos, más no proporciona integridad completa de mensajes. El perfil de seguridad sólo autenticación se aplica a mensajes de señalización que atraviesan dispositivos NAT/cortafuego. El usuario puede aseguar autenticación siempre que aplique correctamente una clave de secreto compartido.

Cuando se utilizan las técnicas de clave simétrica, los servicios de seguridad de autenticación/integridad sólo se aplican en un modo 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

El siguiente procedimiento se describe 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, Q.931 y H.245 (véase D.6.3.2).

El procedimiento IA es un mecanismo de sólo autenticación basado en una clave simétrica simple, y que consta de una contraseña compartida entre dos entidades (por ejemplo, un controlador de acceso y un punto extremo H.323). Este procedimiento proporciona autenticación, más no integridad completa de mensaje. El procedimiento de sólo autenticación es opcional y se aplica en los casos en donde los mensajes de señalización H.323 atraviesen NAT/cortafuegos.

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 seguro o controlador de acceso par, responde con un mensaje de rechazo que señala el fallo de seguridad fijando el motivo del rechazo a **securityDenial**, u otros códigos de error de seguridad adecuados, conforme a B.2.2. Dependiendo de la capacidad para reconocer un ataque, y de la manera más adecuada para reaccionar ante él, un controlador de acceso que reciba una **xRQ** asegurada con identificadores de objeto indefinidos (**tokenOID**, **algorithmOID**) puede responder con **xRJ** no seguro y con razón de rechazo puesta a **securityDenial**, o puede simplemente descartar este mensaje. Debería incluirse en un registro cronológico el evento de seguridad encontrado. De otra parte, el punto extremo descartará el mensaje no seguro recibido, se desconectará y podrá tratar de nuevo escogiendo otros OID. De la misma forma, un controlador de acceso que reciba un mensaje SETUP H.225.0 seguro con identificadores de objeto indefinidos (**tokenOID**, **algorithmOID**) puede responder con un RELEASE COMPLETE no seguro y la razón de rechazo puesta a **securityDenied**, o puede simplemente descartar ese mensaje. Como antes, el evento de seguridad encontrado debería ser registrado.

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 números generadores criptográficos e introducidos en los campos generadores de **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:

- Los algoritmos HMAC-SHA1-96 generan un valor hash (96 bits) de 12 bits como el autenticador resultante. 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 o de señalización de llamada H.225.0.
    - **hashedVals**, que contiene el campo **ClearToken** utilizado con los siguientes campos:
      - **tokenOID** puesto a: "T", indicando que se está utilizando, como se muestra a continuación, el **ClearToken** básico para la autenticación del mensaje y protección contra reproducción, así como (facultativamente) para la gestión de clave Diffie-Hellman descrita en D.7.1. Se pueden también usar **ClearTokens** con otros OID en lugar del **ClearToken** básico.
      - **timeStamp** que contiene la indicación de tiempo.
      - **random**, que contiene un número secuencial monotónicamente creciente. Este número permite la elaboración de 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** para **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 deberían enviarse entonces parámetros Diffie-Hellman y tampoco debería haber **dhkey**; los **halfkey**, **modsize** y **generator** pueden fijarse a {'0'B,'0'B,'0'B}.
  - **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 y de señalización de llamada 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 número generador se efectuará sobre el mensaje de señalización de llamada **H.225.0** completo con todos los campos, de conformidad con el procedimiento descrito en 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/CEI 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 destinatario es capaz de detectar la utilización del procedimiento I mediante la evaluación del **tokenOID** dentro del **EncodedGeneralToken** generado numéricamente (detectando la presencia de "AB").

D.6.3.3 Cálculo del número generador basado en contraseñas

Tanto el emisor como el receptor de un mensaje de autenticación/integridad calculan el número generador con clave sobre todos los campos del mensaje codificado en ASN.1 (utilizando el OID "A"). Para el caso del perfil de sólo autenticación, tanto el remitente como el recipiente calculan un número generador con clave en todo el ClearToken codificado mediante ASN.1 (utilizando el OID "B").

D.6.3.3.1 HMAC-SHA1-96

HMAC-SHA1-96 es el número generador 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 número generador se utilizaran como resultado. RFC 2104 describe el procedimiento con la clave secreta *K* fijada al secreto (= SHA1-contraseña generada numéricamente) 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 número generador 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; para RAS esto incluirá el mensaje completo RAS H.225.0; en el caso de la señalización de llamada incluirá el mensaje completo de señalización de llamada H.225.0.
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 número generador 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 mensaje codificado por el número generador calculado.

El recipiente recibe el mensaje y procede como sigue:

1) Decodifica el mensaje en ASN.1.
2) Extrae el número generador recibido y lo guarda en un RV variable local.
3) Busca y localiza el número generador RV en el mensaje codificado recibido.

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

---

[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.

4) Sobrescribe el esquema de bits en el mensaje codificado con los 96 bits cero.

5) Calcula el número generador criptográfico en el mensaje codificado utilizando HMAC-SHA1-96 (véase D.6.3.3.1).

6) Compara RV con el número generador calculado. El mensaje sólo se considera incorrupto si los dos números generadores 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 números generadores correcta, la autenticación ha fallado y el mensaje ha sido alterado (accidental o deliberadamente) durante el tránsito.

D.6.3.3.3 Sólo autenticación (procedimiento IA)

Se puede elegir en cada terminal si se implementa la sola autenticación (utilizando OID "B", véase la cláusula E.18). En este caso, se calcula el autenticador en un subconjunto (**ClearToken** dentro de **CryptoToken**) del mensaje RAS/H.225.0. La sola autenticación puede ser útil al atravesar NAT/cortafuegos que cambien direcciones/puertos IP dentro de cabidas útiles H.323.

Puesto que la autenticación cubre solamente una porción muy limitada del mensaje, este procedimiento de sólo autenticación no proporciona integridad de mensaje como sí lo hace el procedimiento I. Es decir, la sola autenticación es menos segura.

En el procedimiento de sólo autenticación se utilizarán los siguientes campos en los mensajes protegidos:

- El campo **CryptoH323Token** en cada mensaje RAS/H.225.0 contendrá los siguientes campos:
  - **nestedCryptoToken**, que incluye un **CryptoToken** que a su vez contiene el **cryptoHashedToken**, que tiene los siguientes campos:
    - **tokenOID**, fijado a:
      - "B" (véase E.18), lo que indica que el cálculo de sólo autenticación incluye todos los campos en el **ClearToken**.
    - **hashedVals**, que contienen el campo **ClearToken** utilizado con los siguientes campos:
      - **tokenOID**, fijado a:
        - "T" (como en el ejemplo del ClearToken básico para el resto de los contenidos de ClearToken) o cualquier otro OID adecuado para otros propósitos.
      - **timeStamp**, que contiene la indicación de tiempo;
      - **random**, que contiene un número secuencial monótonamente creciente. Este número permite la elaboración de dos mensajes que tengan la misma indicación de tiempo (dentro de la resolución del reloj);
      - **generalID**, que contiene el identificador del recipiente (sólo en el caso de mensajes unidifusión);
      - **sendersID**, que contiene el identificador del emisor;
      - **dhkey**, que se utiliza para hacer pasar los parámetros Diffie-Hellman, como se especifica en la Rec. UIT-T H.235 durante **Setup** a **Connect**.
        - **halfkey**, que contiene la clave pública aleatoria de una parte;
        - **modsize**, que contiene el DH primo (véase el cuadro D.4);
        - **generator**, que contiene el grupo DH (véase el cuadro D.4).

NOTA 1 – Cuando se use el perfil de seguridad básico sin el perfil de seguridad de criptación de voz no deberían enviarse parámetros Diffie-Hellman y no debería haber **dhkey**; **halfkey**, **modsize** y **generator** se pueden fijar a {'0'B,'0'B,'0'B}.

– **token**, que contiene **HASHED** con los campos:
  • **algorithmOID** fijado a "U", que indica la utilización de HMAC-SHA1-96;
  • **params** fijado a NULL;
  • **hash**, que contiene el autenticador calculado mediante HMAC-SHA1-96. El autenticador se calculará para:
    – todos los campos de **ClearToken**, si **tokenOID** en el **CryptoHashedToken** se ha fijado a "B" (lo que indica que se utiliza la sola autenticación).

• Al final de cada tramo de canal de terminación se verifica el autenticador (EP1-GK1, GK1-GK2, GK2-EP2, EP1-GK2, GK1-EP2 o EP1-EP2 como puede ser el caso), y se recalcula antes de enviar el mensaje al tramo subsiguiente.

NOTA 2 – El autenticador se calcula en el **ClearToken**.

NOTA 3 – Se utilizará el método de relleno con la norma SHA1 [ISO/CEI 10118-3].

NOTA 4 – Para evitar ataques de reproducción, se hace énfasis en la recomendación de que las implementaciones garanticen que la contraseña (clave) se cambie antes del incremento (o cuando se complete el ciclo) del número secuencial monótonamente creciente.

NOTA 5 – El recipiente debe poder detectar la utilización del procedimiento IA evaluando el **OID** "B" dentro del **tokenOID**.

Se calculará el autenticador en el **ClearToken** dentro del **CryptoH323Token** (es decir **ClearToken**) del **token** del **cryptoHashedToken**. Se calculará el número generador criptográfico en la cadena de bits codificados ASN.1 de **ClearToken**.

Los puntos extremos de las versiones 1 y 2 de la Rec. UIT-T H.235 pueden utilizar sólo autenticación, en cuyo caso se utilizarán los OID correspondientes para "B". Los puntos extremos de la versión 1 han de seguir el procedimiento descrito en D.6.6.

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.

NOTA 1 – 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.

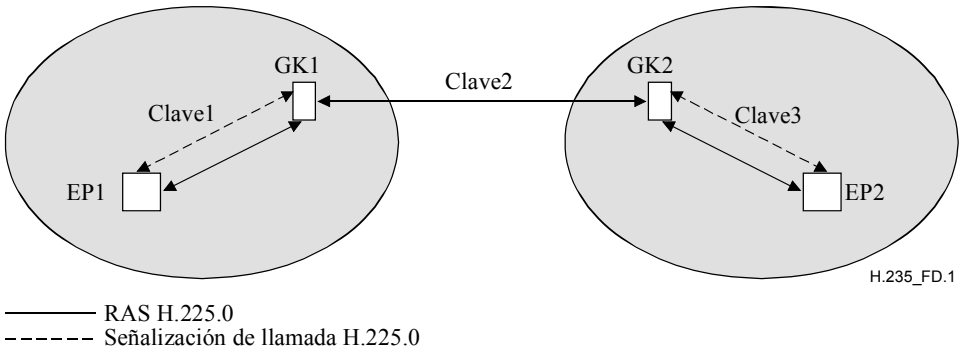


Figura D.1/H.235 – Ilustración de la utilización del procedimiento I en un escenario GK-GK con ambos EP en zonas de encaminamiento por controlador de acceso

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.

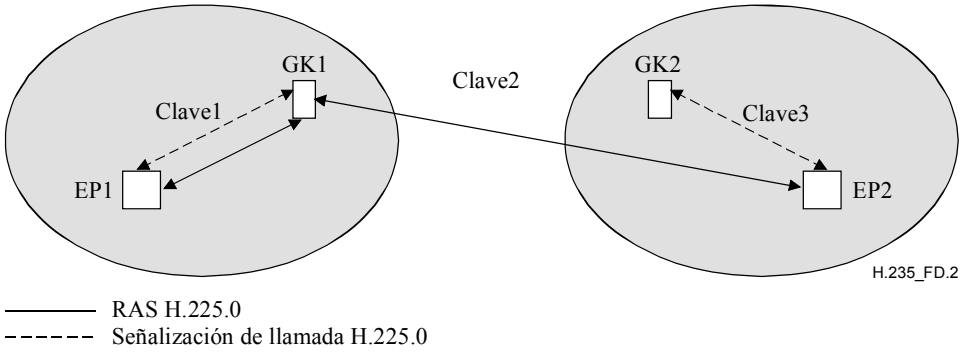


Figura D.2/H.235 – Ilustración de la utilización del procedimiento I en un escenario mixto con EP1 en una zona de encaminamiento por controlador de acceso y EP2 en una zona de encaminamiento directo

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.

NOTA 2 – 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.

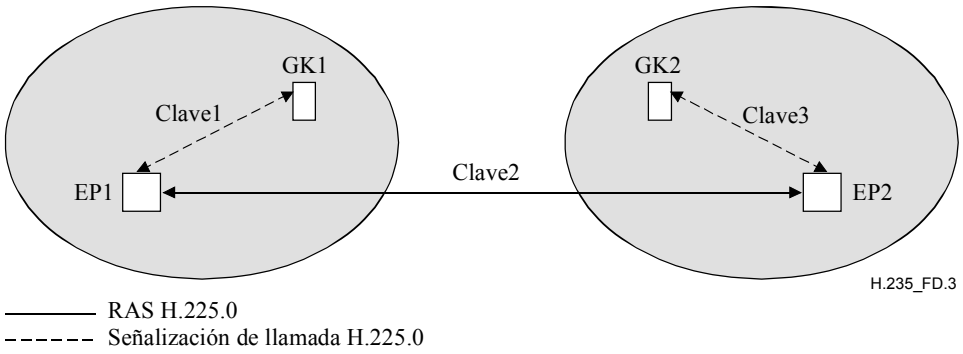


Figura D.3/H.235 – Ilustración de la utilización del procedimiento I en un escenario con ambos EP en zonas que utilizan un modelo de GK con encaminamiento directo

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, respectivamente. A partir de estas contraseñas se generan tres claves de 20 bytes – *Key1 (clave1), Key2 (clave2)* y *Key3 (clave3)* – basándose en el procedimiento descrito en 10.3.5. 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 3 – Los escenarios que se muestran en las figuras 1 a 3 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 a todos los campos en el mensaje **ARQ** se les ha aplicado la generación numérica. 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 20 bytes *Key1*. El autenticador es calculado sobre el mensaje RAS entero.

EP1 incluye el autenticador calculado dentro de **hash** en el campo **token** 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 a todos los campos en el mensaje de señalización H.225.0 se les ha aplicado la generación numérica. 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 20 bytes, *Key1*. El autenticador es calculado de conformidad con el método de generación numérica elegido (A) tomando en consideración el mensaje de señalización de llamada 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 **random**.
- 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 de señalización de llamada H.225.0 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;
- **h245Tunnelling** 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 generador calculado sobre todos los campos del mensaje de señalización de llamada H.225.0.

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;
- **h245Tunnelling** 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. La adición de seguridad al modelo con encaminamiento directo queda 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 la Rec. UIT-T H.235 versión 2 [Rec. UIT-T H.235 (2000)], se pueden también aplicar a la Rec. UIT-T H.235 versión 1 [Rec. UIT-T 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 la Rec. UIT-T H.235 versión 1 [Rec. UIT-T 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 AES, compatibles con RC2, DES y DES triple.

NOTA – 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.3.1.
- Relleno, si es necesario, de acuerdo con la descripción de B.3.2.

La cabida útil audio será criptada mediante el algoritmo de criptación negociado ("X", "Y", "Z3" o "Z") 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. Se puede criptar la cabida útil de audio utilizando el algoritmo de criptación negociado ("X1", "Y1", "Z1" o "Z2") con un modo de cifrado de trenes (EOFB).

D.7.1 Gestión de claves

Los puntos extremos que se ajusten a este anexo deberían utilizar el procedimiento de conexión rápida conforme a 8.6.1. Si no se aplica el arranque rápido, se utilizará entonces la tunelización H.245 para asegurar los mensajes de control de llamada H.245, según este anexo. El procedimiento de arranque rápido permite el establecimiento de uno o dos canales lógicos unidireccionales. El procedimiento de arranque rápido tiene en cuenta la negociación de las capacidades de seguridad, para la distribución de un secreto común compartido (secreto DH compartido) que funciona como clave maestra, y para la distribución segura de una clave de criptación.

En el cuadro D.4 se proporcionan los OID atribuidos a los diversos algoritmos de criptación, y se muestra su relación con los OID atribuidos al grupo Diffie-Hellman. Se identifican tres grupos DH mediante un OID:

- "DHdummy": Se debería aplicar un ejemplar de este grupo DH siempre que se aspire a tener una seguridad exportable (512 bit), o se utilice cualquier grupo DH no estándar.

  NOTA 1 – No se define un grupo particular DH; el OID se refiere a cualquier grupo DH no estándar.

- Se utilizará un ejemplar de un grupo DH de 512 bits para generar una clave maestra para la distribución de clave(s) de sesión compatible RC2 ("X") o los algoritmos de criptación DES de 56 bits ("Y").
- "DH1024": Este grupo DH se ha de aplicar siempre que se pretenda conseguir una alta seguridad (1024 bits). El OID se refiere a un grupo DH normalizado y fijo. Este grupo DH se utilizará para generar una clave maestra para la distribución de clave(s) de sesión para los algoritmos de criptación DES triple ("Z").
- "DH1536": Este grupo DH se ofrece como opción para los puntos extremos de la versión 3 que posean requisitos muy exigentes de seguridad, superiores a los del grupo DH de 1024 bits. El OID se refiere a un grupo DH fijo. Este grupo se utilizará para generar una clave maestra para la distribución de clave(s) de sesión para DES triple ("Z", "Z1") o para algoritmos de criptación AES-128 ("Z2", "Z3").

Se recomienda aplicar los grupos DH de 1024 bits (o en su lugar los de 1536) definidos, a menos que por otros requisitos de seguridad se prefiera utilizar otros parámetros Diffie-Hellman. Además, se recomienda utilizar los OID definidos que identifican los grupos DH, como se explica en 8.8. No obstante, las implementaciones deberían estar preparadas para obtener literalmente los parámetros de grupo DH sin necesidad de una indicación explícita de OID. En este caso, deberían afirmar que el grupo DH correcto está siendo transportado conforme al cuadro D.4.

Los puntos extremos pueden utilizar parámetros de grupo DH no estándar. La utilización del OID "DHdummy" indica la presencia de dichos grupos DH no estándar. Es potestad del destinatario de la llamada aceptar o no dichos grupos DH.

NOTA 2 – La selección de uno de dichos grupos DH no implica que no sea necesario negociar el algoritmo real de criptación de medios. Esto se debe lograr mediante el procedimiento de negociación de capacidades de terminal H.245.

NOTA 3 – Durante el establecimiento de la conexión (SETUP-a-CONNECT) no se utilizarán los OID de algoritmo de criptación para indicar un ejemplar Diffie-Hellman.

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

Algoritmo de criptación OID	DH-OID	Descripción del grupo D-H
"X", "X1" (compatible con RC2), "Y", "Y1" (DES)	"DHdummy"	Mod-P, cualquier primo de 512 bits adecuado
"Z", "Z1" (DES triple), "Z2", "Z3" (AES)	"DH1024"	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 (nota) = 2
"Z", "Z1" (DES triple), "Z2", "Z3" (AES)	"DH1536"	Mod-P, primo de 1536 bits Primo = $2^{1536} - 2^{1472} - 1 + 2^{64} \times \{ [2^{1406}\ \text{pi}] + 741804 \}$ = (241031242692103258855207602219756607485695054850245994265411 694195810883168261222889009385826134161467322714147790401219 650364895705058263194273070680500922306273474534107340669624 601458936165977404102716924945320037872943417032584377865919 814376319377685986952408894019557734611984354530154704374720 774996976375008430892633929555996888245787241299381012913029 459299994792636526405928464720973038494721168143446471443848 852094012745984428885933652689632091963391 9)$_{10}$ Generador (nota) = 2
NOTA – El generador se utiliza para generar el testigo DH.		

D.7.2 Actualización de claves y sincronización

Para cifrados de bloque de 64 bits, 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 11.1). Cuando se trate de cifrados de bloque de 128 bits, la tasa de renovación de claves *deberá* ser tal que no se cifren más de 2^{64} bloques con la misma clave. Las implementaciones *deberían* renovar las claves antes de que se hayan criptado 2^{62} bloques utilizando la misma clave (véase 11.1). 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** o **encryptionUpdateCommand** del mensaje **miscellaneousCommand**. Por otra parte, el terminal subordinado puede solicitar una nueva clave de sesión al terminal director utilizando la **encryptionUpdateRequest** del mensaje **miscellaneousCommand**, véase también B.2.6.

El mensaje **MiscellaneousCommand** contiene la **encryptionUpdate** y **encryptionUpdateCommand** 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", "X1" para el compatible con RC2 de 56 bits, puesto a "Y", "Y1" para el DES de 56 bits o puesto a "Z", "Z1" para el DES triple de 168 bits o puesto a "Z3" para AES de 128 bits.

  NOTA 1 – El algoritmo de criptación de clave de sesión es el algoritmo de criptación de medios negociado.
- **paramS**: puesto al valor inicial. Para cifrados de tren de bloques de 64 bits, **iv8** contiene un esquema de bits de bloques de 64 bits aleatorios que genera el iniciador. Para cifrados de tren de bloques de 128 bits, **iv16** contiene un esquema de bits de bloques de 128 bits aleatorios que genera el iniciador. Este campo no se usará en el modo CBC y se fijará a NULO (NULL), lo que indica que se ha de poner a 0 la CBC-IV para la criptación de clave de sesión; se utilizará solamente para el transporte del IV en el modo EOFB.
- **encryptedData**: puesto al resultado del **KeySyncMaterial** criptado.

Como parte del **KeySyncMaterial**:

- **generalID**: identificador de la fuente que distribuye la clave.

  NOTA 2 – En esta Recomendación se supone que cada punto extremo se ha registrado con un controlador de acceso y ha obtenido un identificador de punto extremo que puede ser transportado en **generalID**. En esta Recomendación no se soportan los casos en que no haya controladores de acceso; esto queda en estudio.
- **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 y para AES es una clave de 128 bits. El terminal director deberá generar una nueva clave de sesión que cumpla al menos los siguientes criterios de seguridad: no es 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 3 – 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.

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 decriptación (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.

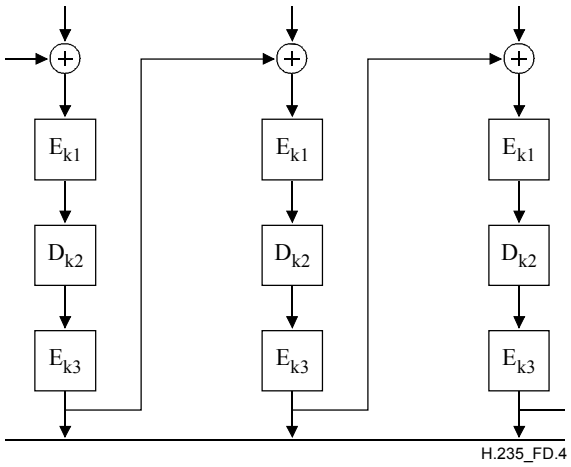


Figura D.4/H.235 – Criptación DES triple en modo CBC exterior

D.7.4 Algoritmo DES que funciona en modo EOFB

Se puede criptar la voz utilizando el algoritmo DES que funciona en el modo de encadenamiento de bloque cifrado de tren EOFB. El modo EOFB permite aprovechar los paralelismos entre las implementaciones. Si se funciona en dicho modo, se recomienda tanto por motivos de calidad de funcionamiento como de seguridad, retroalimentar el bloque criptado completo (es decir, todos los 64 bits para DES, por ejemplo con $n = j = 64$). No obstante, puesto que este modo no proporciona el encadrenamiento entre los bloques y los bits, puede ser susceptible a ataques particulares dependiendo de las propiedades estadísticas de los datos de texto básico de entrada. Es decir, se debería efectuar una actualización de clave (véase D.7.2) regularmente y, en todo caso, antes de que regrese el valor inicial. En B.3.1.2 se describe el cálculo del valor inicial.

D.7.5 DES triple en el modo EOFB exterior

En este perfil de seguridad se puede utilizar la DES triple de 168 bits en modo EOFB exterior, como se muestra en la figura D.5. En la figura, cada k_i representa una clave de 56 bits. Hay que utilizar una clave de 56 bits diferente dentro de cada bloque de criptación (E, *encryption*) y decriptación (D, *decryption*). Aparentemente, ninguna de las 64 claves débiles para DES debilita la DES triple. Sin embargo, las implementaciones conformes a este perfil deberían rechazar la clave cuando está implicada una clave DES débil [RFC 2405].

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

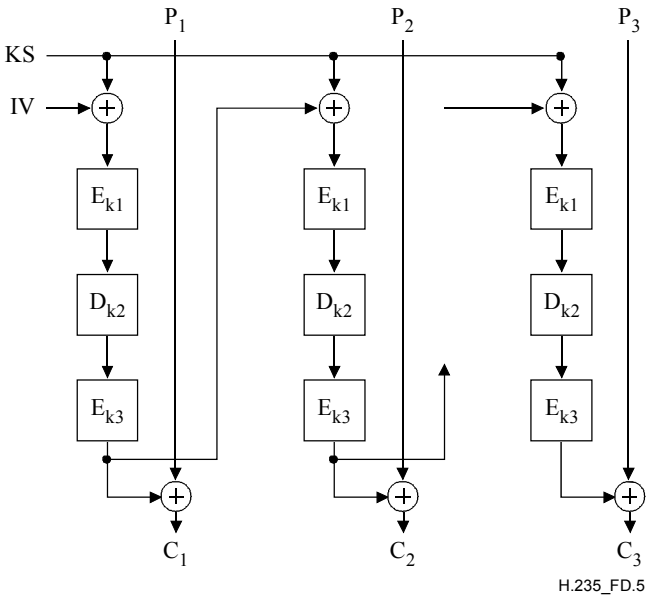


Figura D.5/H.235 – Criptación DES triple en modo EOFB exterior

D.8 Interceptación legal

Queda en estudio (véase [LI]).

D.9 Lista de mensajes de señalización seguros

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, estado UUIE, petición de estado UUIE, acuse de establecimiento UUIE, notificación UUIE	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.235v1] ni para H.235v2 [H.235v2].

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 y de señalización de llamada H.225.0 (autenticación e integridad).
"E"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 9} {itu-t (0) recommendation (0) h (8) 235 version (0) 2 9}	ClearToken de extremo a extremo que transporta sendersID para la verificación en el lado del recipiente.
"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 los procedimientos I y en el IA como el ClearToken básico para la autenticación de mensaje y protección contra los ataques de reproducción y, como una opción, también para la gestión de clave Diffie-Hellman, como se describe en D.7.1.
"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.
"DHdummy"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 40} {itu-t (0) recommendation (0) h (8) 235 version (0) 3 40}	Se proporciona explícitamente el grupo DH no estándar.
"DH1024"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 43} {itu-t (0) recommendation (0) h (8) 235 version (0) 3 43}	Grupo DH de 1024 bits
"DH1536"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 44}	Grupo DH de 1536 bits
"X"	{iso(1) member-body(2) us(840) rsadsi(113549) encryptionalgorithm(3) 2}	Criptación vocal utilizando compatible con RC2 (56 bits) o compatible con RC2 en modo CBC y grupo DH de 512 bits.
"X1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 27}	Criptación vocal que utiliza el RC2-compatible (56 bits) o RC2-compatible en modo EOFB 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 bits) en modo CBC y grupo DH de 512 bits.
"Y1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 28}	Criptación vocal que utiliza el DES (56 bits) en modo EOFB y grupo DH de 512 bits con retroalimentación de 64 bits.
"Z1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 29}	Criptación vocal que utiliza el DES triple (168 bits) en el modo EOFB exterior y grupo DH de 1024-bits con retroalimentación de 64 bits.
"Z2"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 30}	Criptación vocal que utiliza AES (128 bits) en el modo EOFB y grupo DH de 1024 bits.
"Z3"	{joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) 3 nistAlgorithm(4) aes(1) cbc(2)}	Criptación vocal que utiliza AES (128 bits) en el modo CBC y grupo DH de 1024 bits.
"Z"	{iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) desEDE(17)}	Criptación vocal utilizando DES triple (168 bits) 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 seguridad de firmas

E.1 Visión general

Este anexo describe un perfil de firmas, el cual utiliza 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 servidores intermedios 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[7] 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.

[7] "Salto" tiene aquí el sentido de un elemento de red H.235 de confianza (por ejemplo, GK, GW, MCU, servidor intermedio, 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.

- 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 servidores intermedios.

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.
- 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 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 utiliza 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 servidor intermedio 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: Los depósitos de certificados (por ejemplo, un directorio X.500) mantienen los certificados de usuario y las listas de revocación de certificados (CRL, *certificate revocation lists*). 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: El servidor intermedio (sustituto) es una entidad H.323 similar a un controlador de acceso. El servidor intermedio 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 servidor intermedio 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 número generador y un identificador de número generador.

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 la generación numérica 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 III para la autenticación verdadera de extremo a extremo.

Conforme a este anexo, se puede aplicar protección de integridad de mensaje al mensaje completo. Para los RAS H.225.0, la protección de integridad cubre el mensaje completo RAS; para la señalización de llamada, el mensaje completo de señalización de llamada H.225.0 incluyendo los encabezamientos Q.931.

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[8].

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 (azul 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.

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

Servicios de seguridad	Funciones de llamada			
	RAS	H.225.0	H.245 (nota)	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		
NOTA – 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, servidores intermedios 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.

[8] 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.

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 servidores intermedios intermedios de seguridad que verifican la autenticidad del mensaje (autenticación del origen de los datos) cuando se reenvía[9] 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 3280).

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.

---

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

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 la cláusula 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** u otro error de seguridad adecuado, conforme a B.2.2. Dependiendo de la capacidad para reconocer un ataque, y de la manera más adecuada para reaccionar ante él, un controlador de acceso que recibe un **xRQ** seguro con identificadores de objetos no definidos (**tokenOID**, **algorithmOID**) debería responder con un **xRJ** no seguro, o simplemente descartar ese mensaje. El evento de seguridad encontrado debería ser registrado. De otra parte, el punto extremo descartará el mensaje no seguro recibido, se desconectará y tratará de nuevo escogiendo diferentes OID. De igual manera, un controlador de acceso que recibe un mensaje SETUP H.225.0 seguro con identificadores de objetos no definidos (**tokenOID**, **algorithmOID**) debería responder con un COMPLETE RELEASE no seguro y una razón puesta a recuperar **securityDenied**, o simplemente descartar dicho mensaje. Asimismo, se debería registrar el evento encontrado.

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 la cláusula 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**.

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 o de señalización de llamada H.225.0 (véase la cláusula E.9);
      - "B", que indica que el cálculo de la autenticación/integridad incluye solamente un subconjunto de campos (véase la cláusula 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** a **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, se enviarán los parámetros Diffie-Hellman y no debería haber **dhkey**; **halfkey**, **modsize** y **generator** pueden fijarse a {'0'B,'0'B,'0'B}.

  certificate, que contiene el certificado digital del emisor donde el tipo indica el tipo de certificado ("V" para los certificados MD5-RSA o "W" para los certificados SHA1-RSA) y **certificate** transporta el certificado real (véase la cláusula 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 campots (si tokenOID es "A", véase la cláusula E.9) o en determinados campos críticos (si tokenOID es "B", véase la cláusula E.8) del mensaje RAS y mensajes o señalización llamada 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 de señalización de llamada H.225.0 completo con todos los campos, de conformidad con el procedimiento descrito en la cláusula 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 la cláusula 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 la asociación del 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 la cláusula E.12).

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

En la figura E.1 se muestra un escenario con servidores intermedios 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 **CryptoTokens** de extremo a extremo y deben reenviar el **CryptoToken** en tránsito.

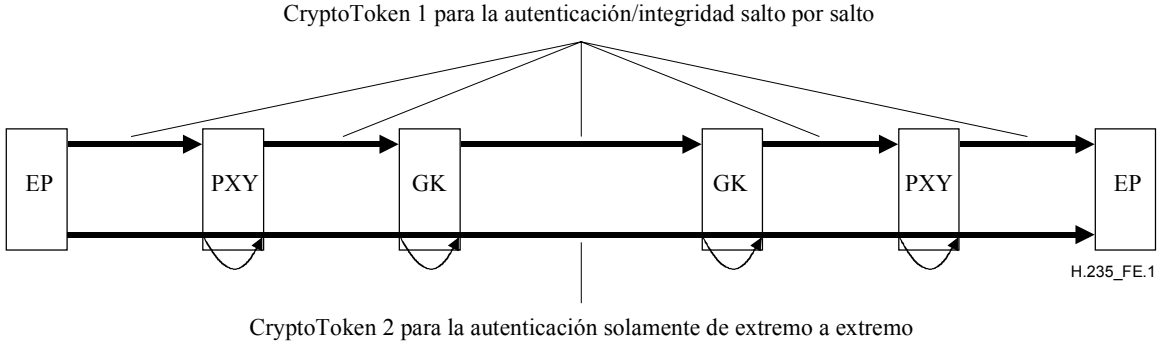


Figura E.1/H.235 – Utilización simultánea de la seguridad salto por salto y la autenticación de extremo a extremo

NOTA 1 – El servidor intermedio 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 de GK.

NOTA 2 – Dependiendo de cual sea el **tokenOID** señalado el servidor intermedio será capaz de determinar si el **CryptoToken** recibido esta destinado al servidor intermedio ("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 servidores intermedios 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 la cláusula E.9).
    - "B", que indica que el cálculo de la autenticación incluye solamente un subconjunto de campos (véase la cláusula E.8) del mensaje RAS o de señalización de llamada 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[10] 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.
    - **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 la cláusula E.12).
    - **dhkey**, utilizado para transferir los parámetros Diffie-Hellman especificados en esta Recomendación durante **Setup** a **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 se debería enviar ningún parámetro Diffie-Hellman y no debería haber **dhkey**; **halfkey**, **modsize** y **generator** pueden ser fijados a {'0'B,'0'B,'0'B}.

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

- **Token** que contiene 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 o de señalización de llamada H.225.0.

El servidor intermedio 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 servidor intermedio 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 se utilizan durante los mensajes **Setup** a **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 calcular 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; para RAS, esto incluirá el mensaje completo RAS H.225.0; para la señalización de llamada, el mensaje completo de señalización de llamada H.225.0.
3) Localizará[11] 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 la cláusula 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 la cláusula 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.
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**.

[11] 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.

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 número generador del certificado del firmante. La validación de la clave pública del firmante puede basarse en el procesamiento del trayecto de certificación (RFC 3280). 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; en este caso **certificate** contiene el certificado real y **type** contiene el OID "V" o el OID "W".
- 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).

Siempre que se transporte un certificado digital en un mensaje, la entidad receptora (controlador de acceso o punto extremo) verificará si la identidad del remitente (controlador de acceso o punto extremo) coincide con la identidad presente en el certificado, para evitar ataques intermedios.

En el caso de los mensajes con firma digital enviados desde un controlador de acceso hasta un punto extremo, existen varias posibilidades para que éste verifique la identidad de aquél. A saber:

– Si se dispone del hostname, por ejemplo en el atributo de nombre común del campo subject o del campo subjectAltName en el certificado, el punto extremo puede verificar si este hostname coincide con el identificador del controlador de acceso. De igual manera, el punto extremo puede utilizar el DNS para averiguar la dirección IP correspondiente y compararla con la dirección IP del controlador de acceso que ha sido presentada en el mensaje de respuesta firmado por éste.

– Por ejemplo, se puede construir el identificador de controlador de acceso concatenando la dirección IP (representada como un valor de 4 bytes en el orden de bytes de red) con otra información que identifique al controlador de acceso, truncado al valor de la longitud máxima del campo ID del remitente (senders_ID), que transporta la identidad del controlador de acceso. Asimismo, el punto extremo puede verificar si la dirección IP que pertenece al hostname coincide con la presentada en el encabezamiento de IP de la respuesta del controlador de acceso.

NOTA – Es probable que este método no funcione como se espera cuando se utilicen mecanismos de traducción de dirección de red (NAT, *network address translation*).

– Si no aparece el hostname en el certificado, la dirección IP que debería ser parte de dicho certificado (*iPAddress subjectAltName*), se tomará directamente a fin de efectuar las pruebas antes mencionadas.

Los usuarios deberían estudiar con cuidado el certificado presentado por el controlador de acceso para decidir si satisface sus expectativas. Cuando el punto extremo tenga información externa del tipo de identidad esperada del controlador de acceso, se puede omitir la verificación del hostname. Por ejemplo, puede ocurrir que aunque un punto extremo se esté conectando a un controlador de acceso cuya dirección y hostname sean dinámicos, ya conozca el certificado que éste presentará. En dichos casos, conviene disminuir tanto como se pueda el alcance de los certificados que pueden ser aceptados, a fin de evitar ataques intermedios. En casos especiales, puede ser conveniente que el punto extremo simplemente ignore la identidad del controlador de acceso, aunque esto implique dejar la conexión abierta a ataques activos.

Cuando el hostname no equivalga a la identidad presente en el certificado, los puntos extremos orientados al usuario notificarán a éste (pueden darle la oportunidad de continuar con la conexión en cualquier caso) o terminarán la conexión con un error certificado incorrecto. Los puntos extremos automatizados registrarán el error en un registro auditor (si se dispone de él) adecuado y deberían terminar la conexión (con un error certificado incorrecto).

Si bien los puntos extremos automatizados pueden proporcionar una configuración que inhabilite esta verificación, deberán en todo caso proveer una que la habilite.

De igual manera, se recomienda que el controlador de acceso efectúe una verificación de la de identidad de cualquier mensaje con firma digital enviado desde el punto extremo hasta él. Cómo se efectúa concretamente dicha verificación es asunto local y debería estar sujeto a la implementación de la política de seguridad del controlador de acceso. Por ejemplo, se puede pensar que un nombre de usuario transportado dentro del certificado puede también formar parte del identificador H.323. Más aún, el controlador de acceso puede verificar si dicha información de identidad corresponde con los datos de usuario administrado/configurado localmente, si los hubiere, y puede basar en ello una decisión relacionada con política.

Cuando el controlador de acceso tenga información externa sobre la identidad esperada del punto extremo, se puede omitir la verificación de hostname. Por ejemplo, puede ocurrir que un controlador de acceso se esté conectando a un punto extremo cuyos dirección y hostname sean dinámicos, pero para el que ya conoce el certificado que será presentado. En tales casos, es importante reducir tanto como se pueda el alcance de los certificados aceptables, a fin de evitar ataques intermedios. En casos especiales, puede convenir que el controlador de acceso ignore simplemente la identidad del punto extremo, aunque esto deba implicar que se deja la conexión abierta a ataques activos.

Cuando el hostname no corresponda con la identidad presentada en el certificado, el controlador de acceso registrará el error en un registro cronológico de auditoría adecuado (si lo hubiere) y debería terminar la conexión (con un error certificado incorrecto).

Cuando haya una extensión subjectAltName de tipo dNSName, se la utilizará como identidad. De lo contrario, se utilizará el campo Common Name (más específico) en el campo Subject del certificado. Aunque se acostumbre utilizar el Common Name, no se aconseja y las autoridades de certificación insisten en que debe utilizarse en su lugar el dNSName.

La correspondencia se efectuará conforme a las reglas especificadas en [RFC 3280]. Cuando haya más de una identidad de un tipo determinado en el certificado (por ejemplo, más de un nombre dNSName), se considera aceptable una correspondencia en cualquiera de los elementos del conjunto. Los nombres pueden incluir el carácter comodín (wildcard) * que se supone corresponde a cualquier nombre único de dominio a cualquier componente o fragmento de componente de

nombre único de dominio. Por ejemplo, *.a.com corresponde a foo.a.com, mas no a bar.foo.a.com. De igual manera f*.com corresponde a foo.com, mas no a bar.com.

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, se deberá utilizar X.509v3 (1997). Otros formatos de certificado quedan en estudio.

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 servidor intermedio H.323 separa EP1 de GK1.

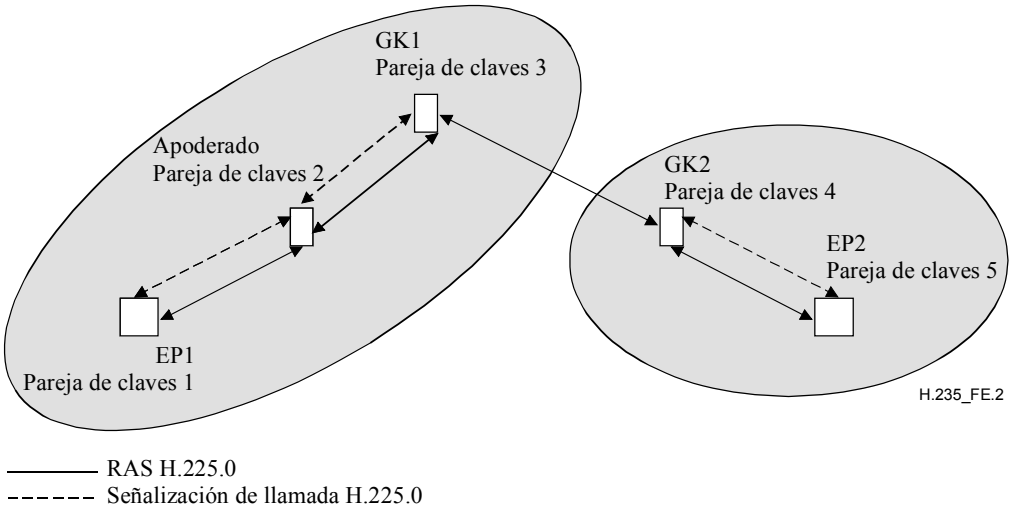


Figura E.2/H.235 – Ilustración de la utilización de claves públicas en un modelo encaminado por GK-GK

El servidor intermedio H.323 actúa doblemente: Por un lado, el servidor intermedio finaliza la autenticación e integridad de cada una de sus ramas. El servidor intermedio 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 servidor intermedio permite que la información de seguridad de extremo a extremo pase sin modificación. El servidor intermedio 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, de señalización de llamada 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 una indicación de tiempo y un número

secuencial y los incluye en los campos **timeStamp** y **random** respectivamente, junto con el alias del servidor intermedio 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 servidor intermedio, 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 servidor intermedio.

Tras la recepción del mensaje **ARQ**, el servidor intermedio 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 servidor intermedio calcula una nueva firma para insertarla (sustituirla) en el mensaje **ARQ** antes de reenviar éste al GK1 como sigue. El servidor intermedio sustituye los campos **timeStamp**, **random**, **sendersID** y **generalID** en el campo **ClearToken** (**toBeSigned**) utilizando valores pertinentes a la rama servidor intermedio-GK1. El campo **timestamp** contiene la indicación de tiempo actual, el campo **random** contiene el siguiente número secuencial monotónicamente creciente para la rama servidor intermedio-GK1, el **sendersID** del servidor intermedio y el campo **generalID** contienen el alias de GK1. El servidor intermedio 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 servidor intermedio 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 servidor intermedio, 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 servidor intermedio 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 servidor intermedio.

Tras la recepción del mensaje **ARQ**, el servidor intermedio 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 servidor intermedio calcula una nueva firma para insertarla (sustituirla) en el mensaje **ARQ** antes de reenviar éste al GK1 como sigue. El servidor intermedio reemplaza los campos **timeStamp**, **random**, **sendersID** y **generalID** en el campo **ClearToken** de **toBeSigned** utilizando valores pertinentes a la rama servidor intermedio-GK1. El campo **timeStamp** contiene la indicación de tiempo actual, el campo **random** contiene el siguiente número secuencial monotónicamente creciente para la rama servidor intermedio-GK1 y el campo **generalID** contiene el alias de GK1. El servidor intermedio 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 servidor intermedio, 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 de señalización de llamada 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 servidor intermedio. 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.
- **h245Tunnelling** 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.
- **h245Tunnelling** puesto a VERDADERO (TRUE).
- **h245Control** contiene la cadena de octetos PDU H.245.

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

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 servidor intermedio 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 servidor intermedio-GK1, 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 de señalización de llamada H.225.0, se calcula una nueva firma para el mensaje H.225.0 antes de su transmisión desde el servidor intermedio al GK1. La firma que fue enviada desde el EP1 al servidor intermedio y que no estaba destinada a este último es transferida del servidor intermedio 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 la Rec. UIT-T H.235

Si bien estos perfiles de seguridad se han desarrollado pensando en la Rec. UIT-T H.235 versión 2 [H.235v2], se pueden también aplicar a la Rec. UIT-T H.235 versión 1 [H.235v1] 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 cláusula E.18).

Implementaciones de la Rec. UIT-T H.235 versión 1 [H.235v1]:

- 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 Estado UUIE Indagación de Estado UUIE Acuse de Establecimiento UUIE Notificación UUIE	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 [WEBOID]). Hay identificadores de objeto para H.235v1 [H.235v1] y para H.235v2 [H.235v2].

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 o de señalización de llamada H.225.0 (autenticación e integridad).
"B"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 2} {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. Utilizado en el procedimiento IA del anexo D para el CryptoToken-tokenOID que indica que el número generador incluye un subconjunto de campos en el mensaje RAS/H.225.0 (ClearToken) para terminales con sólo autenticación y 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 los procedimientos II o III 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 los procedimientos II o III como OID de algoritmo indicando el empleo de la firma digital SHA1 RSA.

Anexo F

Perfil de seguridad híbrido

Resumen

El propósito de este anexo es describir un perfil de seguridad híbrido basado en PKI eficiente y escalable, para la versión 2 de la Rec. UIT-T H.235. El perfil de seguridad híbrido contiene ventajas del perfil de seguridad de los anexos D y E, desplegando firmas digitales del anexo E, y desplegando el perfil de seguridad básico simple del anexo D.

F.1 Visión general

En este anexo se describe un perfil de seguridad híbrido basado en una infraestructura de clave pública (PKI, *public key infrastructure*), eficiente y escalable, que despliega firmas digitales del anexo E y que despliega el perfil de seguridad básica del anexo D. El presente anexo se sugiere como una opción. Las entidades de seguridad H.323 (terminales, controladores de acceso, pasarelas, MCU, etc.) pueden implementar este perfil de seguridad híbrido para mejorar la seguridad o cuando sea necesario.

La noción de "híbrido" en este texto significa que los procedimientos de seguridad del perfil de firmas en el anexo E se aplican realmente en un sentido ligero y las firmas digitales son aún conformes con los procedimientos RSA. Sin embargo, las firmas digitales se despliegan sólo cuando ello es absolutamente necesario; de lo contrario, se utilizan técnicas de seguridad simétrica sumamente eficientes del perfil de seguridad básico descrito en el anexo D.

El perfil de seguridad híbrido es aplicable a la telefonía IP "mundial" escalable. Cuando se aplica estrictamente este perfil de seguridad supera las limitaciones del perfil de seguridad básico simple descrito en el anexo D. Además, cuando se aplica estrictamente este perfil de seguridad resuelve ciertos inconvenientes del anexo E tales como la necesidad de mayor anchura de banda y de una mejor calidad para el procesamiento. Por ejemplo, el perfil de seguridad híbrido no depende de la administración (estática) de los secretos compartidos mutuos de los saltos en diferentes dominios. Así, los usuarios pueden elegir más fácilmente su proveedor VoIP. Por tanto, este perfil de seguridad soporta además cierto tipo de movilidad del usuario. Aplica criptografía asimétrica con firmas y certificados solamente cuando es necesario y en otro caso utiliza técnicas simétricas más simples y eficientes. Proporciona tunelización de los mensajes H.245 para la integridad de los mismos y también implementa algunas disposiciones para el no repudio de mensajes.

El perfil de seguridad híbrido determina el modelo con encaminamiento por GK y se basa en las técnicas de tunelización H.245. Se encuentra en estudio el soporte para los modelos con encaminamiento no efectuado por GK.

Las prestaciones ofrecidas por este perfil incluyen:

Para los mensajes RAS, H.225.0 y H.245:

- La autenticación de usuario a una entidad deseada cualquiera que sea el número de saltos[12] del nivel de aplicación que atraviesa el mensaje.
- La integridad de todas o las porciones críticas (campos) de los mensajes que llegan a una entidad cualquiera que sea el número de saltos del nivel de aplicación atravesados por el mensaje. La integridad del propio mensaje obtenida mediante un número aleatorio generado de forma fuerte es también facultativa.
- La autenticación, integridad y (algún) no repudio del mensaje salto por salto en el nivel de aplicación proporcionan estos servicios de seguridad para el mensaje completo.
- Utilizando la infraestructura disponible de claves públicas, los usuarios pueden elegir su proveedor de servicio. La gestión de claves para la distribución de claves de la sesión está bien integrada en el perfil de seguridad híbrido.

[12] Salto tiene aquí el sentido de un elemento de red H.235 de confianza (por ejemplo, GK, GW, MCU, apoderado o 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.

La provisión adecuada de los servicios de seguridad antes descritos evita varios tipos de ataques, incluyendo:

- *Ataques por intermediarios*: la autenticación e integridad de los mensajes salto por salto en el nivel de aplicación evita tales ataques cuando el intermediario es un salto en el nivel de aplicación, es decir un encaminador hostil.
- *Ataques por reproducción*: La utilización de indicaciones de tiempo y números secuenciales evita estos ataques.
- *Piratería*: la autenticación del usuario evita estos ataques.
- *Asaltos a la conexión*: la utilización de autenticación/integridad para cada mensaje de señalización evita estos ataques.

F.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. En esta Recomendación, la referencia a un documento, en tanto que autónomo, no le otorga el rango de una Recomendación.

– Recomendación UIT-T H.225.0, versión 4 (2000), *Protocolos de señalización de llamada y paquetización de trenes de medios para sistemas de comunicación multimedios por paquetes*.

– Recomendación UIT-T H.235, versión 2 (2000), *Seguridad y criptado para terminales multimedios de la serie H (basados en las Recomendaciones H.323 y H.245)*.

– Recomendación UIT-T H.245, versión 8 (2001), *Protocolo de control para comunicación multimedios*.

– Recomendación UIT-T H.323, versión 4 (2000), *Sistema de comunicación multimedios basados en paquetes*.

– IETF RFC 3280 (2002), *Internet X.509 Public Key Infrastructure Certificate and Revocation List (CRL) Profile*.

F.3 Acrónimos

Este anexo utiliza los siguientes acrónimos.

GCF Confirmación de controlador de acceso (*gatekeeper confirm*)

GK Controlador de acceso (*gatekeeper*)

GRQ Petición de controlador de acceso (*gatekeeper request*)

ICV Valor de comprobación de integridad (*integrity check value*)

LRQ Petición de localización (*location request*)

OID Identificador de objeto (*object identifier*)

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

RCF Confirmación de registro (*registration confirm*)

RRQ Petición de registro (*registration request*)

RSA Algoritmo de criptación Rivest, Shamir y Adleman (*Rivest, Shamir and Adleman encryption algorithm*)

SHA Algoritmo troceado asegurado (*secure hash algorithm*)

URQ Petición de desregistro (*unregistration request*)

F.4 Convenios de especificación

El perfil de seguridad híbrido utiliza términos y definiciones de los anexos D y E.

Si bien el servicio de integridad de mensaje siempre proporciona autenticación de mensaje, lo inverso no siempre es cierto. En el modo sólo autenticación, la integridad asegurada abarca solamente un determinado subconjunto de campos del mensaje. Esto se aplica a los servicios de integridad realizados por medios asimétricos (por ejemplo, firmas digitales). Por tanto, en la práctica, un servicio combinado de autenticación y seguridad utiliza el mismo material de claves sin que con ello introduzca una debilidad en la seguridad.

Este perfil de seguridad es aplicable en ambientes en los cuales puede haber muchos terminales y donde no es factible la asignación de contraseñas estáticas y/o claves simétricas, por ejemplo, en los escenarios a gran escala o a escala global. En cambio, este perfil de seguridad supone la disponibilidad de una infraestructura de claves públicas con certificados asignados y claves privadas/públicas, directorios, etc. Además, este perfil de seguridad despliega criptotécnicas simétricas cuando sean aplicables.

Este perfil de seguridad introduce los términos "primer" mensaje y "último" mensaje enviados. La protección de seguridad del primer mensaje (y probablemente también del último) es diferente de la protección de seguridad de los mensajes restantes.

Por "primer mensaje" enviado se entiende un mensaje que se transmite entre dos entidades H.323 y establece un contexto de seguridad. Pone a disposición de ambas entidades el material de claves simétricas disponible y por ejemplo señala el comienzo de una llamada. En el caso de RAS H.225.0, el primer mensaje es el RRQ y el mensaje de respuesta conexo. Para la señalización de llamada H.225.0 mediante arranque rápido, el primer mensaje es SETUP (ESTABLECIMIENTO) y CONNECT (CONEXIÓN).

El "último mensaje" termina el contexto de seguridad establecido. El material de claves establecido será destruido. Para RAS H.225.0, el último mensaje es el URQ y el mensaje de respuesta conexo, en tanto que para la señalización de llamada H.225.0 el último mensaje es RELEASE-COMPLETE (LIBERACIÓN COMPLETA).

Este perfil de seguridad supone el modelo de llamada con encaminamiento por GK, en el que se aplica el método de señalización de llamada con conexión rápida. Los mensajes de control de llamada H.245 se tunelizan en forma securizada en mensajes de señalización de llamada H.225.0 y heredan por consecuencia el esquema de protección de seguridad H.225.0.

El perfil de seguridad de firma permite tunelizar en forma securizada las PDU de control de llamada H.245 dentro de mensajes de facilidad H.225.0. Los mecanismos de actualización y sincronización de claves H.245 necesitan la tunelización para señalar el mensaje FACILITY (FACILIDAD) de actualización de clave y es útil por ejemplo en las llamadas de muy larga duración.

La zona sombreada diagonalmente en el cuadro F.1 representa los mecanismos de seguridad utilizados por el perfil de seguridad híbrido.

NOTA – Los certificados RSA con troceado MD5 no son parte de este perfil de seguridad.

El perfil de seguridad con criptación de voz del anexo D (véase D.7) podría ser utilizado facultativamente junto con el perfil de seguridad híbrido. Su utilización se negocia como parte de la señalización de establecimiento de la comunicación.

Cuadro F.1/H.235 – Visión general del perfil de seguridad híbrida

Servicios de seguridad	Funciones de llamada			
	RAS	**H.225.0**	**H.245 (nota 3)**	**RTP**
Autenticación	Firma digital RSA (SHA1) HMAC-SHA1-96	Firma digital RSA (SHA1) HMAC-SHA1-96	Firma digital RSA (SHA1) HMAC-SHA1-96	
No repudio	(sólo es posible en el primer mensaje)	(sólo es posible en el primer mensaje)		
Integridad	Firma digital RSA (SHA1) HMAC-SHA1-96	Firma digital RSA (SHA1) HMAC-SHA1-96	Firma digital RSA (SHA1) HMAC-SHA1-96	
Confidencialidad				
Control de acceso				
Gestión de claves	Atribución de certificado Intercambio de claves Diffie-Hellman autenticadas	Atribución de certificado Intercambio de claves Diffie-Hellman autenticadas		

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

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

NOTA 3 – H.245 tunelizado o H.245 incorporado dentro de conexión rápida H.225.0.

La solución de este anexo se puede aplicar para proteger la integridad de todo el mensaje. Para RAS H.225.0, la protección de integridad cubre el mensaje RAS completo; en el caso de la señalización de llamada cubre el mensaje completo de señalización de llamada, H.225.0, incluidos los encabezamientos Q.931.

Para la autenticación, el usuario debe utilizar un esquema de firma con clave pública/privada. Tal esquema generalmente ofrece mejor integridad.

Esta Recomendación no describe procedimientos de registro, certificación y atribución de certificados desde un centro fiduciario y la asignación de claves privadas/públicas, servicios de directorio, parámetros CA específicos, revocación de certificados, actualización/recuperación de pares de claves y otros procedimientos operacionales y de gestión relativos a los certificados, tales como procedimientos para 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 que intervienen son capaces de determinar implícitamente la utilización, bien de los perfiles de seguridad básicos del anexo D, del perfil de firma del anexo E, o bien de este perfil de seguridad híbrido mediante la evaluación de los identificadores de objeto de seguridad señalados en los mensajes (**tokenOID**, y **algorithmOID**; véase también la cláusula E.8).

F.5 Requisitos relativos a H.323

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

- conexión rápida;
- tunelización H.245; y
- modelo con encaminamiento por GK.

F.6 Autenticación e integridad

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

- **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 aplica correctamente la firma digital a algún dato con clave privada, o bien cuando aplica correctamente un secreto compartido, conexo. Además de esto, el mensaje es protegido contra la manipulación fraudulenta. Ambos servicios de seguridad son proporcionados por el mismo mecanismo de seguridad. La autenticación e integridad combinadas sólo son posibles sobre la base de salto por salto.

NOTA – Cuando se aplican firmas digitales se puede soportar un servicio de seguridad de no repudio; esto depende también de los valores fijados a los bits de utilización de clave de la clave de firma en el certificado (véase también RFC 3280).

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

El procedimiento IV se basa en firmas digitales que utilizan un par de claves privada/pública y en el despliegue de criptotécnicas simétricas para proveer autenticación e integridad de mensajes RAS, Q.931 y H.245. Los terminales pueden utilizar este método si se necesita una seguridad eficiente y escalable.

En dependencia de la política de seguridad, la autenticación puede ser unilateral, o mutua (es decir, el caso en el que la autenticación/integridad también se aplica en el sentido inverso, por lo que se proporciona una seguridad superior). El modo de seguridad preferido es el de autenticación mutua.

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 terminal/controlador de acceso par responderán con un mensaje de rechazo correspondiente que indica un fallo de seguridad. Esto se efectúa fijando el motivo de rechazo a **securityDenial** o a otro código de error de seguridad apropiado de acuerdo a la cláusula B.2.2. Dependiendo de la capacidad para reconocer un ataque y de la manera más adecuada para reaccionar en estos casos, un controlador de acceso que recibe un **xRQ** protegido, cuyos identificadores de objeto (**tokenOID**, **algorithmOID**) no están definidos, ha de responder con un **xRJ** no protegido y con un motivo de rechazo fijado a **securityDenial**, pero también puede descartar este mensaje. El punto extremo descartará el mensaje no protegido recibido, se desconectará y tal vez lo volverá a intentar seleccionando otros OID. Muy probablemente, un controlador de acceso que recibe un mensaje SETUP de señalización de llamada H.225.0 protegido, cuyos identificadores de objeto (**tokenOID**, **algorithmOID**) no están identificados, ha de responder con un mensaje RELEASE COMPLETE no protegido y con el motivo de rechazo fijado a **securityDenied**, pero también puede descartar este mensaje. Ahora bien, un controlador de acceso que reciba un FACILITY H.225.0 protegido, cuyos identificadores de objeto (**tokenOID**, **algorithmOID**) no están identificados, ha de responder con un FACILITY no protegido y con el motivo fijado a **undefinedReason**, pero también puede descartar este mensaje. De igual manera, se debe guardar registro del problema de seguridad encontrado. Como parte de la respuesta retomada, el emisor puede proporcionar una lista de certificados aceptables en testigos separados, a fin de facilitar al recibiente la selección de un certificado adecuado.

Hay una señalización H.235 implícita para indicar la utilización del procedimiento IV y el mecanismo de seguridad aplicado basándose en el valor de los identificadores de objeto (véase también la cláusula F.12) y en los campos del mensaje que han sido llenados. En este texto se hace referencia a los identificadores de objeto mediante letras (por ejemplo, "A").

Este perfil no utiliza los campos ICV de H.235. En su lugar, los valores de comprobación de la integridad criptográfica se introducen en el campo **signature** del **token** en el **cryptoSignedToken** cuando se hace referencia al anexo E, o los valores de comprobación de la identidad se introducen en los campos de troceado del **CryptoToken** cuando se hace referencia al anexo D.

F.7 Procedimiento IV

Cuando se emplea el procedimiento IV para la seguridad salto por salto, deberán aplicarse los siguientes procedimientos. El procedimiento IV une el procedimiento I del anexo D (véase D.6.3.2) y el procedimiento II del anexo E (véase la cláusula E.5).

Para el primer mensaje, incluida la respuesta correspondiente, enviado en cada sentido de transmisión, se utilizará el procedimiento II del anexo E (autenticación e integridad salto por salto, véase E.5) para el que se fijarán los siguientes valores:

- OID "A1" en lugar de OID "A" y OID "S1" en lugar de OID "S". La utilización de estos OID permite identificar el perfil de seguridad híbrida.
- **algorithmOID** en **tokenOID** se fijará a "W", que indica la utilización de la firma RSA-SHA1.
- **signature** contendrá una firma RSA codificada en ASN.1 (véase la cláusula E.10).
- **certificate** debe contener el certificado de usuario del emisor si el receptor no lo ha obtenido por otro medio; **type** contendrá OID "W", que indica que se incluye un certificado RSA-SHA1, u OID "P" (véase la cláusula E.18), en cuyo caso **certificate** contiene un URL.

En un escenario con un solo dominio administrativo, el primer mensaje/respuesta se define como el mensaje/respuesta RAS H.225.0 inicial; éste es generalmente GRQ/GCF o RRQ/RCF. En un escenario de múltiples dominios administrativos, el primer mensaje/respuesta dentro de cada dominio se define como en el caso anterior; el primer mensaje entre los dominios se define como SETUP.

Siempre que se transporte un certificado digital en un mensaje, la entidad receptora verificará si la identidad del emisor coincide con la identidad del certificado, conforme al procedimiento E.12, para evitar ataques por intermediarios.

El emisor y el recibiente intercambian y calculan una cadena de bits secreta Diffie-Hellman autenticada. En el cuadro D.4 se presenta un ejemplo de los parámetros de grupo Diffie-Hellman y se recomienda tomar el número primo de 1024 bits siempre que sea posible, por razones de seguridad. El secreto Diffie-Hellman será calculado para cada tramo, independientemente de que se despliegue o no el perfil de encripción de voz.

A partir de la cadena de bits común que ambas partes calculan, ambas partes derivan un secreto de 160 bits tomando los 160 bits menos significativos. El secreto de 160 bits resultante actúa como la contraseña/secreto compartido que se utiliza en el anexo D.

En un escenario con controladores de acceso en distintos dominios administrativos, el emisor y el receptor utilizarán dos testigos en cada sentido de transmisión para la señalización de llamada H.225.0:

- Un **ClearToken** dentro de **CryptoToken**, que se utiliza para calcular la clave de medios que se comparte entre los terminales (véase D.7.1). Esto es necesario solamente si se va a desplegar criptación de voz.

- Se utiliza un **ClearToken** separado para calcular una clave de enlace que se comparte entre las entidades emisor y receptor para protección del enlace de señalización. Esta clave de enlace sustituye la contraseña compartida entre los controladores de acceso en el anexo D. El **tokenOID** de ese **ClearToken** se fijará a "Q", que indica la utilización de Diffie-Hellman y un perfil de seguridad híbrido. El cálculo de la clave de enlace prosigue de la misma manera que el cálculo de la clave de medios (véase D.7.1).

NOTA 1 – En los entornos encaminados directamente, las entidades y los terminales emisor/receptor corresponden unos con otros. En los entornos con encaminamiento por controlador de acceso, la clave de enlaces se comparte salto por salto entre cada par de controladores de acceso pares, mientras que la clave de medios se comparte de extremo a extremo.

En los entornos con encaminamiento por controlador de acceso, el controlador de acceso reenviará al salto siguiente el testigo Diffie-Hellman recibido del punto extremo.

Se debe utilizar el procedimiento I del anexo D (véase D.6.3.2) para todos los mensajes/respuestas enviados en cada sentido, salvo el primero. Esto se aplica también en un escenario con múltiples controladores de acceso situados dentro de un dominio administrativo. En este caso, no hay necesidad de gestión de claves asimétricas y basta con la aplicación del anexo D.

Se puede utilizar este anexo con los sistemas de la versión 1 de la Rec. UIT-T H.235, teniendo precaución de utilizar, en forma limitada, los ID de los emisores y general ID descritos en la cláusula E.17.

Cabe esperar que un controlador de acceso recibirá de un punto extremo fijo determinado solamente una **RRQ** con testigo DH y firma digital. No obstante, algunos mensajes **RCF**/**RRJ** perdidos o retardados pueden provocar una retransmisión en la que se utilice otro **RRQ** firmado.

Cuando la correspondiente respuesta de registro no llegue a tiempo al punto extremo, éste puede intentarlo de nuevo. En este caso, el punto extremo debe utilizar el testigo DH más reciente, pero el número de secuencia y la indicación de tiempo serán diferentes.

Para un punto extremo fijo determinado, el controlador de acceso ha de utilizar el más reciente de los mensajes **RRQ** firmados recibidos y establecer el secreto compartido a partir del testigo DH, aunque el GK tenga ya un secreto compartido disponible. Es decir, el GK debe reemplazar cualquier secreto compartido existente por el nuevo. El GK tiene que responder con una **RCF** firmada que tenga el testigo DH de respuesta. Conviene que se genere de nuevo el testigo DH de respuesta.

NOTA 2 – El método recomendado y preferido para actualizar la clave es el que utiliza el mensaje FACILITY, como se define en la cláusula F.9. No obstante, se reconoce que la clave se puede actualizar mediante otro **RRQ** firmado aditivo con un nuevo testigo DH.

NOTA 3 – Un controlador de acceso que posea un secreto compartido ha de responder a un **RRQ** protegido por HMAC (conforme al anexo D) con un mensaje de respuesta por HMAC.

F.8 Asociación de seguridad para llamadas concurrentes

Se proporciona una optimización para el caso en que un par fijo de entidades procesen varias llamadas independientes, en paralelo, utilizando un solo canal de señalización de llamada. En lugar de establecer varias claves de enlace con Diffie-Hellman para cada llamada, se define una asociación de seguridad que abarca múltiples llamadas concurrentes.

Dicho sea en una forma más precisa, la asociación de seguridad abarca todas las llamadas entre un par fijo de entidades mientras esté vivo el canal de señalización de llamada. Las entidades utilizan la bandera **multipleCalls** dentro de SETUP para indicar la capacidad de señalización de múltiples llamadas por una sola conexión de señalización de llamada (véase 7.3/H.323).

Si se utiliza una sola conexión de señalización de llamada, sólo se necesita establecer una clave de enlace común; véase la figura F.1.

Por otro lado, si la bandera **multipleCalls** dentro de SETUP no está fijada, se calculará de nuevo, individualmente, una clave de enlace para cada llamada.

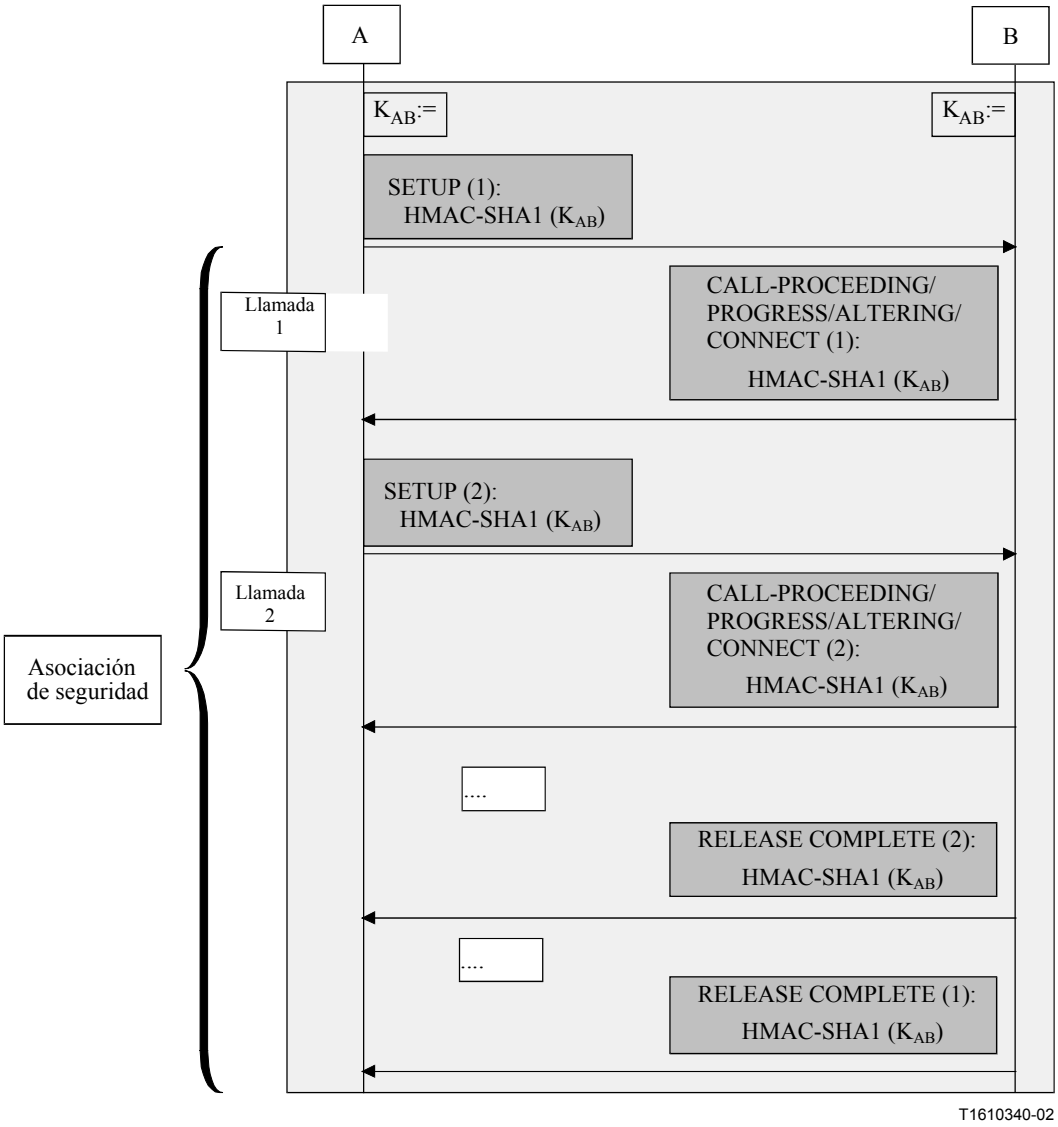


Figura F.1/H.235 – Asociación de seguridad para llamadas concurrentes

F.9 Actualización de clave

Un procedimiento facultativo de actualización de clave permite que cada entidad de comunicación (GK o terminal) renueve la clave de sesión que está utilizando en ese momento, sustituyéndola por una nueva. Tal actualización de clave debe ser iniciada por cualquier entidad que considere que la necesita. Una actualización de clave puede ser motivada por una clave de sesión comprometida, el hecho de considerar que la clave de sesión se ha vuelto, o se volverá, insegura, u otros criterios de políticas de seguridad. Estos aspectos están fuera del alcance de esta Recomendación.

El iniciador invoca la actualización de clave utilizando el mensaje FACILITY. Este mensaje transporta un nuevo testigo Diffie-Hellman, un certificado digital facultativo, y una firma digital del iniciador. Al recibir el mensaje FACILITY, el recibiente contesta con un mensaje FACILITY similar que transporta su testigo Diffie-Hellman, un certificado digital facultativo, y una firma digital del recibiente. Una vez finalizado el procedimiento de actualización de clave, el iniciador y el respondedor utilizarán la nueva clave de enlace calculada.

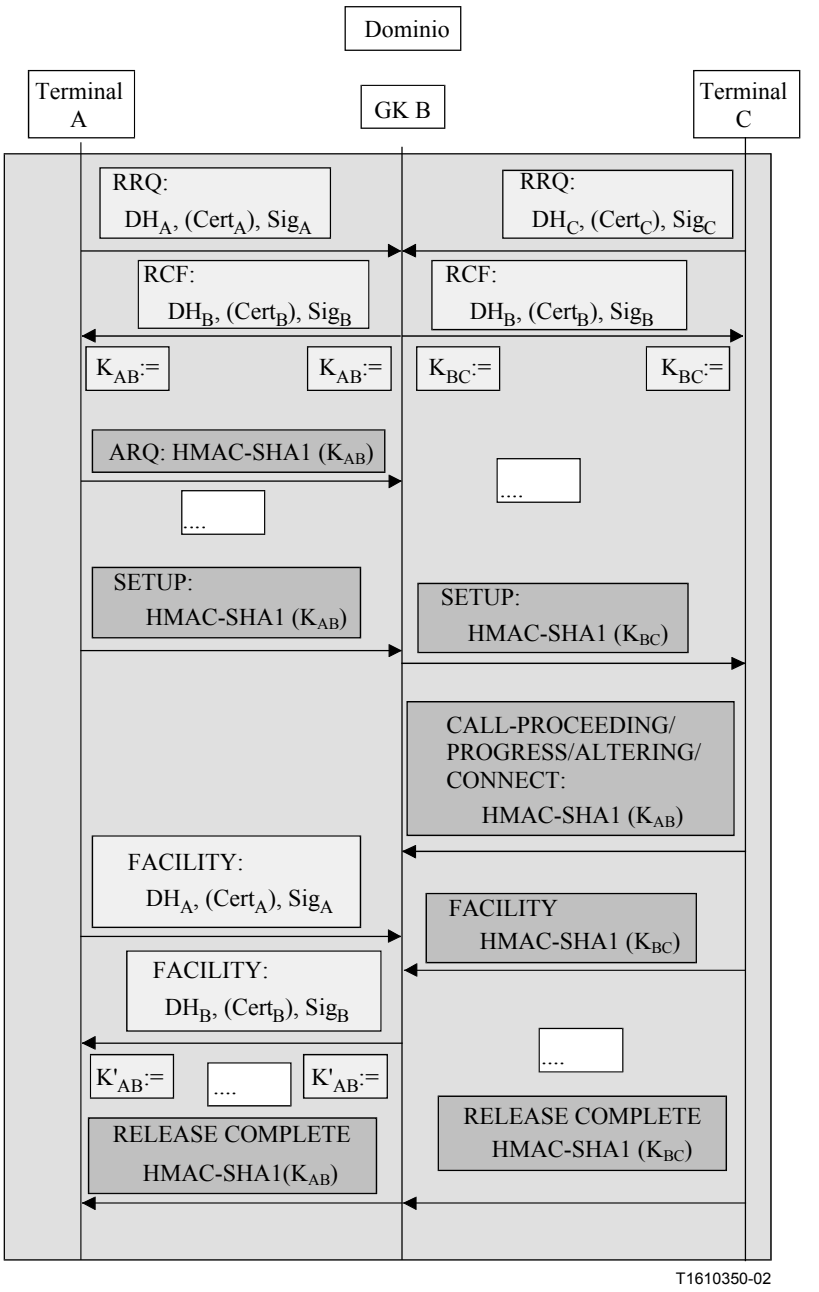
- El **tokenOID** del **ClearToken** dentro de FACILITY se fijará a "Q", que indica la utilización de Diffie-Hellman y el perfil de seguridad híbrido. El cálculo de la clave de enlace prosigue de la misma manera que el cálculo de la clave de sesión de medios (véase D.7.1).

El mensaje FACILITY para fines de actualización de clave se protegerá de conformidad con el procedimiento II del anexo E. Todo otro mensaje FACILITY sin el transporte del testigo Diffie-Hellman no se desplegará para fines de actualización de clave y se protegerá de conformidad con el procedimiento I del anexo D.

F.10 Ejemplos ilustrativos

En los diagramas de flujo de las figuras F.2 y F.3 se ilustra la utilización del anexo F en un flujo de mensaje básico. Se debe observar que los diagramas no muestran el flujo de mensaje completo y que por razones de simplicidad se omiten varios mensajes. Los mensajes resaltados en gris claro se relacionan con el perfil de firma del anexo E, en tanto que los mensajes en gris oscuro se relacionan con el perfil básico del anexo D. Las figuras destacan las partes de seguridad (más importantes) de cada mensaje (CryptoTokens H. 235, testigos) pero se omiten los detalles.

En el diagrama de flujo de la figura F.2 se ilustra el flujo de mensaje básico en un escenario con un controlador de acceso dentro de un dominio administrativo simple. Suponiendo que el certificado del controlador de acceso es conocido por todos los terminales participantes, y que los terminales conocen el certificado del controlador de acceso de la misma manera, no hay necesidad de transmitir los certificados dentro de banda durante el procedimiento de registro.



Cert	Certificado de usuario	K, K'	Clave de enlace simétrica
DH_A	Testigo Diffie-Hellman g^a mod p	Sig	Firma digital
DH_B	Testigo Diffie-Hellman g^b mod p		
EP	Punto extremo (Terminal)		
GK	Controlador de acceso		

Figura F.2/H.235 – Diagrama de flujo en un dominio administrativo simple

NOTA 1 – Las figuras F.2 y F.3 comprenden también el procedimiento de arranque rápido cuando los mensajes de señalización de llamada SETUP y CALL PROCEEDING/PROGRESS/ALERTING/CONNECT incluyen el testigo faststart (véase 8.1.7/H.323). En otro caso, se supone un modo no faststart de conformidad con 7.3.1/H.323. La figura F.2 muestra también el procedimiento de actualización de clave entre el terminal A y el controlador de acceso B mediante FACILITY.

En la figura F.3 se muestra un ejemplo de un flujo de mensaje en un escenario con diferentes dominios administrativos. Si bien el perfil de seguridad híbrido se aplica dentro de cada dominio entre el terminal y el controlador de acceso como se ilustra en la figura F.2, también puede aplicarse entre ambos dominios durante la fase de establecimiento de la comunicación.

NOTA 2 – En la figura F.3 se han omitido todas las comunicaciones entre los elementos de frontera (BE, *border elements*) y todas las comunicaciones entre GK y BE. En la figura F.3 se ilustra también el procedimiento de actualización de clave entre ambos dominios mediante FACILITY.

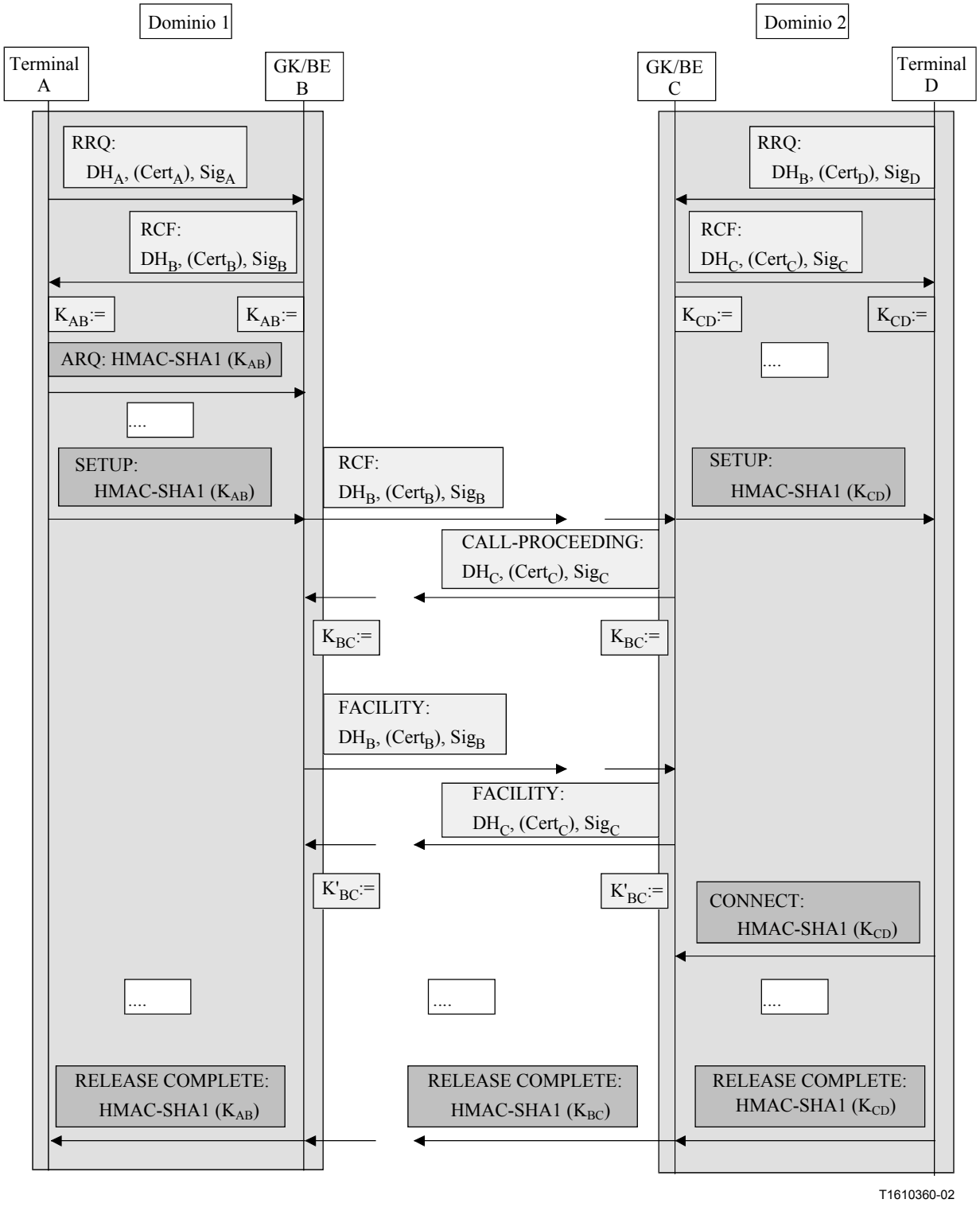


Figura F.3/H.235 – Diagrama de flujo en un dominio administrativo múltiple

F.11 Comportamiento multidifusión

Los mensajes multidifusión H.225.0 tales como **GRQ** o **LRQ** incluirán un **CryptoToken** de conformidad con el procedimiento II en el que no está fijado el **generalID**. Cuando dichos mensajes se envían en modo unidifusión, el mensaje incluirá un **CryptoToken** con el **generalID** fijado.

F.12 Lista de mensajes de señalización securizados

El procedimiento IV despliega el procedimiento I del anexo D o el procedimiento II del anexo E, lo que depende del escenario y del mensaje real, como se indica a continuación.

F.12.1 RAS H.225.0

Mensaje RAS H.225.0	Campos de señalización H.235	Autenticación e integridad	No repudio
GatekeeperRequest, GatekeeperConfirm, GatekeeperReject si se aplica el descubrimiento de GK RegistrationRequest, RegistrationConfirm, RegistrationReject si no se aplica el descubrimiento de GK	CryptoToken, ClearToken	Procedimiento II	Procedimiento II
Cualquier otro mensaje RAS (nota 2)	CryptoToken	Procedimiento I	
NOTA 1 – Para mensajes de unidifusión se aplicarán procedimientos II con los campos seguridad en el **CryptoToken** utilizado. NOTA 2 – No se envían los mensajes de descubrimiento de GK y multidifusión.			

F.12.2 Señalización de llamada H.225.0 (dominio administrativo simple)

Mensaje de señalización de llamada H.225.0	Campos de señalización H.235	Autenticación e integridad	No repudio
Setup-UUIE, Connect-UUIE (nota 1), Facility-UUIE (nota 2), Alerting-UUIE, CallProceeding-UUIE, Facility-UUIE, Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE, Status-UUIE, StatusInquiry-UUIE, SetupAcknowledge-UUIE, Notify-UUIE	CryptoToken, ClearToken	Procedimiento I	
Facility-UUIE (nota 3)	CryptoToken	Procedimiento II	Procedimiento II
NOTA 1 – Se supone que cualquiera de los dos mensajes es el primero en cada sentido de transmisión. NOTA 2 – No se utiliza para actualización de clave. NOTA 3 – Se utiliza para actualización de clave.			

F.12.3 Señalización de llamada H.225.0 (dominio administrativo múltiple)

Mensaje de señalización de llamada H.225.0	Campos de señalización H.235	Autenticación e integridad	No repudio
Setup-UUIE, Connect-UUIE (nota 1), Alerting-UUIE (nota 2), CallProceeding-UUIE, Facility-UUIE (nota 3), Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE	CryptoToken, ClearToken	Procedimiento II	Procedimiento II
Alerting-UUIE (nota 4), CallProceeding-UUIE, Facility-UUIE (nota 5), Progress-UUIE, Information-UUIE, ReleaseComplete-UUIE, Status-UUIE, StatusInquiry-UUIE, SetupAcknowledge-UUIE, Notify-UUIE	CryptoToken, ClearToken	Procedimiento I	Procedimiento I
NOTA 1 – Se supone que cualquiera de los dos mensajes es el primero en cada sentido de transmisión. NOTA 2 – Cualquiera de estos mensajes se transmite como primer mensaje en cualquier sentido. NOTA 3 – Se utiliza para actualización de clave. NOTA 4 – Ninguno de estos mensajes se transmite como primer mensaje en cualquier sentido. NOTA 5 – No se utiliza para actualización de clave.			

F.13 Lista de identificadores de objeto

En el cuadro F.2 se indican todos los OID a que se hace referencia.

Cuadro F.2/H.235 – Identificadores de objeto utilizados en el anexo F

Referencia de identificador de objeto	Valor(es) de identificador de objeto	Descripción
"A1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 20}	Se utiliza como sustituto de OID "A" en el procedimiento II del anexo E para el CryptoToken-tokenOID e indica que la firma/troceado RSA incluye *todos* los campos en los mensajes RAS/ o de señalización de llamada H.225.0 (autenticación e integridad).
"S1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 21}	Se utiliza como sustituto de OID "S" en el procedimiento II del anexo E para el ClearToken-tokenOID e indica que el ClearToken se está utilizando para autenticación e integridad de mensaje. Este OID en el CryptoToken de extremo a extremo también indica, implícitamente, la utilización de DH durante el arranque rápido.
"Q"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 22}	Se utiliza en el procedimiento IV e indica que el ClearToken en el enlace salto por salto transporta un testigo Diffie-Hellman.
"W"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 23}	Se utiliza en el procedimiento IV como un algoritmo OID e indica la utilización de una firma digital basada en SHA1 de RSA.

Anexo G

Utilización del protocolo de transporte en tiempo real seguro (SRTP, *secure real-time transport protocol*) junto con el protocolo de gestión de clave MIKEY en la Rec. UIT-T H.235

Este anexo queda en estudio.

Anexo H

Gestión de clave RAS

Este anexo queda en estudio.

Anexo I

Soporte de llamadas con encaminamiento directo

I.1 Alcance

El objetivo de este anexo es recomendar procedimientos de seguridad para la señalización de llamadas con encaminamiento directo, junto con los perfiles de seguridad D y F de la Rec. UIT-T H.235.

Este perfil de seguridad se ofrece como una alternativa y puede ser complementario con los perfiles de seguridad de los anexos D o F.

En este anexo se proporcionan detalles de implementación para la cláusula B.6 utilizando técnicas de gestión de clave simétrica.

NOTA – Actualmente, en este anexo se presenta un procedimiento de seguridad para un caso limitado, pero puede ocurrir que en un futuro se desarrollen procedimientos de seguridad más complejos para un caso general. Esto será objeto de estudio ulterior.

I.2 Introducción

Con frecuencia se implementa H.323 utilizando el modelo de encaminamiento por controlador de acceso. Por ejemplo, la utilización de este modelo permite soportar mejor facturación ("best billing") y también otras funcionalidades. Asimismo, el uso difundido del modelo de llamada encaminada por controlador de acceso es el motivo por el cual en la Rec. UIT-T H.235 se definen diversos perfiles de seguridad (tales como los de los anexos D, E, F) basados precisamente en este modelo de llamada.

No obstante, debido a que se necesita soportar cada vez más canales paralelos, el modelo de encaminamiento directo con un controlador de acceso podría conducir a mejores calidades de funcionamiento y propiedades escalables. La ventaja de este modelo es que se utiliza el controlador de acceso para el registro, admisión, resolución de direcciones y control de ancho de banda, mientras que se efectúa el establecimiento de llamada directamente entre los puntos extremos.

En este anexo se describen las mejoras a los perfiles de seguridad básicos del anexo D e híbrido del anexo F para el soporte de las llamadas con encaminamiento directo por un controlador de acceso.

I.3 Convenios de especificación

Se hace referencia a los identificadores de objeto mediante un símbolo en el texto (por ejemplo "I1") y en la cláusula I.12 se enumeran los valores numéricos reales para los identificadores simbólicos de objeto, véase también la cláusula 5.

I.4 Términos y definiciones

A efectos de esta Recomendación, se aplican las definiciones dadas en la cláusula 3 de las Recomendaciones UIT-T H.323, H.225.0, H.235 y X.800, junto con las de esta cláusula.

I.5 Símbolos y abreviaturas

En este anexo se usan las siguientes siglas.

$\{M\}_{K;\,S,\,IV}$ Criptación EOFB de M que utiliza claves secretas K y adicional secreta S, además de vector inicial IV

CT Testigo despejado (*ClearToken*)

DRC Llamada con encaminamiento directo (*direct-routed call*)

EPID Identificador de punto extremo (*endpoint identifier*)

GKID Identificador de controlador de acceso (*gatekeeper identifier*)

K_{AG} Secreto compartido (anexo D, anexo F) entre el punto extremo A y el controlador de acceso G

K_{BG} Secreto compartido (anexo D, anexo F) entre el punto extremo B y el controlador de acceso G

KS_{AG} Clave adicional compartida secreta entre el punto extremo A y el controlador de acceso G

KS_{BG} Clave adicional compartida secreta entre el punto extremo B y el controlador de acceso G

K'_{AG} La clave de criptación compartida entre el punto extremo A y el controlador de acceso G

K'_{BG} La clave de criptación compartida entre el punto extremo B y el controlador de acceso G

K_{AB} La clave de criptación compartida entre el punto extremo A y el punto extremo B

I.6 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. En esta Recomendación, la referencia a un documento, en tanto que autónomo, no le otorga el rango de una Recomendación.

– Recomendación UIT-T H.225.0 (2003), *Protocolos de señalización de llamada y paquetización de trenes de medios para sistemas de comunicación multimedios por paquetes*.

– Recomendación UIT-T H.323 (2003), *Sistemas de comunicación multimedios basados en paquetes*.

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

– ISO/CEI 10118-3:2004, *Information technology – Security techniques – Hash functions – Part 3: Dedicated hash-functions*.

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

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

I.7 Generalidades

Los perfiles de seguridad básico del anexo D (véase la parte principal de esta Recomendación) e híbrido del anexo F (véase el anexo F), (tras la primera toma de contacto) se aplican a un secreto compartido para garantizar autenticación de mensaje y/o integridad en un modo de funcionamiento salto por salto, utilizando el controlador de acceso como un intermediario fiable. En el modelo de llamada con encaminamiento directa, no se puede suponer la existencia de un secreto compartido entre dos puntos extremos. Tampoco es práctico utilizar un secreto compartido preestablecido para garantizar la comunicación, puesto que, en este caso, todos los puntos extremos tendrían que saber por adelantado cuál punto extremo será llamado.

En este anexo se trata el caso mostrado en la figura I.1, donde se conectan los puntos extremos a un solo controlador de acceso y se utiliza la señalización de llamada con encaminamiento directo. Se supone que existe una red IP no asegurada en la región del controlador de acceso.

Se supone también que cada punto extremo tiene una relación de comunicación y una asociación de seguridad con el controlador de acceso y que se ha registrado seguramente con él utilizando bien el perfil de seguridad básico o bien el híbrido.

Por lo tanto, el controlador de acceso puede proporcionar un secreto compartido para los puntos extremos que se comunican directamente utilizando un modelo del tipo Kerberos.

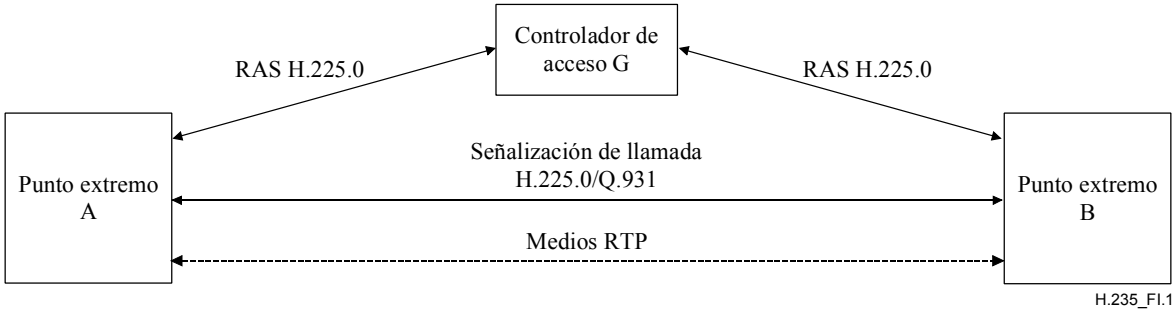


Figura I.1/H.235 – Caso de una llamada con encaminamiento directo

I.8 Limitaciones

En su versión actual, este anexo no trata los casos de encaminamiento directo en que los puntos extremos se conectan a diferentes controladores de acceso. Además, tampoco trata aquellos en los que no hay ningún controlador de acceso. Esto queda en estudio.

I.9 Procedimiento DRC

Los puntos extremos que puedan soportar este perfil de seguridad lo indicarán durante GRQ y/o RRQ incluyendo un ClearToken independiente con **tokenOID** puesto a "I0"; no se debería utilizar ningún otro campo en este ClearToken. El controlador de acceso que tenga las capacidades de este anexo I y desee proporcionar esta funcionalidad responderá con GCF respuesta a RCF con un ClearToken aparte que tenga **tokenOID** puesto a "I0", todos los demás campos en ese ClearToken inutilizados.

Antes de que un punto extremo A empiece a enviar mensajes de señalización de llamada a otro punto extremo B directamente, uno de los dos solicitará admisión en el controlador de acceso G utilizando ARQ. El punto extremo A incluirá dentro de **ARQ** un ClearToken independiente con **tokenOID** puesto a "I0" y todos los demás campos en ese ClearToken inutilizados.

El controlador de acceso al reconocer que los puntos extremos A y B soportan este anexo, generará material de clave y los ClearToken como se indica a continuación.

El controlador de acceso puede evaluar un secreto compartido K_{AB} basado en llamada, además de las operaciones normales ARQ. Este secreto compartido basado en llamada se propaga entonces a ambos puntos extremos utilizando g ClearToken. Estos ClearToken se transportan dentro del mensaje ACF y se envían de nuevo al llamante.

Se incluirán dos ClearToken, un CT_A para el llamante A y otro CT_B para el destinatario B. Cada **ClearToken** contendrá un OID ("I1" o "I2") dentro del **tokenOID** que indique si está destinado al llamante (OID "I1" para CT_A) o al destinatario (OID "I2" para CT_B).

El **ClearToken**, como se define en este anexo, puede ser utilizado junto con otros perfiles de seguridad, como aquellos del anexo D o anexo F, que utilizen también los **ClearToken**. En dicho caso, el **ClearToken** del anexo I también utilizará aquellos otros campos **ClearToken**. Por ejemplo, si se quiere utilizar el anexo I junto con el anexo D, los campos **timeStamp**, **random**, **generalID**, **sendersID**, y **dhkey** estarán presente y se utilizarán como se describe en los perfiles de seguridad del anexo D.

El identificador de controlador de acceso (GKID) se pondrá dentro del **sendersID**, mientras que el **generalID** mantendrá el identificador de punto extremo del punto extremo A (CT_A) o el de punto extremo B (CT_B).

K' indica la clave de criptación compartida entre un punto extremo y el controlador de acceso. Las claves de criptación K'_{AG} y K'_{BG} para la clave extremo a extremo criptada K_{AB} se calcularán a partir del secreto compartido entre el controlador de acceso y los puntos extremos (K_{AG} o K_{BG}) utilizando el procedimiento de cálculo de clave **basado** en PRF que se define en I.10, donde **keyDerivationOID** en **V3KeySyncMaterial** mantendrá "Annex I-HMAC-SHA1-PRF", véase I.12.

El controlador de acceso generará un secreto de sesión compartida común K_{AB}, que será compartido entre los puntos extremos A y B.

Este secreto de sesión K_{AB} será criptado por K'_{AG} (para un CT destinado al punto extremo A) o por K'_{BG} (para un CT destinado al punto extremo B) utilizando un algoritmo de criptación.

El modo de criptación OFB ampliado (EOFB) (véase B.2.5) será utilizado con la clave adicional secreta específica del punto extremo. Los algoritmos de criptación que se pueden utilizar son (véase la cláusula D.11):

- DES (56 bits) en modo EOFB utilizando el OID "Y1": facultativo
- 3DES (168 bits) en el modo EOFB externo utilizando el OID "Z1": facultativo
- AES (128 bits) en el modo EOFB utilizando el OID "Z2": algoritmo recomendado y utilizado por defecto
- RC2 compatible (56 bits) en el modo EOFB utilizando el OID "X1": facultativo.

Para el modo de criptación EOFB, el GK generará un valor aleatorio inicial IV. Si se trata de los OID "X1", "Y1" y "Z1", el IV tiene 64 bits y ha de ser transportado dentro del **iv8** de **paramS** en **V3KeySyncMaterial**; mientras que para el OID "Z2", el IV tiene 128 bits y ha de ser transportado dentro del **iv16** de **params** en **V3KeySyncMaterial**.

El texto cifrado obtenido $\{K_{AB}\}_{K'_{AG},\ KS_{AG},\ IV}$ resp. $\{K_{AB}\}_{K'_{BG},\ KS_{BG},\ IV}$ será entonces transportado en la estructura de datos **h235key** como parte de **secureShareSecret**, donde estará en el

encryptedSessionKey de la estructura de datos **secureSharedSecret**. Se indicará cuál es el algoritmo de criptación en **algorithmOID** ("X1", "Y1", "Z1" o "Z2") en **V3KeySyncMaterial**.

Para el ClearToken destinado al punto extremo A, el identificador de punto extremo B ($EPID_B$) irá dentro de **generalID** de **V3KeySyncMaterial**. De igual manera, para el ClearToken destinado al punto extremo B, el identificador del punto extremo A ($EPID_A$) será ubicado en **generalID** de **V3KeySyncMaterial**.

En el caso de los algoritmos de criptación EOFB, no se utilizará **encryptedSaltingKey**.

El controlador de acceso incluirá tanto los ClearToken CT_A como CT_B en la **ACF** hacia el punto extremo A.

El punto extremo A identificará CT_A inspeccionando el **tokenOID** "I1" dentro de ClearToken.

El punto extremo A verificará que el CT_A obtenido es nuevo comprobando el **timeStamp**. Algunas pruebas adicionales de seguridad permitirán verificar el **generalID** y **sendersID** del ClearToken y el **generalID** dentro de **V3KeySyncMaterial**. Si se verificó el CT_A recibido y se concluyó que era nuevo, el punto extremo A recuperará el IV y calculará K'_{AG} y KS_{AG} como se describe antes para el controlador de acceso. El punto extremo A decriptará la información **encryptedSessionKey** encontrada dentro de **V3KeySyncMaterial** de CT_A para obtener el K'_{AB}.

Si se verificó el CT_A recibido y se concluyó que era nuevo, el punto extremo A puede enviar un mensaje SETUP al punto extremo B. Este mensaje incluye CT_B. El mensaje SETUP se asegurará (autenticándolo y/o protegiendo su integridad) conforme al anexo D o F, utilizando K_{AB} como el secreto compartido aplicado. Para ello, el **generalID** del ClearToken generado numéricamente del anexo D (¡no CT_B!) se fijará a $EPID_B$.

El punto extremo B identificará CT_B inspeccionando el **tokenOID** "I2" dentro de ClearToken.

El punto extremo B verificará que el CT_B obtenido es nuevo revisando la **timeStamp**. Otras pruebas de seguridad adicionales permitirán verificar el **generalID** y el **sendersID** del ClearToken y el **generalID** dentro de **V3KeySyncMaterial**. Si se verificó el CT_B recibido y se encontró que era nuevo, el punto extremo B podrá recuperar el IV y calcular K'_{BG} y KS_{BG} como en el caso descrito para el controlador de acceso. El punto extremo B descifrará la información **encryptedSessionKey** encontrada dentro del **V3KeySyncMaterial** del CT_B para obtener K'_{AB}.

Cuando se haya verificado CT_B y se haya encontrado que es nuevo, el punto extremo B puede proseguir con la señalización de llamada respondiendo con CALL-PROCEEDING, ALERTING o CONNECT, etc. cuando sea necesario. Cuando se haya encontrado que CT_B no es nuevo o que la verificación de seguridad del mensaje SETUP ha fallado, el punto extremo B responderá con RELEASE-COMPLETE y con la **ReleaseCompleteReason** puesta a error de seguridad, como se define en B.2.2.

Cuando se deba utilizar seguridad de medios (véase la cláusula D.7), los puntos extremos A y B intercambiarán medias claves Diffie-Hellman conforme a D.7.1, y establecerán una clave maestra dinámica basada en la sesión a partir de la cual se puedan calcular las claves de sesión específica de medios.

En la figura I.2 se muestra el flujo básico de comunicación.

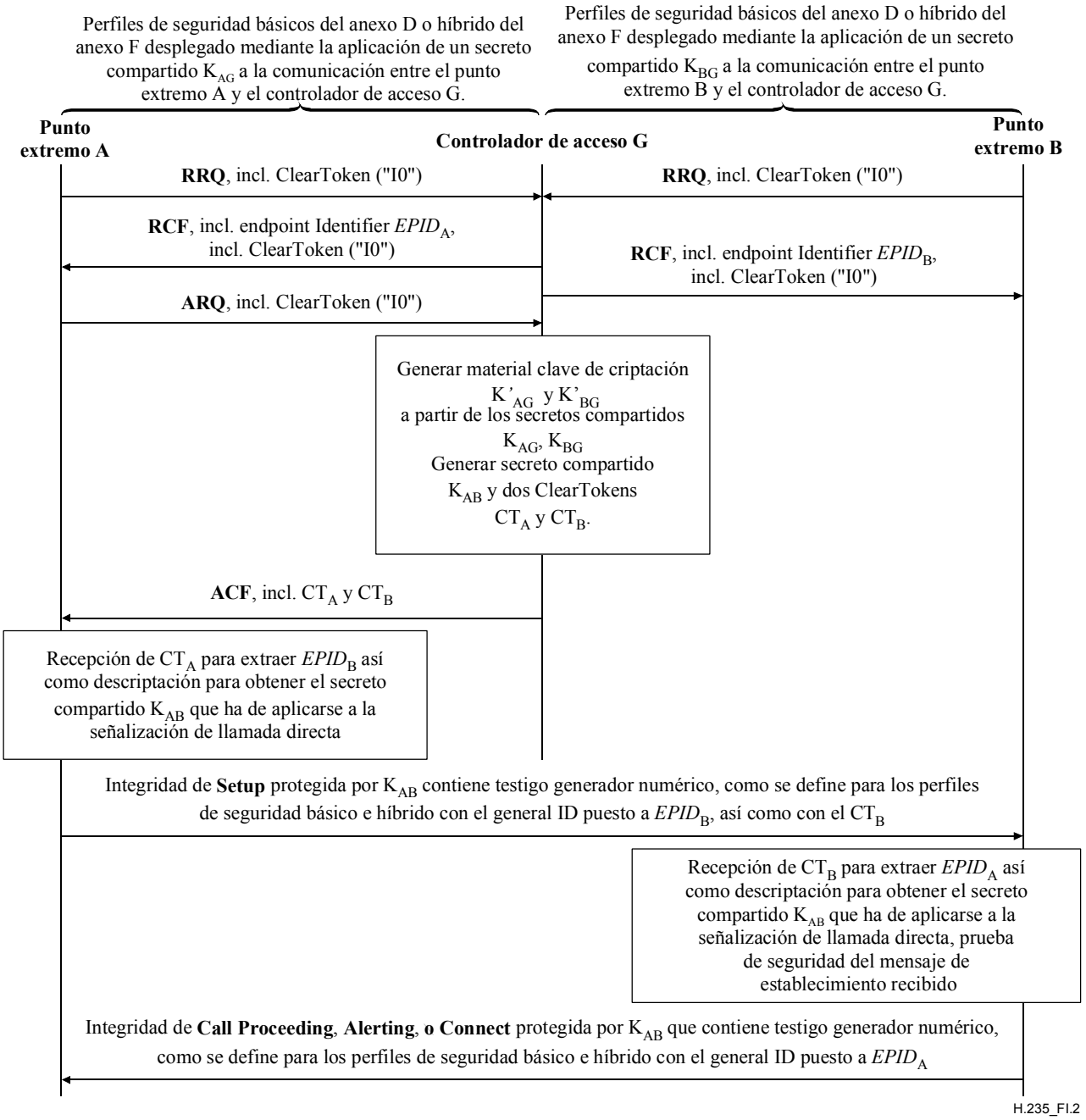


Figura I.2/H.235 – Flujo básico de comunicación

I.10 Procedimiento de cálculo de clave basado en PRF

En esta cláusula se describe un procedimiento para calcular material clave a partir del secreto compartido y otros parámetros.

La clave de criptación K'_{AG} se calculará utilizando la PRF (véase la cláusula B.7) con el parámetro *inkey* puesto a K_{AG} y *label* se fijará a la constante 0x2AD01C64 || **challenge**.

De igual manera, la clave de criptación K'_{BG} se calculará utilizando la PRF con el parámetro *inkey* puesto a K_{BG} y *label* se pondrá al valor constante 0x1B5C7973 || **challenge**. En ambos casos, se asignará a *outkey_len* la longitud requerida de la clave de criptación para el algoritmo de criptación seleccionado.

Utilizando la misma PRF, el controlador de acceso y cada punto extremo generarán la clave adicional compartida y secreta. La clave adicional, siempre que se utilice junto con el modo de

criptación EOFB, protege contra ataques de la CT_B del tipo texto claro conocido por un punto extremo A, siempre que dicho punto pueda de lo contrario intentar descubrir la K_{BG}.

KS_{AG} es la clave adicional compartida y secreta entre el punto extremo A y el controlador de acceso G. KS_{AG} se calculará utilizando la PRF con el parámetro *inkey* puesto a K_{AG} y *label* se pondrá a 0x150533E1 || **challenge**. KS_{BG} se calculará utilizando la PRF con el parámetro *inkey* puesto a K_{BG} y *label* se pondrá a 0x39A2C14B || **challenge**.

NOTA – Los enteros constantes de 32 bits (por ejemplo 0x2AD01C64, etc.) se toman de los dígitos decimales de *e* (es decir, 2,7182 ...), y cada constante consta de nueve cifras decimales (por ejemplo, las primeras nueve cifras decimales 718281828 = 0x2AD01C64). Las cadenas de nueve cifras decimales no se escogen al azar, sino como "pedazos" de las cifras decimales de *e*.

I.11 Procedimiento de cálculo de clave basado en FIPS-140

En esta cláusula se puede describir un procedimiento que define cómo calcular material clave a partir del secreto compartido y otros parámetros utilizando el módulo de criptografía, conforme a FIPS-140. Queda en estudio.

I.12 Lista de identificadores de objeto

Cuadro I.1/H.235 – Identificadores de objeto utilizados por el anexo I

Referencia de identificador de objeto	Valor de identificador de objeto	Descripción
"I0"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 48}	Utilizado en el procedimiento DRC durante GRQ/RRQ y GCF/RCF y ARQ para dejar que los puntos extremos/ controlador de acceso indiquen soporte del anexo I.
"I1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 49}	Utilizado en el procedimiento DRC por el tokenOID del ClearToken que indica que éste mantiene una clave extremo a extremo para el llamante.
"I2"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 50}	Utilizado en el procedimiento DRC por el tokenOID del ClearToken que indica que éste mantiene una clave extremo a extremo para el llamado.
"Annex I-HMAC-SHA1-PRF"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 51}	Usado en el procedimiento DRC para el keyDerivationOID dentro del V3KeySyncMaterial para indicar el método de derivación de cálculo de clave aplicado en I.10 utilizando la función seudoaleatoria HMAC-SHA1.

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.

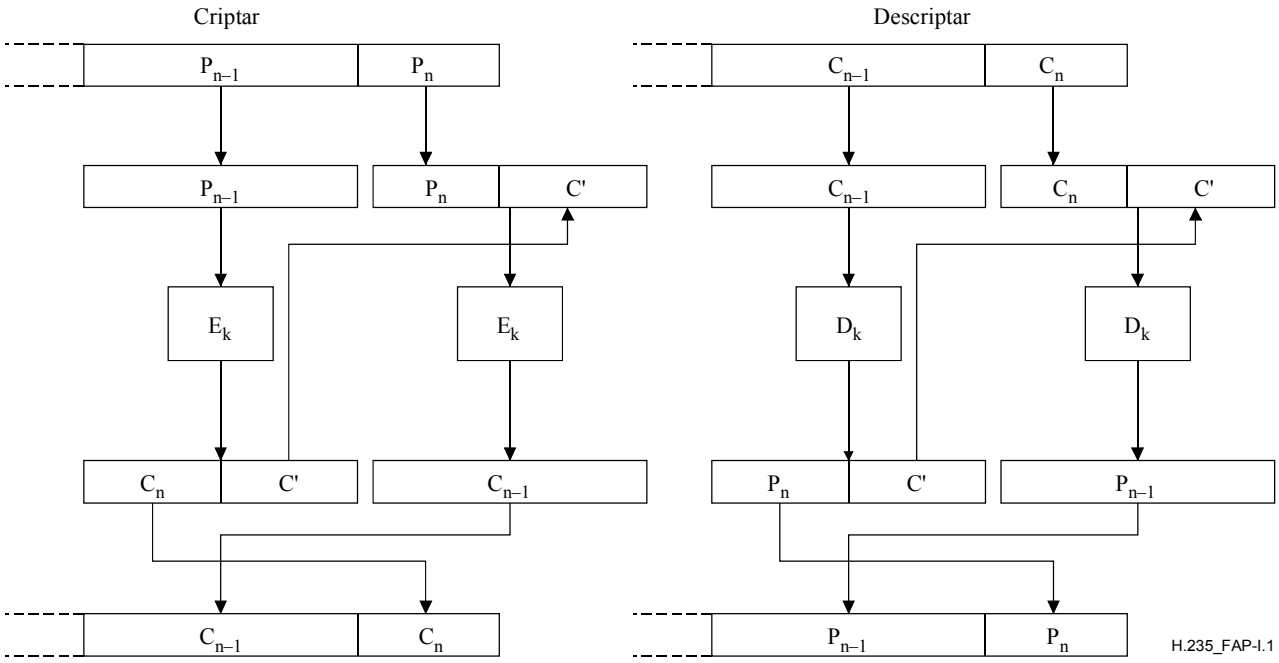


Figura I.1/H.235 – Apropiación de texto cifrado en modo ECB

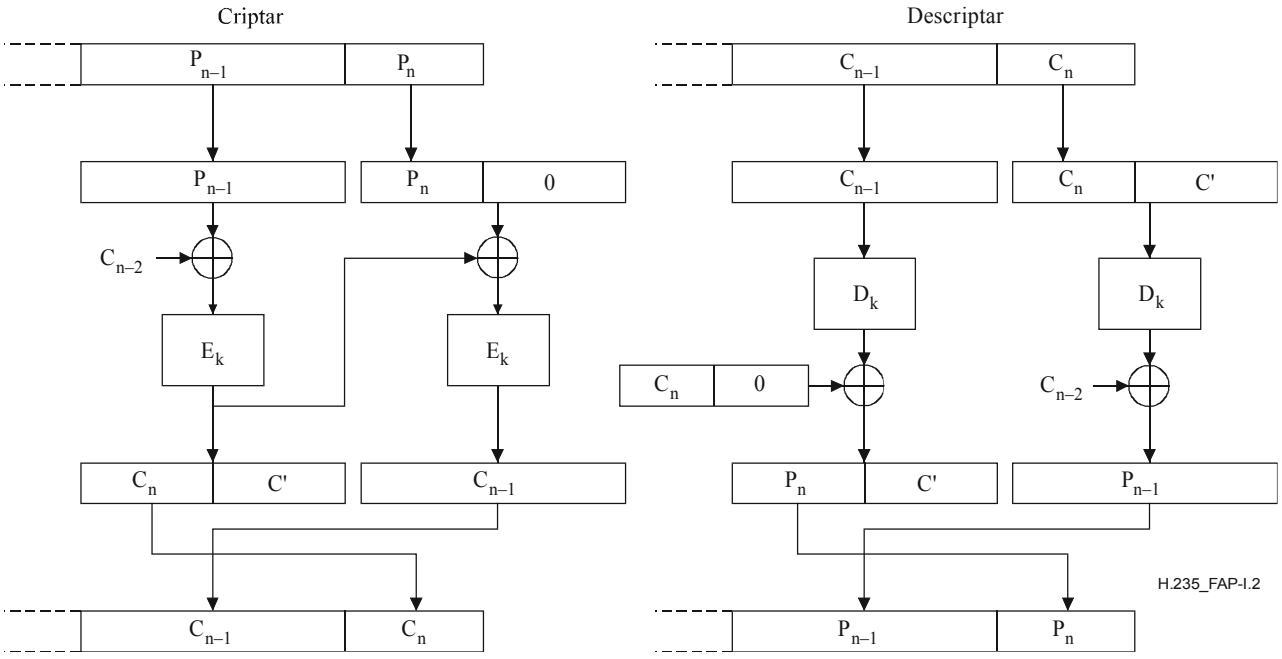


Figura I.2/H.235 – Apropiación de texto cifrado en modo CBC

NOTA – Para la apropiación de texto cifrado en los modos ECB o CBC es necesario que la cabida útil transporte al menos un bloque completo. Las implementaciones que utilicen apropiación de texto cifrado en el modo ECB o los modos CBC deberían garantizar que la cabida útil siempre transporta al menos un bloque criptado; por ejemplo, escogiendo adecuadamente la tasa de muestreo/paquetización o mediante la selección del algoritmo de criptación adecuado.

Cuando la cabida útil ocupe más de un bloque, el vector inicial (IV) se utilizará como el bloque anterior de texto cifrado cuando se aplique la apropiación de texto cifrado en el modo CBC.

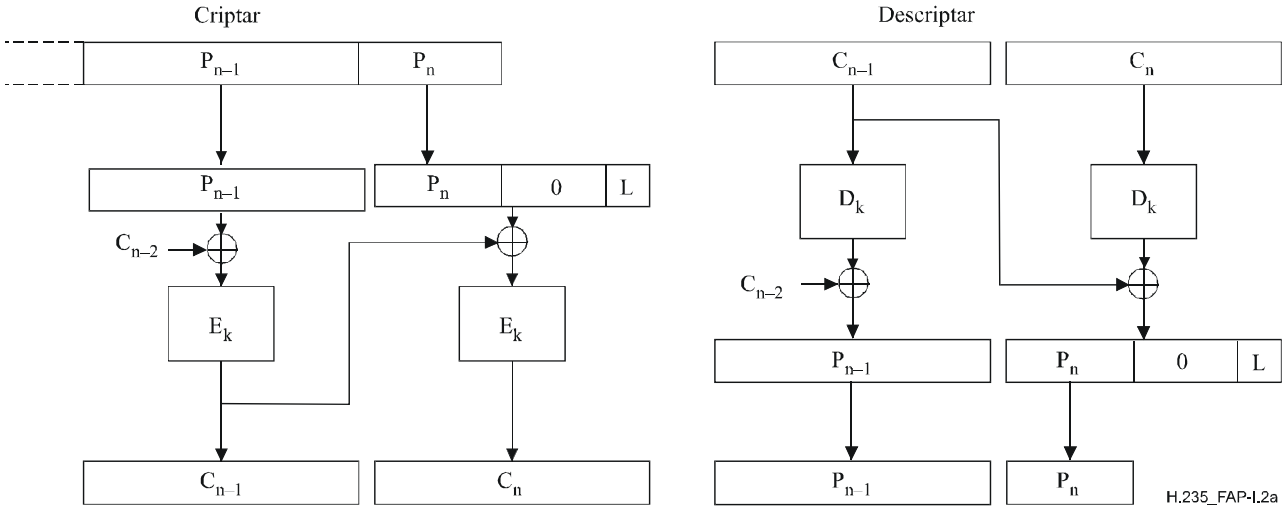


Figura I.2a/H.235 – Relleno de ceros en modo CBC

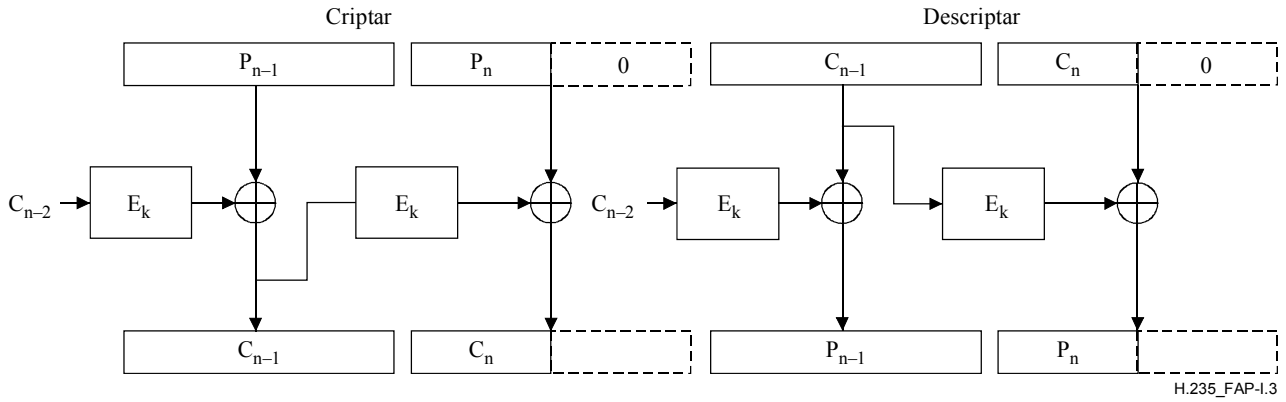
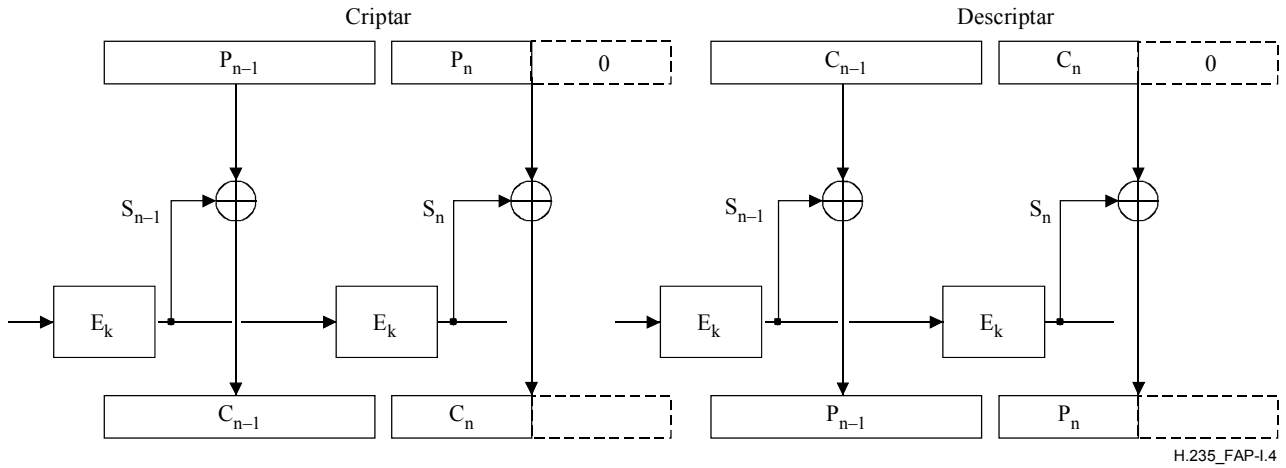


Figura I.3/H.235 – Relleno de ceros en modo CFB



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

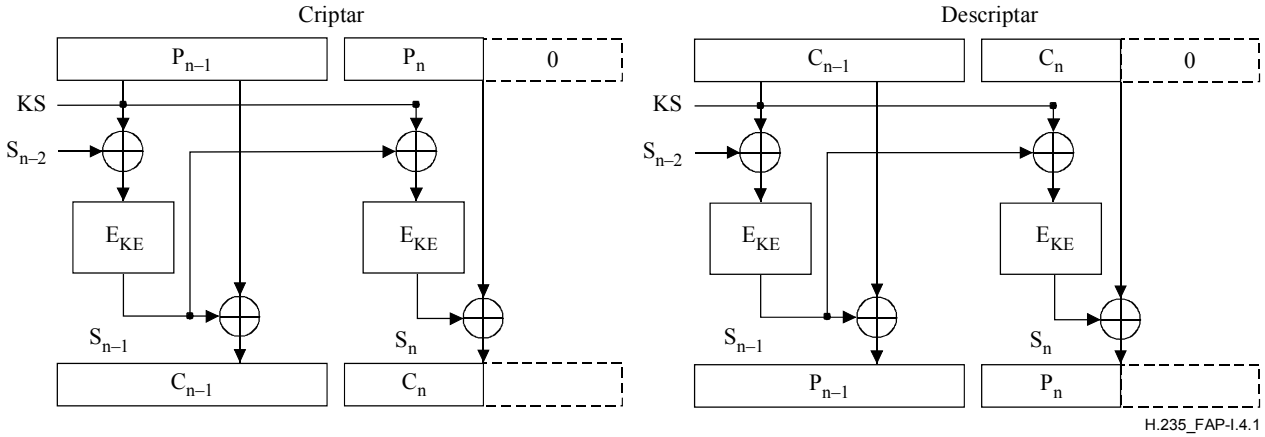


Figura I.4.1/H.235 – Modo EOFB con relleno de ceros

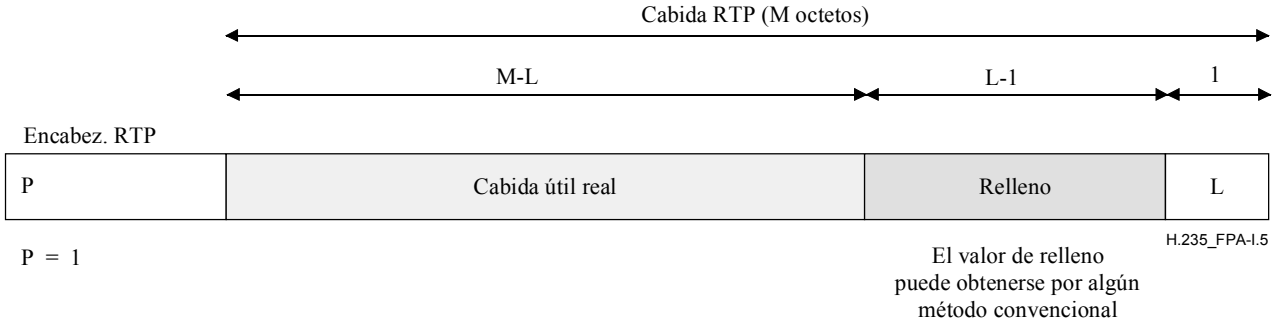


Figura I.5/H.235 – Relleno prescrito por RTP

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 servidores intermedios/cortafuegos (aunque no son elementos específicos H.323), porque terminan conexiones, y pueden tener que manipular los mensajes y los trenes de medios.

I.4 Ejemplos de implementaciones

En las siguientes subcláusulas 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 la Rec. 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.

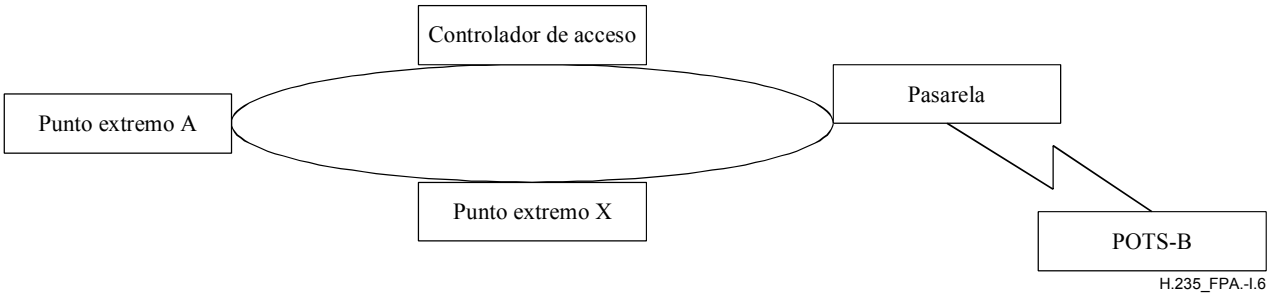


Figura I.6/H.235 – Testigos

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 una **indicación de tiempo**.) 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** u otro código de error de seguridad adecuado, conforme a B.2.2. 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.

   NOTA – El mensaje **GRQ** puede ser unidifundido al controlador de acceso.

I.4.5 IPSEC

En general IPSEC [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 servidores intermedios 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 la Rec. 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 la Rec. UIT-T H.323 el modo en los BES comunican exactamente con el GK. Se pueden aplicar varios métodos y protocolos: RADIUS (véase RFC 2865) se considera uno de los más importantes, y es desplegado ampliamente por los proveedores del servicio.

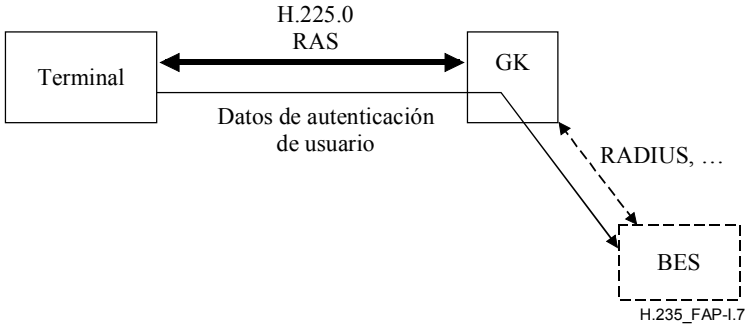


Figura I.7/H.235 – Escenario con servidor fuera del terminal

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

1) **modo por defecto**: en este modo 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 descripta 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**: en este modo el BES y el usuario terminal comparten un secreto común y el GK no debería ser servidor intermedio para la autenticación RADIUS de BES. El GK simplemente reenvía una consulta 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 31}	indica una consulta RADIUS en el ClearToken
"L"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 32}	indica una respuesta RADIUS (cursada en el campo consulta) en el ClearToken
"M"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 33}	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 | CEI 14888-3] *Information technology – Security techniques – Digital signatures with appendix; Part 3: Certificate-based mechanisms*, 1998.

[J.170] ITU-T J.170 (2002), *IPCablecom security specification*.

[MIKEY] ARKKO (J.), CARRARA (E.), LINDHOLM (F.), NASLUND (M.), NORRMAN (K.), "*MIKEY: Multimedia Internet KEYing*", Internet Draft <draft-ietf-msec-mikey-06.txt>, RFC xxxx, Work in Progress (MSEC WG), IETF, 02/2003.

{Editor's note: This RFC # will be included when available}

[PKCS] PKCS #1 v2.0: *RSA Cryptography Standard;* RSA Laboratories; October 1, 1998; http://www.rsa.com/rsalabs/pubs/PKCS/index.html.

[PKCS] PKCS #7: *Cryptographic Message Syntax Standard*, An RSA Laboratories Technical Note, version 1.5, Revised November 1, 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 3550, *Internet Engineering Task Force*, 2003.

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

[SRTP] Baugher, McGrew, Oran *et al*: *The Secure Real-Time Transport Protocol*; draft-ietf-avt-srtp-07.txt, RFC xxxx; Internet Engineering Task Force, 2003.

{Editor's note: This RFC # will be included when available}

[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	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Y	Infraestructura mundial de la información y aspectos del protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación

Impreso en Suiza
Ginebra, 2004