UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T H.235

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

Enmienda 1
(04/2004)

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)

Enmienda 1

Recomendación UIT-T H.235 (2003) – Enmienda 1

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.349
Arquitectura de servicios de directorio para servicios audiovisuales y multimedios	H.350–H.359
Arquitectura de la calidad de servicio para servicios audiovisuales y multimedios	H.360–H.369
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)

Enmienda 1

Resumen

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) 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.

Esta enmienda amplía la versión 3 de la Rec. UIT-T H.235 al incluir el nuevo anexo H y ampliar la funcionalidad del anexo I. Se han introducido modificaciones en la ASN.1 para mejorar el anexo H. Pueden utilizarse para cualquier otro fin identificado por el elemento **profileInfo** de ClearToken. Esta enmienda contiene también algunas correcciones y actualiza el texto de la versión 3 de la Rec. UIT-T H.235.

Orígenes

La enmienda 1 a la Recomendación UIT-T H.235 (2003) fue aprobada el 6 de abril de 2004 por la Comisión de Estudio 16 (2001-2004) del UIT-T por el procedimiento de la Recomendación UIT-T A.8.

PREFACIO

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

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

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

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

NOTA

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

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

PROPIEDAD INTELECTUAL

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

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

ÍNDICE

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)

Enmienda 1

...

2 Referencias normativas

...

– 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 (2003~~0~~), *Seguridad y criptado para terminales multimedios de la serie H (basados en las Recomendaciones H.323 y H.245)*.

...

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

– IETF RFC 3546 (2003), *Transport Layer Security(TLS) Extensions*.

...

3 Términos y definiciones

...

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

3.8*bis* EC-GDSA: Firma digital de curva elíptica con apéndice análoga al algoritmo de firma digital NIST (DSA, *digital signature algorithm)*]; (véase también ISO/CEI 15946-2, capítulo 5).

3.8*ter* criptosistema de curva elíptica (ECC, *elliptic curve cryptosystem*): Un criptosistema de claves púplicas (véase la sección 8.7 del Foro *ATM Security Specification Version 1.1*).

3.8*quat* Esquema de convenio de claves de curva elíptica – Diffie-Hellman (ECKAS-DH, *elliptic curve key agreement scheme Diffie-Hellman*): El esquema de convenio de claves Diffie-Hellman que utiliza criptografía de curva elíptica.

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.

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*)
CTR	Modo contador (*counter mode*) (véase NIST800-38A)
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*)
EC-GDSA	Firma digital de curva elíptica con apéndice análoga al algoritmo de firma digital NIST [*elliptic curve digital signature with appendix analog of the NIST digital signature algorithm (DSA)*]
ECKAS-DH	Esquema de convenio de claves de curva elíptica – Diffie-Hellman (*elliptic curve key agreement scheme – Diffie-Hellman*)
EOFB	Modo OFB mejorado (*enhanced OFB mode*)
EP	Punto extremo (*endpoint*)
GCF	Confirmación de controlador de acceso (*gatekeeper ConFirm*)
GK	Controlador de acceso (*gatekeeper*)
GRJ	Rechazo de controlador de acceso (*gatekeeper ReJect*)
GRQ	Petición de controlador de acceso (*gatekeeper ReQuest*)
GW	Pasarela (*gateway*)
HMAC	Código de autenticación de mensaje troceado (*hashed message authentication code*)
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)*
LCF	Confirmación de localización (*location ConFirm*)
LDAP	Protocolo ligero de acceso al directorio (*lightweight directory access protocol*)
LRJ	Rechazo de localización (*location ReJect*)
LRQ	Petición de localización (*location ReQuest*)

MAC	Código de autenticación de mensaje (*message authentication code*)
MCU	Unidad de control multipunto (*multipoint control unit*)
MD5	Message Digest 5
MIM	Hombre-en-el-medio (*man-in-the-middle*)
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*)
PIN	Número de identificación personal (*personal identification number*)
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*)
RAS	Registro, admisiones y situación (*registration, admissions, and status*)
RCF	Confirmación de registro (*registration ConFirm*)
RRJ	Rechazo de registro (*registration ReJect*)
RRQ	Petición de registro (*registration ReQuest*)
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*)
SHA	Algoritmo de troceado securizado (*secure hash algorithm*)
SHA1	Algoritmo de generación numérica seguro N.º 1 (*secure hash algorithm 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*)
XOR, ⊕	O exclusivo (*exclusive OR*)
X || Y	Concatenación de X e I (*concatenation of X and Y*)

5 Convenios

...

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)**.

Esta Recomendación utiliza la noción de fijar una estructura de datos ASN.1 compuesta a NULL; por ejemplo, "fija **paramS** a NULL" (véanse D.6.3.2, D.6.3.3.3, D.6.3.4.1, D.6.3.4.2, E.5, E.7, E.13.1 y E.13.2). Esto significará que todos los elementos facultativos en la SEQUENCE de que se trate (es decir, **Params**) están ausentes.

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.

...

6 Presentación del sistema

6.1 Resumen

1) El canal de señalización de llamada se puede asegurar utilizando TLS ([RFC 2246~~TLS~~], [RFC 3546]) o IPSEC ([RFC 2402~~IPSEC~~], [ESP]) en un puerto conocido seguro (Rec. UIT-T H.225.0).

...

6.2 Autenticación

...

Como una tercera opción, la autenticación puede ser completada dentro del contexto de un protocolo de seguridad distinto, tal como TLS ([RFC 2246~~TLS~~], [RFC 3546]) o IKE~~IPSEC~~ [IKE~~PSEC~~].

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.

...

6.9 Perfiles de seguridad

Esta Recomendación tiene varios anexos (por ejemplo, anexos D, E, ~~y~~ F y H) 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.

...

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
     keyExch   OBJECT IDENTIFIER -- key exchange profile

}
```

```
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
     profileInfo    SEQUENCE OF ProfileElement OPTIONAL  -- profile-specific

}

--   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...
--

ProfileElement      ::= SEQUENCE
{
     elementID      INTEGER (0..255), -- element identifier, as defined by
                                      -- profile
     paramS         Params OPTIONAL,  -- any element-specific parameters
     element        Element OPTIONAL,         -- value in required form
     …
}

Element ::= CHOICE
{
     octets              OCTET STRING,
     integer             INTEGER,
     bits                BIT STRING,
     name                BMPString,
     flag                BOOLEAN,
     …
}


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
```

...

El anexo H siguiente es nuevo.

Anexo H

Marco para autenticación securizada en RAS mediante el empleo de secretos compartidos débiles

Resumen

Este anexo proporciona el marco para la autenticación mutua de las partes durante intercambios RAS. Los métodos "prueba-de-posesión" aquí descritos permiten securizar el uso de secretos compartidos tales como contraseñas que, si se utilizaran solas, no proporcionarían seguridad suficiente.

Se describen también extensiones al marco para permitir la negociación simultánea de parámetros de seguridad de la capa de transporte para la protección de un subsiguiente canal de señalización de llamada.

Palabras clave

Autenticación, contraseña, seguridad.

H.1 Introducción

En muchas aplicaciones, un punto extremo (o su usuario) y su controlador de acceso pueden compartir solamente un "pequeño" secreto como una contraseña o un número de identificación personal (PIN). Ese secreto (que en adelante se designará por "contraseña"), y toda clave de criptación derivada del mismo, es criptográficamente débil. Los esquemas de autenticación descritos en la cláusula 10, proporcionan ejemplos de texto simple y el correspondiente texto cifrado, y son por tanto susceptibles de ataques de tipo fuerza bruta por un observador de la transacción cuando las autenticaciones se introducen por contraseñas simples. Por tanto, el observador puede recuperar la contraseña y o PIN y después hacerse pasar como el punto extremo para obtener el servicio.

Una familia de protocolos bajo el encabezamiento genérico de intercambio con clave criptada utiliza un secreto compartido para "oscurecer" un intercambio con clave Diffie-Hellman de tal manera que el atacante tenga que resolver una serie de problemas de logaritmo finito para validar un ataque por fuerza bruta contra el secreto compartido. En el intercambio de clave criptada (EKE, *encrypted key exchange*) de Bellovin y Merritt [B&M], el secreto compartido se utiliza para criptar las claves públicas Diffie-Hellman con un algoritmo simétrico. En el método SPEKE de Jablon [Jab], el secreto compartido se utiliza para elegir un generador diferente del grupo Diffie-Hellman. Estos protocolos combinan la seguridad de un intercambio de claves Diffie-Hellman fuertes con el uso del secreto compartido, de tal manera que un atacante no pueda obtener un texto simple conocido para uso en un ataque de tipo fuerza bruta contra el secreto sin haber resuelto el problema de logaritmo finito de Diffie-Hellman. Una ventaja de esos protocolos es que multiplican la fuerza criptográfica del problema Diffie-Hellman por la fuerza de la criptación de la clave secreta (o viceversa). Una desventaja potencial es que están típicamente sujetos a la protección de patentes.

H.2 Alcance

Este anexo lo puede utilizar cualquier controlador de acceso o punto extremo mediante el empleo de los protocolos RAS H.225.0.

H.3 Referencias

H.3.1 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.

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

[NIST 800-38A] NIST Special Publication 800-38A 2001, Recommendation for Block Cipher Modes of Operation – Methods and Techniques. http://www.csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf.

H.3.2 Referencias informativas

[AES] CHOWN (P.): Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS), RFC 3268, junio de 2002.

[B&M] BELLOVIN (S.), MERRITT (M.): U.S. Patent 5,241,599, 31 de agosto de 1993, originally assigned to AT&T Bell Laboratories, now assigned to Lucent Technologies.

[Jab] JABLON (D.): Strong Password-Only Authenticated Key Exchange, Computer Communication Review, ACM SIGCOMM, Vol. 26, No. 5, pp. 5-26, octubre de 1996.

[NIST 800-57] NIST Draft Special Publication 800-57, Recommendation on Key Management, Part 1: General Guideline, http://csrc.nist.gov/CryptoToolkit/kms/guideline-1-Jan03.pdf. http://csrc.nist.gov/CryptoToolkit/kms/.

H.4 Definiciones

Ninguna.

H.5 Abreviaturas

Véase la cláusula 4.

H.6 Marco básico

H.6.1 Capacidades de negociación mejoradas en H.235v3

La versión 3 de la Rec. UIT-T H.235 ha sido ampliada ([H235Amd.1]) con miras al soporte de este marco de seguridad, mediante la adición del siguiente elemento genérico al **ClearToken**:

- **profileInfo** es una secuencia de elementos específicos del perfil, identificados cada uno de ellos por su propio valor entero definido por el perfil específico cuyo OID es transportado en el **ClearToken.tokenOID**.

En las siguientes descripciones se pasan en **profileInfo** varios elementos; para facilitar el análisis, a cada uno de estos elementos se le dará un nombre, en vez de un valor identificador.

H.6.2 Utilización entre punto extremo y controlador de acceso

En el marco básico, en el cual el solicitante es un punto extremo que desea inscribirse ante un controlador de acceso, y el respondedor es ese controlador de acceso, se procede de una manera directa. En adelante se supone implícitamente que cada **ClearToken** mencionado se identifica con el **tokenOID** del perfil de identificación. Se supone que el **ClearToken** está extendido. Los elementos **random** y/o **random2** pueden ser utilizados por un perfil de una de estas dos maneras: pueden ser incluidos en el cálculo de la clave de autenticación, y/o pueden incluirse en un **ClearToken** de perfil en cada mensaje RAS subsiguiente (por ejemplo, RRQ/RCF) para evitar ataques por reproducción. El intercambio de registro de punto extremo se efectúa como sigue:

1) El punto extremo anuncia su intención de participar en uno o más esquemas de negociación y autenticación de claves incluyendo el (los) ID de objeto apropiados para el perfil o perfiles deseados en elementos **authenticationMechanism.keyExch** del elemento **authenticationCapability** de la **G**atekeeper**ReQ**uest. Se supone que cada OID específico define completamente un procedimiento de autenticación en términos de un sistema de claves públicas (por ejemplo, Diffie-Hellman o curva elíptica) y un grupo específico (por ejemplo, uno de los grupos OAKLEY de RFC 2412), algoritmo de criptación simétrico (por ejemplo, AES-128-CBC con robo de texto cifrado), función de derivación de clave (por ejemplo, mediante la función seudoaleatoria del anexo B), código de autenticación de mensajes (por ejemplo, HMAC-SHA1-96), y la secuencia en que se utilizan. El punto extremo incluye también uno o más **ClearToken** de perfil en la GRQ, cada uno de los cuales transporta el OID para el perfil específico ofrecido y el material de clave pública (criptado) en la forma siguiente:
   a) **tokenOID** transporta el perfil OID como se ofrece en la **authenticationCapability** de la GRQ encapsulante.
   b) **timeStamp** puede utilizarse para asegurar que la transacción está en curso y proteger contra los ataques por reproducción.
   c) **password** no se utilizará para la contraseña real.
   d) **dhkey** transporta los parámetros de clave Diffie-Hellman, si se utilizan. El elemento **halfkey** encerrado se cripta como se especifica por el perfil seleccionado.
   e) **challenge** no se requiere.
   f) **random** lo suministra la parte iniciadora y se utiliza para prevenir ataques por reproducción.
   g) **certificate** puede utilizarse si el intercambio de certificados forma parte del perfil.
   h) **generalID** puede utilizarse si lo requiere el perfil.
   i) **eckasdhkey** transporta los parámetros de clave por curva elíptica, si los utiliza el perfil. El elemento **public-key** encerrado debe criptarse como se especifica por el perfil.
   j) **sendersID** puede especificarse como se especifica por el perfil.
   k) Un elemento **profileInfo**, **initVect**, puede suministrarse junto con el material de clave pública (criptado) (**dhkey** o **eckasdhkey**) si el perfil requiere un vector de inicialización para descriptación.
   l) Si el iniciador desea utilizar material de clave derivado de un intercambio anterior, incluirá un elemento **profileInfo**, designado por **sessionID**, que contiene el identificador asignado durante el intercambio anterior. En este caso, **dhkey**, **eckasdhkey** y/o **initVect** no deben incluirse.
   m) Si el iniciador desea establecer una sesión TLS para una conexión de señalización de llamada, puede incluir uno o más elementos **profileInfo** que contienen sucesiones cifradas TLS; el mensaje contendrá una sola sucesión cifrada (la negociada previamente) si **sessionID** está presente.

n) Si el iniciador desea establecer una sesión TLS para señalización de llamada, puede incluir un elemento **profileInfo** que contiene una lista de métodos de compresión; sólo un método de compresión (el negociado previamente) deberá incluirse si **sessionID** está presente.

o) Pueden utilizarse más elementos **profileInfo** para todo parámetro adicional requerido por los procedimientos en el perfil.

2) Al recibir la GRQ, el controlador de acceso selecciona un perfil **AuthenticationMechanism** de la lista ofrecida, genera una clave privada adecuada, calcula la clave pública correspondiente, genera un vector de inicialización si se necesita para criptación simétrica utilizando la contraseña, cripta la clave pública, genera un ID de sesión único, y genera un número aleatorio, todo lo cual se codifica en un **ClearToken**. En función del perfil, los elementos del ClearToken se utilizan como sigue:

a) **tokenOID** transporta el OID de perfil, seleccionado por el **authenticationMethod** de la GCF encapsulante.

b) **timeStamp** puede utilizarse para asegurarse que la transacción está en curso y proteger contra ataques por reproducción.

c) **password** no se utilizará para la contraseña real.

d) **dhkey** transporta los parámetros Diffie-Hellman, si se utilizan. El elemento **halfkey** encerrado se cripta como se especifica por el perfil seleccionado.

e) **challenge** se utiliza para transportar un vector de inicialización, si se requiere para criptación de clave como se especifica por el perfil, o puede utilizarse para transportar una cadena aleatoria que habrá de ser devuelta por el punto extremo para prevenir ataques por reproducción.

f) **random** puede contener el valor único, impredecible, suministrado por el solicitante para prevenir ataques por reproducción.

g) **certificate** puede utilizarse si el intercambio de certificados forma parte del perfil.

h) **generalID** puede utilizarse si lo requiere el perfil.

i) **eckasdhkey** transporta los parámetros de curva elíptica, si los utiliza el perfil. El elemento **public-key** encerrado debe criptarse como se especifica por el perfil.

j) **sendersID** puede utilizarse como se especifica por el perfil.

k) **random** (o un elemento **profileInfo** adicional, designado por **random2**, si el perfil requiere que ambos números permanezcan en el intercambio de mensajes) debe contener un valor único, impredecible, suministrado por el respondedor para proteger contra ataques por reproducción.

l) **initVect** se suministra junto con el material de clave pública (criptada) (**dhkey** o **eckasdhkey**) si el perfil requiere un vector de inicialización para descriptación.

m) **sessionID** es un identificador único (para el controlador de acceso) utilizado para identificar esta sesión de registro. En el caso de ciertos perfiles puede utilizarse también como un ID de sesión TLS para establecimiento rápido de un canal de señalización de llamada protegido por TLS.

n) **profileInfo** puede utilizarse para todo parámetro adicional requerido por los procedimientos en el perfil.

El controlador de acceso calcula entonces el secreto compartido o clave maestra utilizando su clave privada y la clave pública (descriptada) a partir de la GCF, y deriva, la clave maestra, las claves de criptación, las claves de autenticación y todo otro material necesario, de acuerdo con el perfil. El **ClearToken** antes descrito se coloca en el mensaje **G**atekeeper**C**on**F**irm. La GCF debe ser verificada en su integridad y/o autenticada

utilizando la clave de autenticación derivada, y después enviada al punto extremo. La autenticación/verificación de integridad puede devolverse de una de varias maneras, como se especifica por el perfil: mediante un elemento **profileInfo** específico del perfil, o mediante uno de los procedimientos especificados en el anexo D.

3) El punto extremo examina el **authenticationMechanism.keyExch** seleccionado de la GCF y extrae los parámetros del **ClearToken** identificado por el **tokenOID** correspondiente. El punto extremo selecciona entonces su clave privada, calcula la clave pública correspondiente, y selecciona todo otro parámetro requerido por el perfil. El punto extremo calcula después el secreto compartido o la clave maestra utilizando su clave privada y la clave pública (descriptada) a partir de la GCF, y deriva las claves de criptación, claves de autenticación y todo otro material necesario, de dicho mensaje, de acuerdo con el perfil. El punto extremo verificará entonces la integridad de la GCF. Si la verificación de la GCF fracasa, el punto extremo la descartará, junto con todo el material de clave derivado del mismo, y continuará en espera de un mensaje GRQ válido. Una recuperación de RAS estándar conducirá a una retransmisión de la GRQ y, es de suponer, a la recepción de una GCF no dañada. Si tras unas pocas retransmisiones no se obtiene una respuesta exitosa, el punto extremo debe abandonar los intentos de registrarse e informar a su usuario que algo ha fallado. Obsérvese que cada GRQ enviada da a un impostor que se está haciendo pasar por una pasarela una oportunidad más de adivinar la contraseña del usuario y de que su impostura sea validada por la aceptación de la GRQ. Si la verificación de integridad de la GCF tiene éxito, el punto extremo ha validado al controlador de acceso, y puede proceder a registrarse y, en el proceso, a autenticarse a sí mismo ante el controlador de acceso.

4) El punto extremo llena entonces un **ClearToken** con el **tokenOID** de perfil en una forma similar a la empleada por el controlador de acceso, antes descrita. Todo campo del testigo despejado de la GCF que el perfil considera como un desafío debe incluirse en el **ClearToken**. Si así lo especifica el perfil para evitar ataques por reproducción, el **ClearToken** incluirá **random** y **random2** que se toman de la GCF recibida, como se ha expuesto antes. El **ClearToken** se coloca entonces en una **R**egistration **ReQ**uest que habrá de devolverse al controlador de acceso. El punto extremo debe luego autenticar el mensaje RRQ completo y enviarlo al controlador de acceso. A partir de este momento, el punto extremo no debe aceptar, ni tampoco enviar, mensajes RAS que no hayan sido autenticados por el perfil convenido, utilizando la clave de autenticación derivada del material de clave compartido.

5) El controlador de acceso recibe la RRQ, y utilizará el material de clave compartido para verificar su integridad cotejándola con la autenticación y la verificación de integridad incluidas. Si la verificación de la integridad fracasa, el controlador de acceso no tendrá en cuenta la RRQ recibida, y esperará una RRQ verificable. Si no llega ninguna, el punto extremo abandonará finalmente el intento de registro y volverá a la búsqueda de un controlador de acceso. Si la verificación de integridad tiene éxito, el controlador de acceso preparará un mensaje de confirmación de registro para devolverlo al punto extremo. En función del perfil, este mensaje de RCF podrá contener un **ClearToken** que incluya los elementos **random**, **random2**, y/o **challenge** tomados del **ClearToken** de perfil de autenticación proporcionado en la RRQ. La RCF, y todos los mensajes RAS subsiguientes, contendrán una autenticación verificable y una comprobación de integridad calculada utilizando la clave de autenticación y el algoritmo negociados.

6) Cuando el punto extremo recibe el mensaje RCF, verifica la integridad mediante el elemento de autenticación y verificación de integridad incluido. Si la verificación fracasa, se descartará la RCF; si se recibe una RCF no válida, incluso después de haberse retransmitido la RRQ, se abandonará la sesión y el punto extremo volverá a la búsqueda de un nuevo controlador de acceso. Si la RCF se verifica, el ID de sesión y la sucesión cifrada

seleccionada, si están presentes, pueden extraerse de su **ClearToken** para ulterior utilización en el establecimiento de un canal de señalización de llamada securizado.

H.6.3 Utilización de perfiles entre controladores de acceso

Esencialmente, se puede utilizar el mismo procedimiento entre controladores de acceso en un intercambio LRQ/LCF. En esta situación, no es posible una selección explícita del perfil; el controlador de acceso iniciador ofrecerá uno o más perfiles incluyendo el o los **ClearToken** apropiados como se ha descrito antes para el mensaje GRQ. El controlador de acceso respondedor puede elegir un perfil ofrecido y debe devolver el correspondiente **ClearToken** como se ha descrito antes con relación al mensaje GCF. Obsérvese que, en este caso, el controlador de acceso iniciador no se autentica a sí mismo ante el controlador de acceso respondedor hasta que establece un canal de señalización de llamada hacia ese controlador de acceso.

Este procedimiento puede emplearse en un modo multidifusión si un grupo de controladores de acceso comparte un secreto único que habrá de utilizarse con este fin. La LRQ multidifusión se basará en ese secreto; los controladores de acceso que respondan con LCF utilizarán la clave para decodificar la clave pública Diffie-Hellman ofrecida, y cada uno de ellos elegirá su propio **nonce** y clave privada Diffie-Hellman para su respuesta. Las claves de sesión resultantes serán únicas para la pareja final de controladores de acceso.

H.6.4 Criptación y autenticación de canales de señalización

Si el controlador de acceso soporta el encaminamiento por controlador de acceso, el material de clave maestra últimamente negociado y los parámetros criptográficos identificados pueden utilizarse para autenticar y securizar el canal de señalización de llamada, por ejemplo, estableciendo una sesión TLS para señalización de llamada. Si ha de utilizarse TLS, el controlador de acceso incluirá los elementos **cipherSuite** y **compress** seleccionados en el **ClearToken** de perfil devuelto.

H.7 Perfil de seguridad específico (SP1)

Esta cláusula proporciona un perfil de seguridad normalizado, el cual se espera que proporcionará un secreto compartido que, según se considera, será equivalente a un número aleatorio de 80 bits (véase [NIST800-57]). El perfil consta de lo siguiente:

- ID de objeto para este perfil (designado por "SP1") será {itu-t (0) recommendation (0) h (8) 235 version (0) 3 60}.
- Negociación de clave maestra, K_m: intercambio de claves Diffie-Hellman utilizando el conocido grupo 2 de OAKLEY [RFC 2412], seguido de la reducción, por troceado SHA1, del secreto Diffie-Hellman: K_m = SHA1(secreto compartido Diffie-Hellman).
- Algoritmo de criptación simétrica: será AES-128 en modo contador segmentado con un discriminador de parte (participante) de 2 octetos, D, un vector de inicialización de 12 octetos, IV, y un campo contador de 2 octetos, C, de manera que contador = D || IV || C, y C = 0 inicialmente. Para una descripción del modo CTR, véase [NIST800-38A]. El discriminador de parte, D, se fija a 0x3636 cuando el IV es generado por la parte que emitió la GRQ/RRQ, o LRQ, y se fija a 0x5c5c cuando el IV es generado por la parte que respondió con GCF/RCF, o LCF. Cada parte debe asegurarse de que cada IV que generó es único; puede utilizar su propio método para asegurar esta unicidad.
- Criptación de clave Diffie-Hellman: se utilizará el modo contador segmentado AES-128 para criptar la clave pública Diffie-Hellman (representada por una cadena de octetos en el orden en que se transmiten en la red); el vector de inicialización será transportado en **ClearToken.initVect**, y la clave de 16 octetos, K_p, se construirá como los 128 bits de orden superior del troceado SHA1 de la contraseña del usuario: K_p = Trunc(SHA1(contraseña del usuario), 16), donde Trunc(x,y) trunca la cadena de x octetos a y octetos. Se señala que esta clave suele considerarse débil.

- Prevención de ataques por reproducción: cada parte suministrará un número "aleatorio" ("random") de 32 bits (que puede contener un campo contador para garantizar su unicidad); Los números aleatorios se utilizan explícitamente en el cálculo de claves derivadas, por lo cada número aleatorio sólo hay que transmitirlo una vez.
- Derivación de la clave de autenticación, K_a: utilizando la función PRF del anexo B, que se designa por PRF(*in_key*, *label*, *outkey_len*) con *in_key* = K_m, y *label* = "auth_key" || R_e || R_g, donde R_e es un **nonce** obtenido de **ProfileElement** de la GRQ y R_g es un **nonce** obtenido de un **ProfileElement** de la GCF, y *outkey_len* = 128.
- Función de autenticación e integridad de mensaje: utilizando un **ClearToken** con **tokenOID** fijado a "SP1" y un **ProfileElement.octets** fijado al valor de troceado HMAC-SHA196 calculado sobre el mensaje entero como se describe en la Rec. UIT-T H.225.0; este procedimiento se aplicará a todos los mensajes RAS y de señalización de llamada (salvo una GRQ, o LRQ, que no contiene un **sessionID**).
- Clave de criptación de elemento, K_e: elementos seleccionados de mensajes de señalización de llamada (o elementos tunelizados en estos mensajes) pueden ser criptados utilizando AES-128 en modo contador segmentado mediante el uso de la clave K_e = PRF(K_m, "encrypt_key" || R_e || R_g, 128). Por ejemplo, esta clave puede utilizarse para criptar claves de sesión de medios para distribución en elementos **h235Key** como los utilizados en Fast Connect y/o H.245. Cuando se utilizan de esta manera, "SP1" se emplea como el OID de algoritmo de criptación.

Este perfil utiliza los **ProfileElement** definidos en el cuadro H.1. Estos elementos son transportados en una secuencia de elementos **ClearToken.profileInfo** definida en la enmienda 1 al anexo A/H.235.

Cuadro H.1/H.235 Anexo H – Elementos de los perfiles

Nombre del elemento (utilizado en el texto)	Valor del ID del elemento	Opción del elemento (longitud)	Descripción del elemento
initVect	1	Octetos (12)	Vector de inicialización para criptación EKE
nonce	2	Octetos (cualesquiera)	Un valor único, impredecible
cipherSuite	3	Octetos (2)	Una sucesión cifrada TLS
compression	4	Octetos (1)	Un algoritmo de compresión TLS
sessionID	5	Octetos (1..)	Único, puede concordar con un ID de sesión TLS
integrityCheck	6	Octetos (12)	Valor de comprobación introducido

La secuencia de registro consistirá en lo siguiente:

– El punto extremo enviará GRQ con el elemento **authenticationCapability** que contiene un **AuthenticationMechanism.keyExch** que contiene OID "SP1" y un **ClearToken** correspondiente con **tokenID** = "SP1" y **dhkey** que contiene una clave pública de 1024 bits criptada utilizando **initVect** como el IV y la clave derivada de la contraseña del usuario, y un **nonce** = un número aleatorio de 32 bits seleccionado por el punto extremo.

– El controlador de acceso responderá con GCF con el elemento **authenticationMode** igual a un **AuthenticationMechanism.keyExch** que contiene OID "SP1", y un **ClearToken** con **tokenID** = "SP1" y **dhkey** que contiene una clave pública de 1024 bits no criptada, y un **nonce** = un número aleatorio de 32 bits seleccionado por el controlador de acceso, junto con una **integrityCheck** que contiene el valor de troceado de autenticación calculado utilizando la clave de autenticación, K_a, derivada. Obsérvese que el controlador de acceso

no tiene necesidad de criptar su media clave Diffie-Hellman en la GCF, en este perfil, porque él es la primera parte que se autentica a sí misma al demostrar su aptitud para autenticar la GCF utilizando la clave de autenticación derivada. Este modo permite al controlador de acceso reutilizar sus claves Diffie-Hellman con más de un punto extremo. Véase H.9.5.

– El punto extremo responderá con una RRQ con el valor de autenticación y verificación de integridad en un **ProfileElement** con **elementID** fijado a **integrityCheck**, y **element** fijado al valor calculado utilizando la clave de autenticación, K_a, derivada.

– Los mensajes RAS subsiguientes, incluido el de RCF, serán autenticados y comprobados en su integridad utilizando el mismo procedimiento y clave. Los mensajes de señalización de llamada H.225.0 (y los mensajes H.245 tunelizados, si están presentes) serán autenticados utilizando un **ClearToken**, con **tokenOID** fijado a "SP1", que contiene un **profileInfo ProfileElement** con **elementID** fijado a **integrityCheck** y **element** fijado al valor calculado.

– La clave de criptación, K_e, y el algoritmo de criptación AES-128 en modo contador segmentado pueden ser utilizados por el controlador de acceso y el punto extremo para criptar información seleccionada transportada por RAS, señalización de llamada, y/o H.245. Por ejemplo, el controlador de acceso puede distribuir claves de criptación de medios securizadas con K_e y el algoritmo de criptación de perfil.

– Si se requiere que un punto extremo se registre de nuevo, y este punto extremo retiene el ID de sesión original y el secreto maestro, debe tratar de registrarse de nuevo utilizando el ID de sesión original y el secreto maestro incluyendo explícitamente en su GRQ el ID de sesión (pero no una media clave Diffie-Hellman).

– Este perfil podrá utilizarse entre controladores de acceso (véase H.6.3).

H.8 Extensiones al marco (informativo)

Los siguientes elementos pueden incorporarse en un perfil de seguridad definido dentro de este marco.

H.8.1 Utilización de la clave maestra para securizar el canal de señalización de llamada mediante TLS

El material de clave negociado durante el intercambio de RAS puede utilizarse para derivar claves de sesión destinadas a la protección del canal de señalización de llamada en el protocolo de transporte TLS ([RFC 2246], [RFC 3546]). En efecto, la negociación de RAS reemplaza el protocolo de toma de contacto TLS inicial. Desde luego, esto sólo tiene sentido si la señalización de llamada va a ser encaminada por un controlador de acceso. Es especialmente conveniente para autenticación por controlador de acceso y señalización mediante intercambio de LRQ/LCF. En este caso, no hay un tercer mensaje RAS por el cual el controlador de acceso llamante pueda autenticarse a sí mismo ante el controlador de acceso llamado utilizando el material de clave negociado, pero el llamante puede autenticarse implícitamente por su aptitud para establecer el canal de señalización de llamada con los parámetros de sesión TLS correctos. La figura H.1 ilustra el consiguiente flujo de información: se utiliza RAS para negociar la clave maestra de sesión, el ID de sesión y el correspondiente secreto pre-maestro se distribuyen al soporte lógico de la TLS, y la capa de señalización de llamada utiliza el ID de sesión para establecer el canal de señalización de llamada por TLS. El medio por el cual se realiza la transferencia del secreto depende de la implementación y está fuera del ámbito de esta Recomendación. Obsérvese que esta Recomendación especifica el puerto 1300 como el puerto de escucha TLS por defecto para la señalización de llamada. En cambio, el punto extremo tiene que utilizar una de las direcciones de transporte de señalización de llamada suministradas por el controlador de acceso.

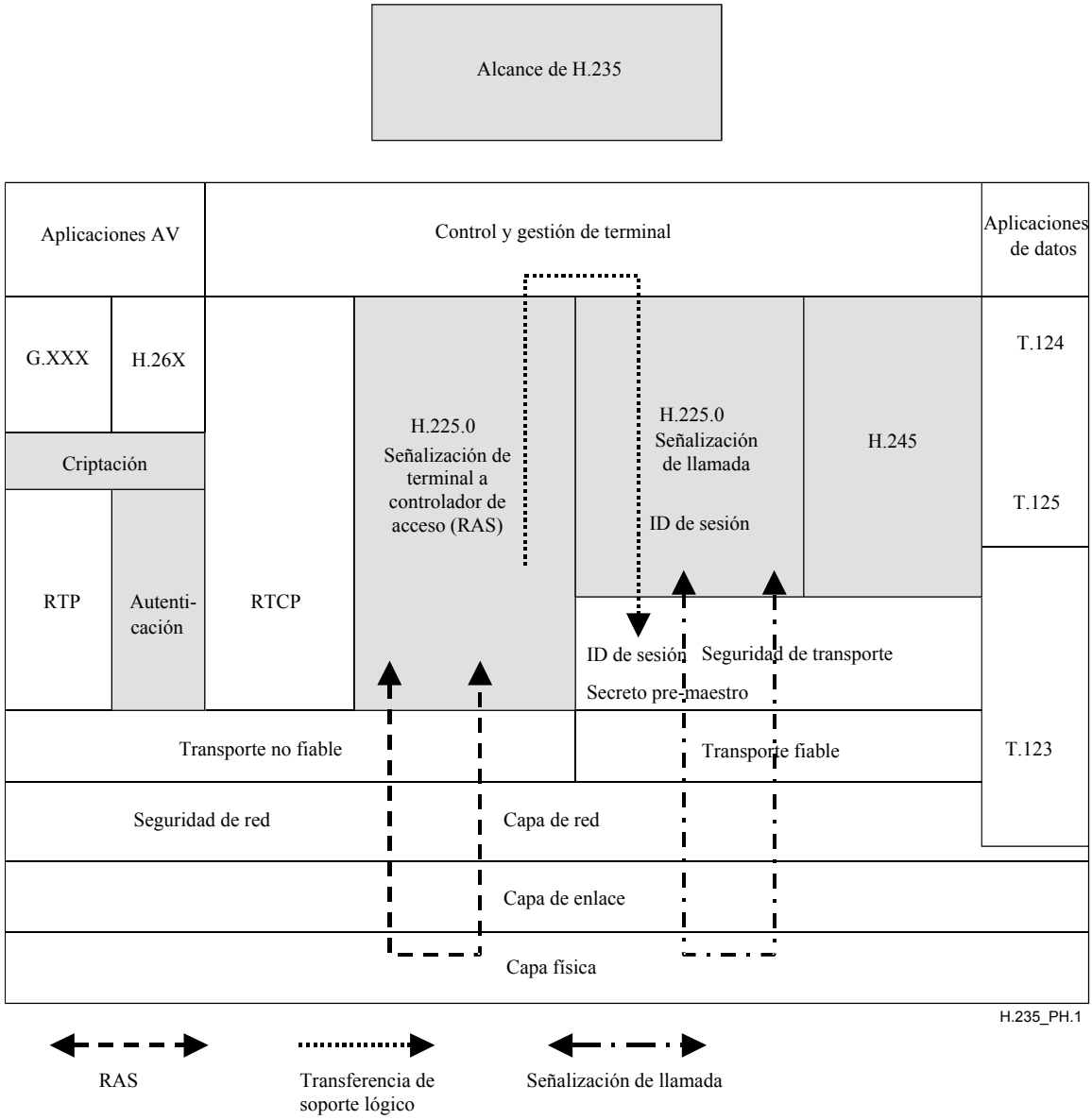

Figura H.1/H.235 – Flujo de información para perfil de seguridad y TLS

En la siguiente descripción se hace referencia a los pasos dentro del marco básico antes mencionado.

H.8.1.1 Registro de punto extremo

Un punto extremo puede probar la aptitud de un controlador de acceso para soportar la señalización de llamada con protección TLS incluyendo uno o más elementos **cipherSuite** y uno o más elementos **compression** en el **ClearToken** de perfil en el mensaje GRQ enviado en el paso 1, antes citado. Si el punto extremo desea utilizar una sesión negociada previamente, incluirá también el **sessionID** en el **ClearToken** (y especificará solamente la sucesión cifrada simple y el método de compresión simple que concuerdan con la sesión solicitada). Si la negociación ha de basarse en una sesión TLS existente, no se requiere material criptográfico en el **ClearToken de perfil**, salvo **nonce**.

Si la sesión que se solicita no existe, el controlador de acceso seleccionará otro perfil de autenticación (si se ha ofrecido) o devolverá un GRJ con **GatekeeperRejectReason.resourceUnavailable**. Si la sesión solicitada no existe, el material de

clave maestra se obtiene de la sesión TLS, y se utiliza (junto con **random** tomado de la GRQ y **random2** generado por el controlador de acceso) para calcular la clave de autenticación con miras al intercambio RAS. El **sessionID**, la **cipherSuite**, el método **compress**, y el **nonce** del controlador de acceso se devolverán en el **ClearToken** de perfil de una GCF.

Si el controlador de acceso puede soportar la negociación de sesión TLS, calculará el material de clave maestra como se especifica por el perfil, asignará un nuevo ID de sesión y lo devolverá en el **ClearToken** de perfil, en el **sessionID**. El **ClearToken** de perfil contendrá también los parámetros de seguridad requeridos del paso 2 anterior, una sola **cipherSuite** seleccionada, un solo método **compress** seleccionado, y el **sessionID** diferente de cero. Obsérvese que el método de intercambio de claves de la sucesión cifrada seleccionada es intrascendente. Si el controlador de acceso está de acuerdo con la protección TLS de una señalización de llamada, todas las direcciones de transporte de señalización de llamada intercambiadas en los subsiguientes mensajes RRQ/RCF o ARQ/ACF serán habilitadas en cuanto a la TLS.

Si la negociación TLS y/o el encaminamiento por el controlador de acceso no son soportados por el controlador de acceso, no se devolverá ningún parámetro TLS, pero los procedimientos de autenticación podrán continuar a partir del paso 3, como se ha descrito antes. El punto extremo decidirá si está preparado para proseguir sin la protección TLS de la señalización de llamada; puede optar por hacer esto y, además, utilizar el perfil de autenticación. Tras una finalización exitosa de la secuencia de registro, la sesión TLS está disponible para uso en establecimiento rápido de una o más conexiones de señalización al controlador de acceso, sin necesidad de renegociar material de clave mediante métodos de clave pública.

Las sesiones TLS tienen tiempos de vida finitos. Por tanto, puede que un punto extremo tenga que renegociar parámetros de sesión y obtener un nuevo ID de sesión. Esto puede conseguirse intercambiado los elementos **ClearToken** necesarios, como se ha descrito antes, en una secuencia de registro de tipo peso ligero ("mantenerse vivo"). Tal secuencia no afectará a la clave de autenticación RAS.

H.8.2 Utilización de certificados para autenticar al controlador de acceso

Aunque puede no ser viable en la práctica intercambiar cadenas de certificados verificables en RAS (debido a limitaciones del tamaño de los paquetes UDP), es posible hacer que un servidor se identifique a sí mismo ante el punto extremo si el punto extremo puede obtener una copia de confianza de la clave pública del servidor a través de otros medios. El servidor puede simplemente incluir, en el mensaje GCF, un **CryptoH323Token.cryptoGKCert** con el **ClearToken.tokenOID** fijado al OID del perfil de seguridad seleccionado.

H.8.3 Utilización de otros mecanismos de seguridad de la señalización

Los parámetros negociados como parte de un perfil de seguridad en el contexto de este anexo pueden emplearse en mecanismos de seguridad en el nivel de transporte y/o de aplicación como esté determinado por el perfil específico. La secuencia **profileInfo** añadida al **ClearToken** H.235 ha sido prevista para tal utilización, si fuera necesario.

H.9 Amenazas (informativo)

H.9.1 Ataque pasivo

En el momento actual, el esquema antes descrito no es vulnerable al ataque pasivo, siempre que la negociación Diffie-Hellman tampoco lo sea.

H.9.2 Ataques por denegación de servicio

Este esquema es susceptible de un ataque, activo, por denegación de servicio, en el cual una parte responde a la GRQ con un GRJ espurio. Este tipo de ataque puede o no ser identificable: si el controlador de acceso es legítimo, y conoce el secreto compartido (por ejemplo, el controlador de

acceso es el controlador de acceso del punto extremo y la **rejectReason** es **resourceUnavailable**), entonces el controlador de acceso podría completar la negociación de la clave y autenticar el GRJ devolviendo, en el GRJ, los mismos elementos descritos para la GCF (salvo que el OID devuelto en **authenticationMode** de la GCF sería devuelto en un elemento **ClearToken.profileInfo** del GRJ). Esto se deja como parte de la definición de un perfil específico.

Si el GRJ no está autenticado, podría provenir de un atacante. Antes de reaccionar al GRJ (por ejemplo, buscando otro posible controlador de acceso), el punto extremo debe esperar la posible recepción de otro GRJ o una GCF autenticada de otro controlador de acceso. En otro caso, el punto extremo debe probar con cada controlador de acceso sugerido en cualquier **altGKInfo** recibida en todos los GRJ (uno de los cuales, cabe suponer, es legítimo). De todas formas, sólo el controlador de acceso apropiado (que conoce el secreto compartido) puede devolver una GCF autenticada.

H.9.3 Ataques de hombre-en-el-medio [ataques MIM (*man-in-the-middle*)]

Existe una inclinación a considerar el intercambio como un intercambio de clave Diffie-Hellman no criptada, seguido del uso de la contraseña o PIN para, a partir del secreto Diffie-Hellman, derivar claves de sesión. Sin embargo, esta forma de intercambio es vulnerable al ataque de hombre-en-el-medio (MIM), que puede utilizarse para descubrir el "pequeño" secreto compartido, por el método conocido por fuerza bruta, utilizando el valor de verificación de integridad proporcionado por el controlador de acceso legítimo en el mensaje GCF.

Desde luego, cualquier MIM puede manipular cualquier mensaje RAS autenticado para asegurarse de que el mensaje será descartado debido un fallo de la verificación de integridad. Si todos los mensajes pueden ser manipulados, el servicio puede ser denegado.

H.9.4 Ataques por intentos de adivinar

Un atacante puede hacerse pasar por un punto extremo legítimo, por un controlador de acceso legítimo, o por los dos (hombre-en-el-medio), y tratar de adivinar el secreto compartido por el método de intentos-fracasos sucesivos. Por ejemplo, el atacante (que se supone que conozca los detalles del perfil de autenticación, pero no el secreto compartido) puede adivinar un secreto compartido y tratar de obtener el registro enviando una GRQ utilizando esta información conseguida por los mencionados intentos de adivinar. En general, el controlador de acceso responde a este intento con una GCF que contiene la clave pública del controlador de acceso (criptada utilizando el secreto compartido real), y un ICV calculado utilizando la clave derivada que depende de la descriptación, por el controlador de acceso, de la clave pública criptada del atacante. El atacante puede utilizar esta información para verificar su intento de adivinar el secreto compartido. Si el intento de adivinar confirma el ICV de la GCF, entonces probablemente es igual al secreto compartido real; esto puede confirmarse continuando con la secuencia de registro. Si el intento de adivinar no es apropiado para reproducir el ICV de la GCF, el atacante tiene que hacer un nuevo intento de adivinar. Cuando el espacio de las claves para el secreto compartido es pequeño, el número de intentos de adivinar que hay que realizar para una búsqueda de tipo fuerza bruta puede no ser prohibitivo. Este ataque requiere la participación activa del controlador de acceso (o del punto extremo si el atacante se está haciendo pasar por el controlador de acceso). El método tradicional para contrarrestar tal ataque es vigilar los intentos infructuosos, contarlos y, cuando su número alcanza cierto umbral, tratar todos los intentos subsiguientes como no válidos (al menos durante un periodo especificado) y señalizar una alarma, pero esos procedimientos dependen de la implementación.

H.9.5 Media clave del controlador de acceso no criptada

Como se ha expresado antes, el intercambio EKE puede mantenerse securizado, en ciertas condiciones, si el controlador de acceso respondedor no cripta su media clave Diffie-Hellman. En particular, el controlador de acceso tiene que ser la primera parte que demuestre su conocimiento del secreto compartido (PIN) mediante el ICV. Si no se da este caso, el controlador de acceso (o un

intruso que se hiciera pasar por el controlador de acceso) podría, simplemente, realizar intentos con todos los PIN posibles para descriptar la media clave D-H del punto extremo, calcular el secreto compartido D-H resultante, derivar la clave de autenticación, y cotejarla con el ICV suministrado por el punto extremo. Esto no es posible si el punto extremo puede verificar el ICV suministrado por el controlador de acceso, primero, y rechazar la continuación del registro si el ICV no es el esperado.

La utilización de una media clave no criptada ofrece ventajas a un controlador de acceso ya que éste puede reutilizar su correspondiente clave privada con múltiples puntos extremos. Esto no sería posible si la misma clave se distribuyera criptada para múltiples secretos compartidos o PIN. Un tercero podría colectar ejemplos de la media clave criptada, por ejemplo, según dos PIN diferentes, después de lo cual podría recorrer todas las combinaciones posibles de dos PIN para determinar qué pareja produjo la misma media clave al efectuarse la decriptación. Si hay, por ejemplo, 10^8 PIN posibles, habrá que tratar solamente 10^{16} combinaciones posibles. Este es un problema equivalente al que se plantea en la búsqueda de un número aleatorio de 54 bits, que no es insoluble en lo absoluto. Incluso si se encontrara más de una solución posible, la correcta podría determinarse rápidamente efectuando una tercera observación.

Anexo I

Soporte de llamadas con encaminamiento directo

...

I.5 Símbolos y abreviaturas

En este anexo se usan las siguientes siglas.

$\underline{ENC_{K;\,S,\,IV}(}\{M\}\underline{)}_{\sim\sim K;\ S,\ IV\sim\sim}$ 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

$K_{BH\sim\sim G\sim\sim}$ Secreto compartido (anexo D, anexo F) entre el punto extremo B y el controlador de acceso H~~G~~

$\underline{K_{GH}}$ Secreto compartido (anexo D, anexo F) entre el controlador de acceso G y el controlador de acceso H

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

$KS_{BH\sim\sim G\sim\sim}$ Clave adicional compartida secreta entre el punto extremo B y el controlador de acceso H~~G~~

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

$\underline{E}K'_{B\underline{H}\sout{G}}$ La clave de criptación compartida entre el punto extremo B y el controlador de acceso $\underline{H}\sout{G}$

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

I.6 Referencias normativas

...

– Recomendación UIT-T H.235 anexo F (200$\underline{3}\sout{2}$), Corrigéndum 1 a *Perfil de seguridad híbrido.*

...

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 $\sout{\text{solo }}$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 su$\sout{\text{el}}$ 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 del punto extremo iniciador puede proporcionar un secreto compartido para los puntos extremos que se comunican directamente utilizando un modelo del tipo Kerberos.

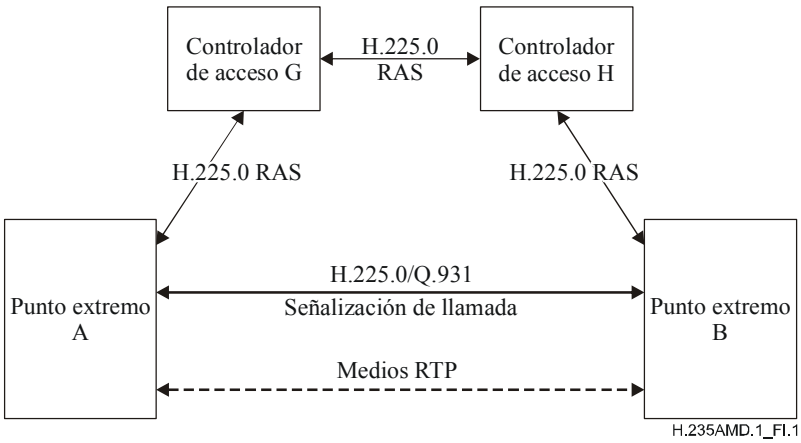

Figura I.1/H.235 – Caso de una llamada con encaminamiento directo

I.8 Limitaciones

Este anexo no trata los escenarios con encaminamiento directo en los que no haya ningún controlador de acceso$\sout{\text{En su versión actual, este anexo no trata los casos de encaminamiento directo}}$

~~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 o H 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.

Este procedimiento comprende tanto el caso de un solo controlador de acceso, común a los dos puntos extremos, como el caso de múltiples controladores de acceso, en cadena. Cuando intervienen múltiples controladores de acceso, el controlador de acceso G – en cuya zona se origina la llamada – debe localizar al controlador de acceso H utilizando el mecanismo **LRQ** (multidifusión) como se describe en 8.1.6/H.323 "Señalización facultativa de punto extremo llamado". La comunicación entre dos controladores de acceso será securizada de acuerdo con el anexo D. Para esto, se supone que está disponible un secreto compartido común K_{GH}. Puesto que **LRQ** entre los controladores de acceso es típicamente un mensaje multidifusión, el secreto compartido K_{GH} no puede usualmente ser un secreto compartido por parejas, sino que se supone que es en realidad un secreto compartido en base a un grupo dentro de la nube potencial de controladores de acceso.

NOTA – Este supuesto limita la escalabilidad en el caso general y no permite la autenticación de la fuente. Sin embargo, se cree que en redes pertenecientes a compañías con un número pequeño, limitado, de controladores de acceso, esa restricción y las limitaciones de seguridad, son aún aceptables. La securización de la comunicación multidifusión entre controladores de acceso utilizando firmas digitales podría solventar esas limitaciones; no obstante, esto queda en estudio.

Si el mecanismo **LQR** se utiliza para localizar al controlador de acceso distante, **LQR** transportará un ClearToken separado con **tokenOID** fijado a "I0"; no debe utilizarse ningún otro campo en ese ClearToken. Para el caso multidifusión, el **generalID** en el CryptoToken de **LQR** no se utilizará. La comunicación entre controladores de acceso mediante H.501 y/o H.510 queda en estudio.

EK_{BH} designa la clave de criptación que es compartida entre el punto extremo B y el controlador de acceso H. El controlador de acceso H generará el material de clave de criptación EK_{BH} a partir del secreto compartido K_{BH} utilizando el procedimiento de derivación de claves PRF-**based** (basado) definido en I.10 donde **keyDerivationOID** en **V3KeySyncMaterial** contendrá "AnnexI-HMAC-SHA1-PRF"; véase I.12.

El controlador de acceso H transmitirá la EK_{BH} al controlador de acceso G. El modo de criptación OFB realzada (EOFB) (véase B.2.5) se utilizará con la clave adicional específica del punto extremo, secreta, KS_{GH}. Son algoritmos de criptación aplicables (véase D/11):

- DES (56 bits) en modo EOFB que utiliza OID "Y1": facultativo.
- 3DES (168 bits) en modo EOFB exterior que utiliza OID "Z1": facultativo.
- AES (128 bits) en modo EOFB que utiliza OID "Z2": por defecto y es recomendado.
- RC2-compatible (56 bits) en modo EOFB que utiliza OID "X1": facultativo.

Para el modo de criptación EOFB, el controlador de acceso H generará un valor inicial aleatorio IV. Para OID "X1", OID "Y1" y OID "Z1", el valor inicial aleatorio IV tiene 64 bits y será transportado

dentro de **iv8** de **params** dentro de **V3KeySyncMaterial**; en tanto que el IV tiene 128 bits para OID "Z2" y será transportado dentro de **iv16** de **params** dentro de **V3KeySyncMaterial**.

El controlador de acceso H incluirá $ENC_{K_{GH},\ KS_{GH},\ IV}(EK_{BH})$ en ClearToken CT_{HG} con **tokenOID** fijado a "I3". El texto cifrado obtenido $ENC_{K_{GH},\ KS_{GH},\ IV}(EK_{BH})$ será transportado en la estructura de datos **h235key** como parte de **secureSharedSecret** donde se colocará dentro de **encryptedSessionKey** de la estructura de datos **secureSharedSecret**. El algoritmo de criptación será el indicado en **algorithmOID** ("X1", "Y1", "Z1" o "Z2") dentro de **V3KeySyncMaterial**. La respuesta **LCF** contendrá el ClearToken CT_{HG}.

El controlador de acceso G 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 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 utilicen 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).

EK' indica la clave de criptación compartida entre un punto extremo y su~~el~~ controlador de acceso. Las claves de criptación EK'_{AG} y EK'_{H} 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 $K_{BH~~G~~}$) utilizando el procedimiento de cálculo de clave **basado** en PRF que se define en la cláusula I.10, donde **keyDerivationOID** en **V3KeySyncMaterial** mantendrá "Annex I-HMAC-SHA1-PRF", véase la cláusula I.12.

El controlador de acceso G 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 EK'_{AG} (para un CT destinado al punto extremo A) o por $EK'_{BH~~G~~}$ (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 KS_{AG} resp. KS_{BG}. 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 $\underline{ENC_{EK_{AG},\ KS_{AG},\ IV}(\{K_{AB}\})}$ ~~$\{K'_{AG,\ KS_{AG,\ IV}}\}$~~ resp. $\underline{ENC_{EK_{BG,}\ KS_{BG,\ IV}}(\{K_{AB}\})}$ ~~$\{K'_{BG,\ KS_{BG,\ IV}}\}$~~ será entonces transportado en la estructura de datos **h235key** como parte de **secureSharedSecret**, 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 $\underline{E}K'_{~~A~~G}$ y KS_{AG} como se describe antes para el controlador de acceso G. El punto extremo A descriptará la información **encryptedSessionKey** encontrada dentro de **V3KeySyncMaterial** de CT_A para obtener el $\underline{E}K'_{AB}$.

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!) no se utilizará, a menos que el punto extremo A ya tenga un $EPID_B$ disponible (por ejemplo, mediante configuración o memorizado de una comunicación anterior). Si el punto extremo A utiliza un valor $EPID_B$ para **generalID** en SETUP, el punto extremo A aceptará el valor del **sendersID** en el mensaje de señalización de llamada devuelto, como el verdadero ~~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 $\underline{E}K'_{BG}$ y KS_{BG} como en el caso descrito para el controlador de acceso. El punto extremo B descriptará la información **encryptedSessionKey** encontrada dentro del **V3KeySyncMaterial** del CT_B para obtener $\underline{E}K'_{AB}$.

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.

El punto extremo B incluirá generalID fijado a $EPID_A$ y sendersID fijado a $EPID_B$ para la protección de cualquier mensaje de señalización de llamada H.225.0 destinado a EP A (por ejemplo, Call Proceeding, Alerting o Connect).

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

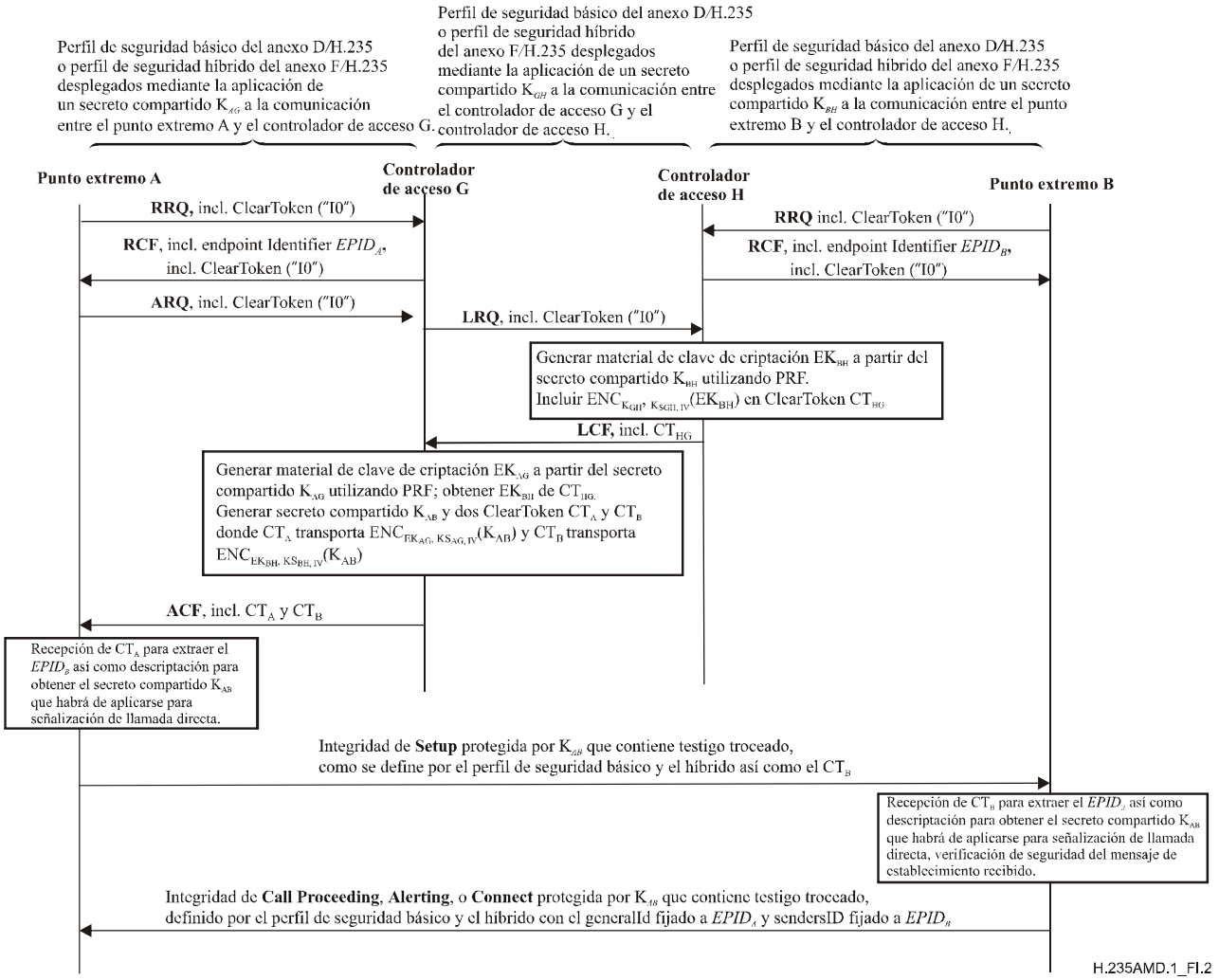

Figura I.2/H.235 – Flujo básico de comunicación

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 EK'_{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 EK'_{BG} se calculará utilizando la PRF con el parámetro *inkey* puesto a K_{BHG} 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_{BHG}.

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_{BHG} se calculará utilizando la PRF con el parámetro *inkey* puesto a K_{BHG} y *label* se pondrá a 0x39A2C14B || **challenge**.

...

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

Referencia de identificador de objeto	Valor de identificador de objeto	Descripción
...	**...**	**...**
"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.
"I3"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 52}	Utilizado en el procedimiento DRC para el tokenOID de ClearToken entre controladores de acceso, que indica que el ClearToken mantiene una clave de criptación para el controlador de acceso iniciador.
...	**...**	**...**

Apéndice I

Detalles de las implementaciones H.323

...

I.4.5 IPSEC

En general IPSEC ([IPSEC], [ESP]) e IKE [IKE] 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 [ISAKMP]/Oakley [RFC 2412] 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 [ISAKMP]/Oakley [RFC 2412] 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 [ISAKMP]/Oakley [RFC 2412] transparente antes de que se transmitan paquetes H.245. La autenticación realizada por esta negociación ISAKMP [ISAKMP]/Oakley [RFC 2412] 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 [ISAKMP]/Oakley [RFC 2412].

5) Después de cada autenticación ISAKMP [ISAKMP]/Oakley [RFC 2412] 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.

...

Apéndice IV

Bibliografía

[Daemon] DAEMON (J.), Cipher and Hash function design, *Ph.D. Thesis, Katholieke Universiteit Leuven*, marzo de 1995.

[ESP] IETF RFC 2406 (1998), *IP Encapsulating Security Payload (ESP)*.

[IKE] IETF RFC 2409 (1998), *The Internet Key Exchange (IKE)*.

[ISAKMP~~PSEC~~] IETF RFC 2408 (1998), ~~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~~.

...

[MIKEY] ARKKO (J.), CARRARA (E.), LINDHOLM (F.), NASLUND (M.), NORRMAN (K.): MIKEY:Multimedia Internet KEYing, *Internet Draft <draft-ietf-msec-mikey-0~~8~~6.txt>*, RFC xxxx, Work in Progress (MSEC WG), IETF, 1~~0~~2/2003.

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

…

[RTP] IETF RFC 3550 (2003), ~~SCHULZRINNE (H.), CASNER (S.), FREDERICK (R.), JACOBSON (V.), RTP:~~ *RTP: A transport Protocol for Real-Time Applications*~~, RFC 3550, *Internet Engineering Task Force*, 2003~~.

…

[SRTP] IETF RFC 3711 (2004), ~~Baugher, McGrew, Oran et al: *The Secure Real-time Transport Protocol (SRTP)*; draft-ietf-avt-srtp-09.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, aspectos del protocolo Internet y Redes de la próxima generación
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación

Impreso en Suiza
Ginebra, 2004