UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T H.235

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Amendement 1
(04/2004)

SÉRIE H: SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

Infrastructure des services audiovisuels – Aspects système

Sécurité et chiffrement pour les terminaux multimédias de la série H (terminaux H.323 et autres terminaux de type H.245)

Amendement 1

Recommandation UIT-T H.235 (2003) – Amendement 1

RECOMMANDATIONS UIT-T DE LA SÉRIE H

SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

CARACTÉRISTIQUES DES SYSTÈMES VISIOPHONIQUES	H.100–H.199
INFRASTRUCTURE DES SERVICES AUDIOVISUELS	
Généralités	H.200–H.219
Multiplexage et synchronisation en transmission	H.220–H.229
Aspects système	**H.230–H.239**
Procédures de communication	H.240–H.259
Codage des images vidéo animées	H.260–H.279
Aspects liés aux systèmes	H.280–H.299
Systèmes et équipements terminaux pour les services audiovisuels	H.300–H.349
Architecture des services d'annuaire pour les services audiovisuels et multimédias	H.350–H.359
Architecture de la qualité de service pour les services audiovisuels et multimédias	H.360–H.369
Services complémentaires en multimédia	H.450–H.499
PROCÉDURES DE MOBILITÉ ET DE COLLABORATION	
Aperçu général de la mobilité et de la collaboration, définitions, protocoles et procédures	H.500–H.509
Mobilité pour les systèmes et services multimédias de la série H	H.510–H.519
Applications et services de collaboration multimédia mobile	H.520–H.529
Sécurité pour les systèmes et services multimédias mobiles	H.530–H.539
Sécurité pour les applications et services de collaboration multimédia mobile	H.540–H.549
Procédures d'interfonctionnement de la mobilité	H.550–H.559
Procédures d'interfonctionnement de collaboration multimédia mobile	H.560–H.569
SERVICES À LARGE BANDE ET MULTIMÉDIAS TRI-SERVICES	
Services multimédias à large bande sur VDSL	H.610–H.619

Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

Recommandation UIT-T H.235

Sécurité et chiffrement pour les terminaux multimédias de la série H (terminaux H.323 et autres terminaux de type H.245)

Amendement 1

Résumé

La version 3 de la Rec. UIT-T H.235, qui remplace la deuxième, définit une procédure applicable aux signaux DTMF chiffrés, des identificateurs d'objet pour l'algorithme de chiffrement AES des charges utiles de médias, le mode de chiffrement amélioré OFB des flux (mode EOFB) pour le chiffrement des flux de médias; elle décrit également une option d'authentification seulement dans l'Annexe D applicable au franchissement des dispositifs NAT/pare-feu, une procédure de distribution des clés sur le canal RAS, des procédures de transport de clé de session mieux sécurisé et des procédures de distribution et de mise à jour de clés de session plus fiables, des procédures permettant de sécuriser des flux de charge utile multiples, une meilleure prise en charge de la sécurité pour les appels acheminés directement (nouvelle Annexe I), des moyens plus souples de signalement des erreurs, des précisions et des améliorations d'efficacité pour la sécurité à démarrage rapide et pour la signalisation Diffie-Hellman avec des paramètres Diffie-Hellman plus longs et introduit des modifications tirées du guide à l'usage des responsables de l'implémentation de la Rec. UIT-T H.323

Le présent amendement à la version 3 de la Rec. UIT-T H.235 complète cette dernière par une nouvelle Annexe H et par de nouvelles fonctionnalités dans l'Annexe I. Les modifications qui sont apportées à la notation ASN.1 pour tenir compte de la nouvelle Annexe H visent à prendre en charge de nouvelles fonctions identifiées par la séquence **profileInfo** du champ ClearToken. Le présent amendement vise également à apporter quelques corrections et mises à jour du texte de la version 3 de la Rec. UIT-T H.235.

Source

L'Amendement 1 de la Recommandation H.235 (2003) de l'UIT-T a été approuvé le 6 avril 2004 par la Commission d'études 16 (2001-2004) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

L'Assemblée mondiale de normalisation des télécommunications (AMNT), qui se réunit tous les quatre ans, détermine les thèmes d'étude à traiter par les Commissions d'études de l'UIT-T, lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution 1 de l'AMNT.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

NOTE

Dans la présente Recommandation, l'expression "Administration" est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue.

Le respect de cette Recommandation se fait à titre volontaire. Cependant, il se peut que la Recommandation contienne certaines dispositions obligatoires (pour assurer, par exemple, l'interopérabilité et l'applicabilité) et considère que la Recommandation est respectée lorsque toutes ces dispositions sont observées. Le futur d'obligation et les autres moyens d'expression de l'obligation comme le verbe "devoir" ainsi que leurs formes négatives servent à énoncer des prescriptions. L'utilisation de ces formes ne signifie pas qu'il est obligatoire de respecter la Recommandation.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT avait été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

TABLE DES MATIÈRES

Recommandation UIT-T H.235

Sécurité et chiffrement pour les terminaux multimédias de la série H (terminaux H.323 et autres terminaux de type H.245)

Amendement 1

…

2 Références normatives

…

– Recommandation UIT-T H.235 (1998), *Sécurité et cryptage des terminaux multimédias de la série H (terminaux H.323 et autres terminaux de type H.245*.

– Recommandation UIT-T H.235 (200~~0~~3), *Sécurité et chiffrement pour les terminaux multimédias de la série H (terminaux H.323 et autres terminaux de type 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 Termes et définitions

…

3.8 **algorithme cryptographique**: fonction mathématique qui calcule un résultat à partir d'une ou de plusieurs valeurs d'entrée.

3.8 *bis* **EC-GDSA**: signature numérique à courbe elliptique avec appendice analogue à l'algorithme de signature numérique NIST (DSA); (voir aussi le chapitre 5 de [ISO/CEI 15946-2].

3.8 *ter* **ECC et EC**: système cryptographique à courbe elliptique (*elliptic curve cryptosystem*) (voir la section 8.7 de "*section ATM forum security specification*" *version 1.1*).

3.8 *quat* **ECKAS-DH**: système de concordance de clés à courbe elliptique – Diffie-Hellman (*elliptic curve key agreement scheme – Diffie-Hellman*). Système de concordance de clés Diffie-Hellman utilisant la cryptographie à courbe elliptique.

3.9 **chiffrement**: processus consistant à rendre des données illisibles par des entités non autorisées après application d'un algorithme cryptographique (ou de chiffrement). Le déchiffrement est l'opération inverse par laquelle le texte chiffré est transformé en texte clair.

4 Symboles et abréviations

La présente Recommandation utilise les abréviations suivantes:

3DES	triple DES
AES	norme de cryptage perfectionné (*advanced encryption algorithm*)
ASN.1	notation de syntaxe abstraite numéro un (*abstract syntax notation No. 1*)

BES	service spécialisé (*back-end server*)
CA	autorité de certification (*certificate authority*)
CBC	chaînage de blocs chiffrants (*cipher block chaining*)
CFB	mode rétroaction de chiffrement (*cipher feedback mode*)
CRL	liste de révocations de certificat (*certificate revocation list*)
CTR	mode compteur (*counter mode*) (voir NIST800-38A)
DES	norme de chiffrement des données (*data encryption standard*)
DH	Diffie-Hellman
DNS	système de dénomination de domaine (*domain name system*)
DSS	norme de signature numérique (*digital signature standard*)
DTMF	multifréquence à deux tonalités (*dual tone multi-frequency*)
ECB	mode dictionnaire (*electronic code book*)
ECC et EC	système cryptographique à courbe elliptique (*elliptic curve cryptosystem*)
EC-GDSA	signature numérique à courbe elliptique avec appendice analogue à l'algorithme de signature numérique NIST (DSA)
ECKAS-DH	système de concordance de clés à courbe elliptique – Diffie-Hellman (*elliptic curve key agreement scheme – Diffie-Hellman*)
EOFB	mode OFB amélioré (*enhanced OFB mode*)
EP	point d'extrémité (*endpoint*)
GCF	confirmation de portier (*gatekeeper confirm*)
GK	portier (*gatekeeper*)
GRJ	refus de portier (*gatekeeper reject*)
GRQ	demande de portier (*gatekeeper request*)
GW	passerelle (*gateway*)
HMAC	code d'authentification de message avec hachage (*hashed message authentication code*)
ICV	valeur de contrôle d'intégrité (*integrity check value*)
ID	identificateur
IPSEC	sécurité du protocole Internet (*Internet protocol security*)
ISAKMP	protocole de gestion des clés d'association Internet (*Internet security association key management protocol*)
IV	vecteur d'initialisation (*initialization vector*)
LCF	confirmation de localisation (*location confirm*)
LDAP	protocole rapide d'accès à l'annuaire (*lightweight directory access protocol*)
LRJ	refus de localisation (*location reject*)
LRQ	demande de localisation (*location request*)
MAC	code d'authentification de message (*message authentication code*)
MCU	unité de commande multipoint (*multipoint control unit*)

MD5	résumé de message n° 5 (*message digest 5*)
MIM	entremetteur (*man-in-the-middle*)
MPS	flux de charge utile multiple (*multiple payload stream*)
NAT	traduction d'adresse de réseau (*network address translation*)
OCSP	protocole de statut de certificat en ligne (*online certificate status protocol*)
OFB	mode à rétroaction de sortie (*output feedback mode*)
OID	identificateur d'objet (*object identifier*)
PDU	unité de données protocolaire (*protocol data unit*)
PIN	numéro d'identification personnel (*personal identification number*)
PKCS	système de chiffrement avec clé publique (*public-key crypto system*)
PKI	infrastructure de clé publique (*public key infrastructure*)
PRF	fonction pseudo-aléatoire (*pseudo-random function*)
QS	qualité de service
RAS	enregistrement, admission et statut (*registration, admissions and status*)
RCF	confirmation d'enregistrement (*registration confirm*)
RRJ	refus d'enregistrement (*registration reject*)
RRQ	demande d'enregistrement (*registration request*)
RSA	algorithme à clé publique de Rivest, Shamir et Adleman (*Rivest, Shamir and Adleman (public key algorithm)*)
RTCP	protocole de commande de transport en temps réel (*real-time transport control protocol*)
RTP	protocole de transport en temps réel (*real-time transport protocol*)
SDU	unité de données de service (*service data unit*)
SHA	algorithme de hachage sécurisé (*secure hash algorithm*)
SHA1	algorithme de hachage sécurisé n° 1 (*secure hash algorithm 1*)
SRTP	protocole de transport sécurisé en temps réel (*secure real-time transport protocol*)
SSL	couche connecteur sécurisé (*secure socket layer*)
TLS	sécurité de la couche de transport (*transport level security*)
TSAP	point d'accès au service de transport (*transport service access point*)
X || Y	concaténation de X et Y
XOR, ⊕	OU exclusif

5 Conventions

…

La présente Recommandation décrit l'utilisation de "n" types de message différents: H.245, RAS, Q.931, etc. Pour établir une distinction entre ces différents types de message, la convention suivante est utilisée: les messages et noms de paramètres H.245 se composent de plusieurs mots concaténés qui sont mis en valeur par un caractère gras (**maximumDelayJitter**); les noms de message RAS sont représentés par des abréviations à trois lettres (**ARQ**); les noms de message Q.931 se composent d'un ou de deux mots dont la première lettre est en majuscule (**Call Proceeding**).

La présente Recommandation utilise la notion consistant à mettre la structure de données ASN.1 composite à NULL; par exemple, "**paramS** mis à NULL" (voir les § 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 et E.13.2). Cela signifie que tous les éléments optionnels dans une structure donnée de type SEQUENCE (c'est-à-dire **Params**) sont absents.

La présente Recommandation définit divers identificateurs d'objet (OID, *object identifier*) destinés à la signalisation des capacités relatives à la sécurité, des procédures et des algorithmes de sécurité. Ces identificateurs renvoient à une arborescence de valeurs attribuées pouvant provenir de sources extérieures ou faisant partie d'une arborescence d'identificateurs d'objet entretenue par l'UIT-T. Les identificateurs d'objet qui sont liées à la Rec. UIT-T H.235 présentent l'aspect suivant:

OID" = {itu-t (0) recommendation (0) h (8) 235 version (0) **V N**} où **V** représente symboliquement un simple chiffre décimal précisant la version correspondante de la Rec. UIT-T H.235; par exemple 1, 2, ou 3. **N** représente symboliquement un nombre identifiant de manière univoque l'instance de l'identificateur d'objet et par conséquent, la procédure associée à l'algorithme ou la capacité de sécurité.

…

6 Introduction au système

6.1 Résumé

1) le canal de signalisation d'appel peut être sécurisé au moyen du protocole TLS ([RFC 2246~~TLS~~], [RFC 3546]) ou IPSEC **(**[RFC 2402~~IPSEC~~], [ESP]) à un accès dont la sûreté est bien établie (Rec. UIT-T H.225.0);

…

6.2 Authentification

…

Une troisième option permet de réaliser l'authentification dans le contexte d'un protocole de sécurité distinct tel que la sécurité TLS ([RFC 2246~~TLS~~], [RFC 3546]) ou IKE~~IPSEC~~ [IKE~~PSEC~~].

Des entités homologues peuvent prendre en charge une authentification aussi bien bidirectionnelle qu'unidirectionnelle. Cette authentification peut se produire sur tout ou partie des voies de communication.

…

6.9 Profils de sécurité

La présente Recommandation comporte un certain nombre d'annexes (Annexes D, E, ~~et~~ F et H) définissant chacune des profils de sécurité H.235. Un profil de sécurité spécifie un usage particulier de fonctionnalités ou sous-ensembles de fonctionnalités H.235 correspondant à des environnements bien définis avec une applicabilité bien délimitée.

…

Annexe A

ASN.1 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
```

...

L'Annexe H suivante est entièrement nouvelle.

Annexe H

Cadre de l'authentification sécurisée dans les communications RAS au moyen de secrets partagés faibles

Résumé

La présente annexe décrit le cadre de l'authentification mutuelle entre participants au cours de l'échange de messages RAS H.225.0. Les méthodes fondées sur la "preuve de la possession" décrites permettent une utilisation sûre des secrets partagés tels que les mots de passe qui, s'ils étaient utilisés en tant que tels, ne garantiraient pas une sécurité suffisante. Sont également décrites les extensions de ce cadre visant à permettre la négociation simultanée des paramètres de sécurité de la couche de transport pour la protection d'un canal de signalisation d'appel ultérieur.

Mots clés

Authentification, mots clés, sécurité.

H.1 Introduction

Dans de nombreuses applications, un point d'extrémité (ou son utilisateur) et son portier ne peuvent échanger qu'un "petit" secret tel qu'un mot de passe ou un "numéro d'identification personnel" (PIN, *personal identification number*). Ce type de secret (ci-après dénommé "mot de passe") et toute clé de chiffrement calculée à partir de celui-ci sont faibles d'un point de vue cryptographique. Les mécanismes d'authentification fondés sur l'épreuve/la réponse, tels que décrits dans le § 10, prévoient des échantillons de textes clairs et de textes chiffrés correspondants, et sont par conséquent exposés à des attaques de type "force brute" de la part d'un observateur de la transaction en question lorsque les authentifications sont effectuées au moyen de simples mots de passe. L'observateur peut ainsi récupérer le mot de passe ou le numéro PIN et se faire ensuite passer pour le point d'extrémité afin d'obtenir un service.

Un ensemble de protocoles classés sous la rubrique générique de l'échange de clés chiffrées utilise un secret partagé pour "occulter" un échange de clés Diffie-Hellman de telle façon que l'attaquant doit résoudre une série de problèmes à logarithmes finis pour valider une attaque de type force brute par rapport au secret partagé. Selon le protocole d'échange de clés chiffrées (EKE, *encrypted key exchange*) de Bellovin et Merritt [B&M], le secret partagé est utilisé pour chiffrer les clés publiques Diffie-Hellman conformément à un algorithme symétrique. Selon la méthode SPEKE de Jablon [Jab], le secret partagé est utilisé pour choisir un générateur différent du groupe Diffie-Hellman. Ces protocoles combinent la sécurité d'un échange efficace de clés Diffie-Hellman avec l'utilisation du secret partagé de telle manière qu'un attaquant ne puisse pas obtenir un texte clair connu en vue d'une utilisation dans le cadre d'une attaque de type force brute à l'encontre du secret sans résoudre le problème de l'algorithme fini Diffie-Hellman. Un des avantages de ces protocoles réside dans le fait qu'ils multiplient les résistances du problème Diffie-Hellman en renforçant l'efficacité du chiffrement de clé secrète (ou vice versa). Un des éventuels inconvénients est qu'ils font généralement l'objet d'une protection par brevet.

H.2 Domaine d'application

La présente annexe peut être utilisée pour tout portier ou point d'extrémité au moyen des protocoles RAS H.225.0.

H.3 Références

H.3.1 Références normatives

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui, de ce fait, en sont partie intégrante. Les versions indiquées étaient en vigueur au

moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée. La référence à un document figurant dans la présente Recommandation ne donne pas à ce document, en tant que tel, le statut d'une Recommandation.

[H.323] Recommandation UIT-T H.323 (2003), *Systèmes de communication multimédia en mode paquet*.

[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 Références informatives

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

[B&M] BELLOVIN (S.), MERRITT (M.), brevet américain 5,241,599, 31 août 1993, initialement attribué à AT&T Bell Laboratories; actuellement attribué à Lucent Technologies.

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

Aucune.

H.5 Abréviations

Se reporter au § 4.

H.6 Cadre de base

H.6.1 Capacités de négociation améliorées H.235v3

La version 3 de la Rec. UIT-T H.235 a été étendue ([H235Amd.1]) afin de prendre en charge le présent cadre de sécurité en y ajoutant l'élément générique qui suit au champ **ClearToken**:

- **profileInfo** est une séquence d'éléments propres à un profil donné, chaque élément étant identifié par sa propre valeur entière telle que définie par son profil dont l'identificateur d'objet (OID) est acheminé dans l'élément **ClearToken.tokenOID**.

Dans les descriptions qui suivent, plusieurs éléments sont acheminés dans la séquence **profileInfo**. Pour faciliter la discussion, on donnera à chacun de ces éléments un nom plutôt qu'une valeur identifiante.

H.6.2 Utilisation entre le point d'extrémité et le portier

Le cadre de base, dans lequel le demandeur est un point d'extrémité souhaitant s'enregistrer auprès d'un portier, et dans lequel le répondant est ce portier, est simple. Dans les paragraphes qui suivent, on part implicitement du principe que chaque profil **ClearToken** mentionné est identifié au moyen de l'élément **tokenOID** du profil d'authentification. Le profil **ClearToken** est censé être étendu. Les

éléments **random** et/ou **random2** peuvent être utilisés par un profil de deux façons différentes: ils peuvent être inclus dans le calcul de la clé d'authentification et/ou dans un profil **ClearToken** dans chaque message RAS ultérieur (par exemple, RRQ/RCF) afin d'éviter les répétitions. L'échange d'enregistrement d'un point d'extrémité se fait de la façon suivante:

1) Le point d'extrémité annonce son intention de participer à un ou plusieurs mécanismes d'authentification ou négociation de clé en incluant le ou les identificateurs d'objet appropriés pour le ou les profils souhaités dans les éléments **authenticationMechanism.keyExch** de l'élément **authenticationCapability** du message **G**atekeeper**ReQ**uest. On suppose que chaque identificateur OID spécifique définit entièrement une procédure d'authentification en fonction du système de clé publique utilisé (par exemple, Diffie-Hellman ou courbe elliptique) et du groupe concerné (par exemple, un des groupes OAKLEY décrits dans la norme RFC 2412), de l'algorithme de chiffrement symétrique (par exemple, AES-128-CBC avec emprunt cryptographique), de la fonction de calcul de clé (par exemple, au moyen de la fonction pseudo-aléatoire décrite dans l'Annexe B), du code d'authentification de message (par exemple, HMAC-SHA1-96) et de la séquence dans laquelle ils sont utilisés. Le point d'extrémité inclut aussi un ou plusieurs profils **ClearToken** dans le message GRQ, chacun acheminant l'identificateur OID pour le profil concerné, ainsi que les données de clé publique (chiffrées nécessaires) de la façon suivante:

   a) **tokenOID** achemine l'identificateur OID du profil, tel qu'il figure dans l'élément **authenticationCapability** du message GRQ encapsulant;

   b) **timeStamp** peut être utilisé pour garantir l'actualité et éviter les répétitions;

   c) **password** ne doit pas être utilisé pour le mot de passe réel;

   d) **dhkey** achemine les paramètres de clé Diffie-Hellman, s'il est utilisé. L'élément **halfkey** inclus est chiffré tel qu'il est spécifié dans le profil choisi;

   e) **challenge** n'est pas requis;

   f) **random** est fourni par le demandeur et est utilisé pour éviter les attaques par répétition;

   g) **certificate** peut être utilisé si l'échange de certificats fait partie du profil;

   h) **generalID** peut être utilisé s'il est requis par le profil;

   i) **eckasdhkey** achemine les paramètres de clé de type courbe elliptique, s'il est utilisé dans le profil. L'élément **public-key** inclus devrait être chiffré, tel qu'il est spécifié par le profil;

   j) **sendersID** peut être utilisé tel qu'il est spécifié par le profil;

   k) un élément **profileInfo**, **initVect**, peut être fourni avec les données (chiffrées) de clé publique (**dhkey** ou **eckasdhkey**) si le profil exige un vecteur d'initialisation pour le déchiffrement;

   l) s'il souhaite utiliser des données de clé, obtenues d'un échange antérieur, le demandeur doit inclure un élément **profileInfo**, appelé **sessionID**, contenant l'identificateur attribué au cours de l'échange antérieur. Dans ce cas, **dhkey**, **eckasdhkey** et/ou **initVect** ne devraient pas être inclus;

   m) s'il souhaite établir une session TLS pour une connexion de signalisation d'appel, le demandeur peut inclure un ou plusieurs éléments **profileInfo** contenant des systèmes cryptographiques TLS. Le message ne doit contenir qu'un système cryptographique (celui négocié préalablement) si l'élément **sessionID** est présent;

   n) s'il souhaite établir une session TLS pour la signalisation d'appel, le demandeur peut inclure un élément **profileInfo** contenant une liste de méthodes de compression; une seule méthode de compression (celle négociée préalablement) doit être incluse si l'élément **sessionID** est présent;

o) plusieurs éléments **profileInfo** peuvent être utilisés pour tout paramètre additionnel requis pour les procédures relevant du profil.

2) Lorsqu'il reçoit le message GRQ, le portier choisit un profil **AuthenticationMechanism** parmi la liste proposée, génère une clé privée appropriée, calcule la clé publique correspondante, génère si nécessaire un vecteur d'initialisation pour le chiffrement symétrique au moyen du mot de passe, chiffre la clé publique, génère un identificateur de session unique, et génère enfin une quantité aléatoire, toutes ces entités étant codées dans un profil **ClearToken**. En fonction du profil concerné, on utilisera les éléments ClearToken suivants:

a) **tokenOID** achemine l'identificateur OID du profil, tel qu'il a été choisi à partir de l'élément **authenticationMethod** du message GCF encapsulant;

b) **timeStamp** peut être utilisé pour garantir l'actualité et éviter les répétitions;

c) **password** ne doit pas être utilisé pour le mot de passe réel;

d) **dhkey** achemine les paramètres de clé Diffie-Hellman, s'il est utilisé. L'élément **halfkey** inclus est chiffré tel qu'il est spécifié par le profil choisi;

e) **challenge** est utilisé pour acheminer un vecteur d'initialisation, s'il est requis pour le chiffrement de clé tel qu'il est spécifié par le profil, ou il peut être utilisé pour acheminer une chaîne aléatoire que le point d'extrémité doit renvoyer pour éviter les attaques par répétition;

f) **random** peut contenir la valeur unique imprévisible fournie par le demandeur pour éviter les attaques par répétition;

g) **certificate** peut être utilisé si l'échange de certificats fait partie du profil;

h) **generalID** peut être utilisé s'il est requis par le profil;

i) **eckasdhkey** achemine les paramètres de clé de type courbe elliptique, s'il est utilisé par le profil. L'élément **public-key** inclus devrait être chiffré tel qu'il est spécifié par le profil;

j) **sendersID** peut être utilisé tel qu'il est spécifié par le profil;

k) **random** (ou un élément **profileInfo** additionnel, appelé **random2**, si le profil exige que les deux nombres aléatoires restent lors de l'échange de message) devrait contenir une valeur unique imprévisible fournie par le répondant pour éviter les attaques par répétition;

l) **initVect** est fourni avec les données relatives à la clé publique (chiffrée) (**dhkey** ou **eckasdhkey**) si le profil exige un vecteur de signalisation pour le déchiffrement;

m) **sessionID** est un identificateur unique (pour le portier) servant à identifier cette session d'enregistrement. Dans certains profils, il peut aussi servir d'identificateur de session TLS pour l'établissement rapide d'un canal de signalisation d'appel protégé par TLS;

n) **profileInfo** peut être utilisé pour tout paramètre additionnel requis pour les procédures relevant du profil.

Le portier calcule ensuite le secret partagé ou la clé maîtresse au moyen de sa clé privée et de la clé publique (déchiffrée) à partir du message GCF, et déduit à partir de la clé maîtresse les clés de chiffrement, les clés d'authentification ou les autres données nécessaires, conformément au profil. Le profil **ClearToken** décrit ci-dessus est placé dans le message **G**atekeeper**C**on**F**irm. On doit contrôler l'intégrité du message GCF et l'authentifier au moyen de la clé d'authentification calculée, puis l'envoyer au point d'extrémité. Le résultat de l'authentification/du contrôle d'intégrité peut être renvoyé de

différentes façons, tel qu'il est spécifié par le profil: au moyen d'un élément **profileInfo** propre au profil, ou au moyen d'une des procédures spécifiées dans l'Annexe D.

3) Le point d'extrémité examine l'élément **authenticationMechanism.keyExch** sélectionné à partir du message GCF et extrait les paramètres du profil **ClearToken** identifié par l'identificateur **tokenOID** correspondant. Le point d'extrémité choisit ensuite sa clé privée, calcule la clé publique correspondante et choisit tout autre paramètre requis par le profil. Il calcule ensuite le secret partagé ou la clé maîtresse au moyen de sa clé privée et de la clé publique (déchiffrée) à partir du message GCF, et en déduit les clés de chiffrement, les clés d'authentification ou les autres données nécessaires, conformément au profil. Le point d'extrémité doit ensuite vérifier l'intégrité du message GCF. Si son intégrité se vérifie, le point d'extrémité doit refuser le message GCF ainsi que toutes les données de clé qui en résultent, et continuer à attendre un message GRQ valable. La reprise RAS standard entraînera la retransmission du message GRQ et, probablement, la réception d'un message GCF intact. Si plusieurs retransmissions ne permettent pas d'obtenir une réponse correcte, le point d'extrémité devrait renoncer à s'enregistrer et informer son utilisateur du problème. Il est à noter que chaque message GRQ envoyé donne à un usurpateur de passerelle une chance supplémentaire de deviner le mot de passe d'un utilisateur et de voir son choix validé sur acceptation du message GRQ. Si l'intégrité du message GCF se vérifie, le point d'extrémité a validé le portier et peut s'enregistrer, et, au cours du processus, s'authentifier auprès du portier.

4) Le point d'extrémité renseigne ensuite un champ **ClearToken** au moyen de l'élément **tokenOID** du profil selon une méthode analogue à celle appliquée par le portier, telle que décrite ci-dessus. Tout champ du jeton en clair du message GCF, qui est considéré comme une épreuve par le profil devraient figurer dans le champ **ClearToken**. S'il est spécifié dans le profil que les répétitions doivent être évitées, le champ **ClearToken** doit comprendre les éléments **random** et **random2** du message GCF reçu (voir ci-dessus). Le champ **ClearToken** est ensuite placé dans un message de demande d'enregistrement à renvoyer au portier. Le point d'extrémité devrait ensuite authentifier le message RRQ entier et l'envoyer au portier. A partir de là, le point d'extrémité ne devrait ni accepter ni envoyer de messages RAS qui ne soient pas authentifiés par le profil convenu au moyen de la clé d'authentification calculée à partir des données de clé partagée.

5) Le portier reçoit le message RRQ et doit utiliser les données de clé partagée pour vérifier l'intégrité du message RRQ par rapport à l'élément inclus d'authentification et de contrôle de l'intégrité. Si le résultat du contrôle d'intégrité est incorrect, le portier doit ignorer le message RRQ reçu et attendre un message RRQ valable. Si aucun message de ce type n'arrive, le point d'extrémité abandonnera finalement toute tentative d'enregistrement et repartira à la recherche d'un portier. Si le contrôle d'intégrité est positif, le portier préparera un message de confirmation d'enregistrement à renvoyer au point d'extrémité. Selon le profil utilisé, ce message RCF peut contenir un champ **ClearToken** qui inclut les éléments **random**, **random2**, et/ou **challenge** du profil d'authentification **ClearToken** fourni dans le message RRQ. Le message RCF ainsi que tous les messages RAS ultérieurs doivent contenir un élément d'authentification et de contrôle de l'intégrité calculé au moyen de la clé et de l'algorithme d'authentification négociés.

6) Lorsqu'il reçoit le message RCF, le point d'extrémité vérifie son intégrité au moyen de l'élément d'authentification et de contrôle de l'intégrité, qui est inclus. Si son intégrité se vérifie, le message RCF doit être refusé; si aucun message RCF valable n'est reçu, même après retransmission du message RRQ, la session doit être abandonnée et le point d'extrémité doit repartir à la recherche d'un nouveau portier. Si l'intégrité du message RCF se vérifie, l'identificateur de session et le système cryptographique choisi peuvent, s'ils sont présents, être extraits du champ **ClearToken** du message en vue d'une utilisation ultérieure lors de l'établissement d'un canal sûr de signalisation d'appel.

H.6.3 Utilisation de profils entre portiers

Pratiquement la même procédure peut être utilisée entre portiers dans un échange de messages LRQ/LCF. Dans ce cas, aucune sélection explicite de profil n'est possible; le portier d'origine doit offrir un ou plusieurs profils en incluant le ou les champs **ClearToken** appropriés tels que décrits ci-dessus pour le message GRQ. Le portier répondant peut choisir un profil proposé et devrait renvoyer le champ **ClearToken** correspondant tel que décrit ci-dessus pour le message GCF. Il est à noter que dans ce cas, le portier appelant ne s'authentifie pas auprès du portier répondant tant qu'il n'établit pas un canal de signalisation d'appel vers le portier.

Cette procédure peut être employée dans un mode multidiffusion si un groupe de portiers partage un seul secret à utiliser à cette fin. Le message LRQ multidiffusion sera fondé sur ce secret; les portiers qui répondent au moyen d'un message LCF utiliseront cette clé pour décoder la clé Diffie-Hellman offerte et choisiront chacun leur propre **nonce** et clé privé Diffie-Hellman pour leur réponse. Les clés de session résultantes seront propres à la paire finale de portiers.

H.6.4 Chiffrement et authentification des canaux de signalisation

Si le routage de portier est pris en charge par le portier, les données de clé nouvellement négociées ainsi que les paramètres cryptographiques identifiés peuvent être utilisés pour authentifier et protéger le canal de signalisation d'appel (par exemple, en établissant une session TLS pour la signalisation d'appel). Si une session TLS doit être utilisée, le portier doit inclure dans le profil **ClearToken** renvoyé les éléments **cipherSuite** et **compress** sélectionnés.

H.7 Profil de sécurité spécifique (SP1)

Le présent paragraphe décrit un profil de sécurité standard qui devrait offrir un secret partagé correspondant à un nombre aléatoire de 80 bits (voir [NIST800-57]). Ce profil a la forme suivante:

- l'identificateur d'objet pour ce profil (appelé "SP1") sera {itu-t (0) recommendation (0) h (8) 235 version (0) 3 60};
- négociation de clé maîtresse, K_m: échange de clés Diffie-Hellman au moyen du groupe 2 bien établi OAKLEY [RFC 2412], suivi de la réduction de hachage SHA1 du secret Diffie-Hellman secret: K_m = SHA1(secret partagé Diffie-Hellman);
- algorithme de chiffrement symétrique: doit être conforme à la norme AES-128 en mode de compteur segmenté avec un discriminateur de participant de 2 octets, D, un vecteur d'initialisation de 124 octets, IV, et un champ de compteur de 2 octets, C, tel que ce compteur soit égal à = D || IV || C, et C = 0 à l'origine. Voir [NIST800-38A] pour une description du mode CTR. Le discriminateur de participant, D, est mis à 0x3636 lorsque le vecteur IV est généré par le participant qui a envoyé le message GRQ/RRQ ou LRQ, et est mis à 0x5c5c lorsque le vecteur IV est généré par le participant qui a répondu au moyen d'un message GCF/RCF ou LCF. Chaque participant doit s'assurer que chaque vecteur IV qu'il génère est unique; pour cela, il peut utiliser sa propre méthode;
- chiffrement de clé Diffie-Hellman: doit utiliser le mode de compteur segmenté AES-128 pour chiffrer la clé publique Diffie-Hellman (représentée par une chaîne d'octets selon l'ordre des octets dans le réseau); le vecteur d'initialisation doit être acheminé dans le champ **ClearToken.initVect** et la clé de 16 octets, K_p, doit être construite sous la forme des 128 bits de plus fort poids du hachage SHA1 du mot de passe utilisateur: K_p = Trunc(SHA1(mot de passe utilisateur), 16), où Trunc(x,y) tronque la chaîne d'octets x à y octets. Il est à noter que cette clé est généralement considérée comme étant une clé faible;
- prévention des répétitions: chaque participant doit indiquer un nombre "aléatoire" de 32 bits (qui peut contenir un champ de compteur garantissant l'unicité); les nombres aléatoires sont utilisés explicitement pour le calcul des clés; par conséquent, il ne faut les transmettre chacune qu'une seule fois;

- calcul de la clé d'authentification, K_a: au moyen de la fonction pseudo-aléatoire (PRF, *pseudo-random function*), décrite dans l'Annexe B, dénommée PRF (*in_key*, *label*, *outkey_len*) avec *in_key* = K_m, et *label* = "auth_key" || R_e || R_g, où R_e est un **nonce** obtenu à partir d'un élément **ProfileElement** du message GRQ, et où R_g est un **nonce** obtenu à partir d'un élément **ProfileElement** du message GCF, et où *outkey_len* = 128;
- fonction d'authentification et d'intégrité de message: au moyen d'un champ **ClearToken** dont l'identificateur **tokenOID** est mis à "SP1" et un élément **ProfileElement.octets** mis à la valeur de hachage HMAC-SHA1-96 calculée sur le message entier tel que décrit dans la Rec. UIT-T H.225.0; cette procédure doit s'appliquer à tous les messages RAS et de signalisation d'appel (sauf le message GRQ ou LRQ, qui ne contient pas d'identificateur **sessionID**);
- clé de chiffrement d'élément, K_e: certains éléments de messages de signalisation d'appel (ou éléments tunnélisés dans ceux-ci) peuvent être chiffrés conformément à la norme AES-128 dans un mode de compteur segmenté au moyen de la clé K_e = PRF(K_m, "encrypt_key" || R_e || R_g, 128). Par exemple, cette clé peut servir à chiffrer des clés de session de média en vue de leur distribution dans des éléments **h235Key** comme en mode de connexion rapide et/ou H.245. Dans ce cas, "SP1" est utilisé comme identificateur OID d'algorithme de chiffrement.

Ce profil utilise des éléments **ProfileElement** définis dans le Tableau H.1. Ces éléments sont acheminés dans la séquence d'éléments **ClearToken.profileInfo** telle que définie dans l'Amendement 1 de l'Annexe A/H.235.

Tableau H.1/H.235 – Eléments de profil

Nom de l'élément (utilisé dans le texte)	Valeur de l'identificateur ElementID	Elément choisi (longueur)	Description de l'élément
initVect	1	**Octets (12)**	vecteur d'initialisation pour chiffrement EKE
nonce	2	**Octets (quelconques)**	valeur unique et imprévisible
cipherSuite	3	**Octets (2)**	système cryptographique TLS
compression	4	**Octets (1)**	algorithme de compression TLS
sessionID	5	**Octets (1..)**	élément unique pouvant correspondre à un identificateur de session TLS
integrityCheck	6	**Octets (12)**	valeur de contrôle chiffrée

La séquence d'enregistrement doit avoir la forme suivante:

– le point d'extrémité doit envoyer le message GRQ avec l'élément **authenticationCapability** qui comporte un élément **AuthenticationMechanism.keyExch** contenant l'identificateur OID "SP1", un champ **ClearToken** correspondant dont l'identificateur **tokenID** = "SP1", la clé **dhkey** contenant une clé publique de 1 024 bits chiffrée à partir du vecteur **initVect** en tant que vecteur IV et en clé calculée à partir du mot de passe utilisateur, et **nonce** = nombre aléatoire de 32 bits choisi par le point d'extrémité;

– le portier doit répondre au moyen d'un message GCF avec l'élément **authenticationMode** égal à un élément **AuthenticationMechanism.keyExch** contenant l'identificateur OID "SP1", un champ **ClearToken** dont l'identificateur **tokenID** = "SP1", une clé **dhkey** contenant une clé publique non chiffrée de 1 024 bits, **nonce** = nombre aléatoire de 32 bits choisi par le portier, et un élément **integrityCheck** contenant la valeur de hachage

d'authentification calculée au moyen de la clé d'authentification calculée, K_a. Il est à noter qu'il n'est pas nécessaire pour le portier de chiffrer sa demi-clé Diffie-Hellman dans le message GCF dans ce profil car il s'agit du premier participant à s'authentifier en montrant sa capacité à authentifier le message GCF au moyen de la clé d'authentification calculée. Ce mode permet au portier de réutiliser ses clés Diffie-Hellman avec plusieurs points d'extrémité. Voir le § H.9.5;

– le point d'extrémité doit répondre au moyen d'un message RRQ contenant la valeur d'authentification et de contrôle d'intégrité dans un élément **ProfileElement** dont l'identificateur **elementID** est mis à **integrityCheck**, et dont l'élément **element** est mis à la valeur calculée à partir de la clé d'authentification déduite, K_a;

– on doit authentifier et contrôler l'intégrité des messages RAS ultérieurs, y compris le message RCF, au moyen de la même procédure et de la même clé. Les messages de signalisation d'appel H.225.0 (et les messages H.245 tunnélisés, s'ils sont présents) doivent être authentifiés au moyen d'un champ **ClearToken**, dont l'identificateur **tokenOID** est mis à "SP1", et contenant une séquence **profileInfo ProfileElement** dont l'élément **elementID** est mis à **integrityCheck** et dont l'élément **element** est mis à la valeur calculée.

– la clé de chiffrement, K_e, ainsi que l'algorithme de chiffrement AES-128 dans le mode de compteur segmenté, peuvent être utilisés par le portier et par le point d'extrémité pour chiffrer certaines informations qui sont transportées selon le protocole RAS, le protocole de signalisation d'appel et/ou le protocole H.245. Par exemple, le portier peut distribuer des clés de chiffrement de média protégées par la clé K_e et l'algorithme de chiffrement de profil;

– s'il doit se réenregistrer et garder l'identificateur de session et le secret maître originaux, le point d'extrémité devrait tenter de le faire en incluant explicitement l'identificateur de session dans le message GRQ (et en n'incluant pas une demi-clé Diffie-Hellman);

– ce profil doit pouvoir être utilisé entre les portiers (voir le § H.6.3).

H.8 Extensions du cadre (à titre indicatif)

Les éléments qui suivent peuvent être incorporés dans un profil de sécurité défini dans le présent cadre.

H.8.1 Utilisation de la clé maîtresse pour protéger le canal de signalisation d'appel via le protocole TLS

Les données de clé, qui sont négociées au cours de l'échange RAS, peuvent aussi servir à calculer des clés de session afin de protéger le canal de signalisation d'appel via le protocole de transport TLS ([RFC 2246], [RFC 3546]). En effet, la négociation RAS remplace le protocole initial de prise de contact TLS. Cela n'est évidemment valable que si la signalisation d'appel est routée par le portier. Cela est particulièrement utile pour l'authentification et la signalisation entre portiers au moyen de l'échange de messages LRQ/LCF. Dans ce cas, il n'existe pas de troisième message RAS par lequel le portier appelant peut s'authentifier auprès du portier appelé à partir des données de clé négociées, mais l'appelant peut être implicitement authentifié par sa capacité à établir le canal de signalisation d'appel au moyen des paramètres corrects de session TLS. La Figure H.1 illustre le flux d'informations en jeu: le protocole RAS est utilisé pour négocier la clé maîtresse de session; l'identificateur de session et la clé-test secrète correspondante sont distribués au logiciel TLS, l'identificateur de session étant utilisé par la couche de signalisation d'appel pour établir le canal de signalisation d'appel via le protocole TLS. Le moyen par lequel le transfert du secret est réalisé dépend de l'implémentation; il n'entre donc pas dans le cadre de la présente Recommandation. Il convient de noter que la présente Recommandation spécifie le port 1300 comme le port d'écoute TLS par défaut pour la signalisation d'appel. Le point d'extrémité doit cependant utiliser une des adresses de transport de signalisation d'appel fournies par le portier.

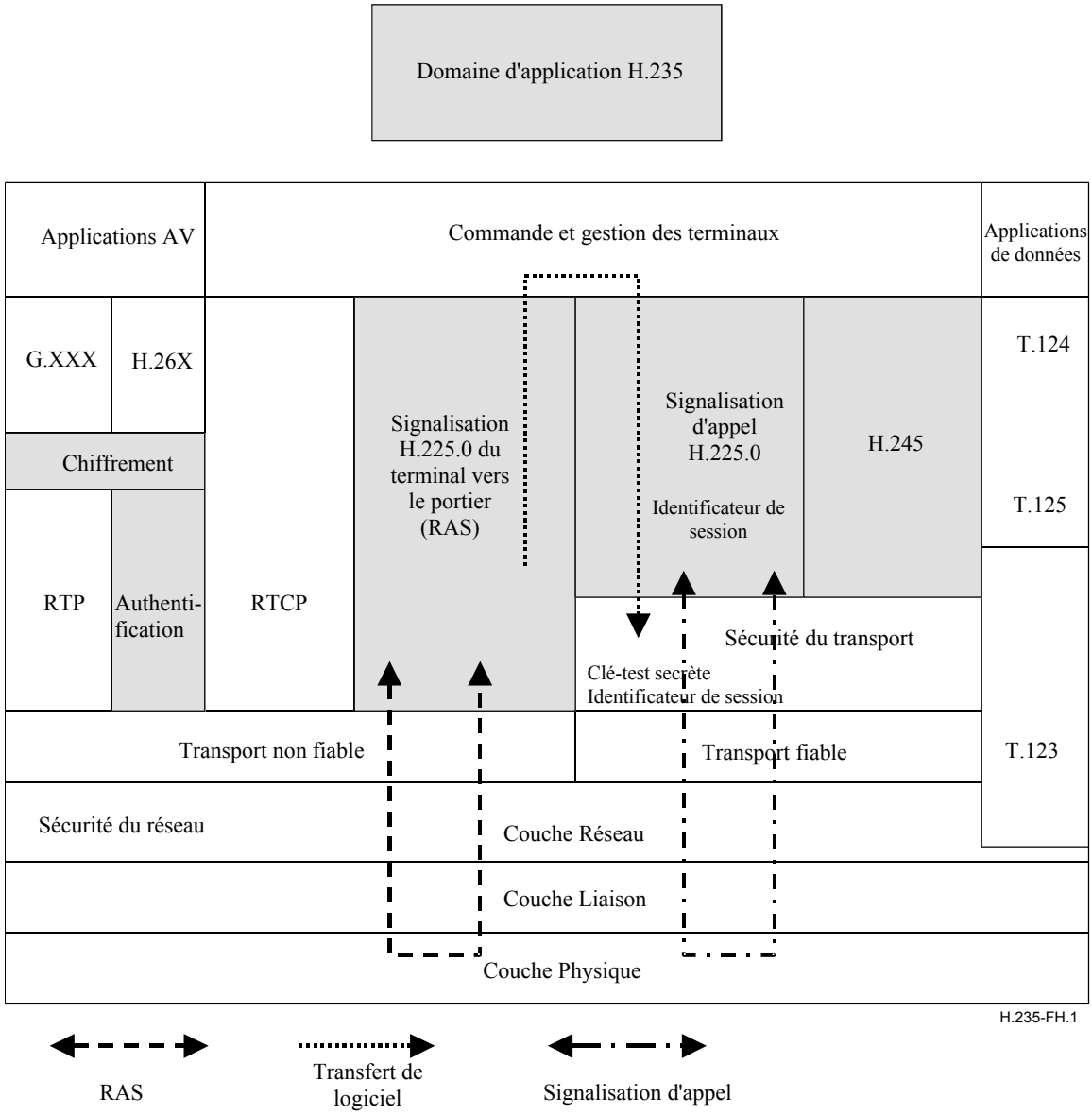

Figure H.1/H.235 – Flux d'informations pour le profil de sécurité et le protocole TLS

Le paragraphe qui suit décrit les différentes étapes du cadre de base de la Figure H.1.

H.8.1.1 Enregistrement du point d'extrémité

Un point d'extrémité peut évaluer la capacité d'un portier à prendre en charge la signalisation d'appel protégée par le protocole TLS en incluant un ou plusieurs éléments **cipherSuite** et un ou plusieurs éléments **compression** dans le profil **ClearToken** du message GRQ envoyé à l'étape 1 ci-dessus. S'il souhaite utiliser une session préalablement négociée, le point d'extrémité doit aussi inclure **sessionID** dans **ClearToken** (et ne doit indiquer que le système cryptographique et la méthode de compression correspondant spécifiquement à la session requise). Si la négociation est fondée sur une session TLS existante, aucune donnée cryptographique n'est requise dans le profil **ClearToken** à part **nonce**.

Si une session demandée n'existe pas, le portier doit choisir un autre profil d'authentification (si cela est possible) ou renvoyer un message GRJ avec un élément **GatekeeperRejectReason.resourceUnavailable**. Si la session demandée n'existe pas, les données de clé maîtresse sont obtenues à partir de la session TLS, et utilisées (avec l'élément **random** du

message GRQ et l'élément **random2** généré par le portier) pour calculer la clé d'authentification en vue de l'échange de messages RAS. L'identificateur **sessionID**, le système cryptographique **cipherSuite**, la méthode **compress** et le **nonce** du portier doivent être renvoyés dans le profil **ClearToken** d'un message GCF.

S'il peut prendre en charge la négociation de session TLS, le portier doit calculer les données de clé maîtresse telles que spécifiées dans le profil, attribuer un nouvel identificateur de session et le renvoyer dans le profil **ClearToken**, dans l'identificateur **sessionID**. Le profil **ClearToken** doit aussi contenir les paramètres de sécurité requis à l'étape 2 ci-dessus, ainsi qu'un seul système cryptographique **cipherSuite** choisi, une seule méthode **compress** choisie et l'identificateur **sessionID** non égal à zéro. A noter que la méthode d'échange de clé du système cryptographique choisi est immatérielle. Si le portier convient d'une protection TLS pour la signalisation d'appel, toutes les adresses de transport de signalisation d'appel échangées dans les messages ultérieurs RRQ/RCF ou ARQ/ACF doivent être activées TLS.

Si la négociation TLS et/ou le routage de portier ne sont pas pris en charge par le portier, aucun paramètre TLS ne doit être renvoyé; cependant, les procédures d'authentification définies à l'étape 3 ci-dessus peuvent être maintenues. Le point d'extrémité doit déterminer s'il est prêt à procéder sans protection TLS de la signalisation d'appel; il peut choisir de procéder ainsi tout en continuant à utiliser le profil d'authentification. Une fois que la séquence d'enregistrement a été exécutée, la session TLS peut être utilisée pour effectuer un établissement rapide d'une ou de plusieurs connexions de signalisation d'appel en direction du portier, sans avoir à renégocier de données de clé au moyen de méthodes applicables aux clés publiques.

Les sessions TLS ont une durée de vie limitée. Par conséquent, il peut être nécessaire pour un point d'extrémité de renégocier les paramètres de session et d'obtenir un nouvel identificateur de session. Cela peut être réalisé en procédant à l'échange des éléments nécessaires **ClearToken** décrits ci-dessus dans une séquence d'enregistrement simplifiée ("de maintien en vie", "keepalive"). Cette séquence ne doit pas avoir d'incidence sur la clé d'authentification RAS.

H.8.2 Utilisation de certificats pour l'authentification du portier

S'il est difficile d'échanger des chaînes de certificats vérifiables en mode RAS (en raison de la taille limitée des paquets UDP), il est possible qu'un serveur s'authentifie lui-même auprès du point d'extrémité si ce dernier peut obtenir une copie fiable de la clé publique du serveur par d'autres moyens. Le serveur peut inclure simplement, dans le message GCF, un élément **CryptoH323Token.cryptoGKCert** dont l'identificateur **ClearToken.tokenOID** est mis à l'identificateur OID du profil de sécurité choisi.

H.8.3 Utilisation de mécanismes de sécurité de signalisation de remplacement

Les paramètres négociés dans le cadre du profil de sécurité dans la présente annexe peuvent être employés dans des mécanismes de sécurité de couche transport et/ou application tels que déterminés par le profil donné. La séquence **profileInfo** qui a été ajoutée au profil **ClearToken** H.235 a été prévue, si nécessaire, pour cette utilisation.

H.9 Menaces (à titre indicatif)

H.9.1 Attaques passives

Le système décrit ci-dessus n'est actuellement pas vulnérable aux attaques passives, sous réserve que la négociation Diffie-Hellman ne soit pas elle-même vulnérable à ces attaques.

H.9.2 Attaques visant la fonction de refus de service

Ce système est exposé aux attaques actives visant la fonction de refus de service, dans lesquelles un tiers répond au message GRQ initial au moyen d'un message GRJ parasite. Ce type d'attaque peut ne pas être détecté: si le portier qui refuse la demande est légitime et connaît le secret partagé

(par exemple, le portier est le portier du point d'extrémité et l'élément **rejectReason** a la valeur **resourceUnavailable**), le portier pourrait alors mener à bien la négociation de la clé et authentifier le message GRJ en renvoyant, dans ce message, les mêmes éléments que ceux décrits pour le message GCF (à ceci près que l'identificateur OID renvoyé dans l'élément **authenticationMode** du message GCF serait renvoyé dans un élément **ClearToken.profileInfo** du message GRJ). Cela dépend de la définition de chaque profil.

S'il n'est pas authentifié, le message GRJ pourrait provenir d'un attaquant. Avant d'intervenir sur le message GRJ (par exemple, en cherchant un portier de remplacement), le point d'extrémité devrait attendre la réception éventuelle d'un autre message GRJ ou d'un message GCF authentifié provenant du portier correct. Sinon, il devrait essayer chaque portier proposé dans un quelconque élément **altGKInfo** reçu dans tous les messages GRJ (l'un d'eux est en principe légitime). Dans tous les cas, seul le portier correct (qui connaît le secret partagé) peut renvoyer un message GCF authentifié.

H.9.3 Attaques par entremetteur

Il est tentant d'envisager comme mode d'échange l'échange de clés Diffie-Hellman non chiffrées, avec utilisation du mot de passe ou du numéro PIN pour calculer des clés de session à partir du secret Diffie-Hellman. Cependant, ce mode d'échange est vulnérable aux attaques par entremetteur qui peuvent être utilisées pour trouver le "petit" secret partagé par la force brute au moyen de la valeur de contrôle d'intégrité fournie par le portier légitime dans le message GCF.

Tout entremetteur peut évidemment manipuler tout message RAS authentifié pour faire en sorte qu'il soit ignoré en raison d'un échec du contrôle d'intégrité. Si tous les messages peuvent être manipulés, le service en question peut être refusé.

H.9.4 Prévoir les attaques

Un attaquant peut se faire passer soit pour un point d'extrémité légitime, soit pour un portier légitime, soit pour les deux (entremetteur) et tenter de deviner le secret partagé de façon empirique. Par exemple, l'attaquant (qui est censé connaître les données du profil d'authentification mais pas le secret partagé) peut deviner un secret partagé et tenter de s'enregistrer en envoyant un message GRQ à partir de cette hypothèse. En général, le portier répondra à cette tentative au moyen d'un message GCF contenant la clé publique du portier (chiffrée au moyen du véritable secret partagé) et une valeur ICV calculée au moyen de la clé déduite qui dépend de la manière dont le portier a déchiffré la clé publique chiffrée par l'attaquant. Ce dernier peut utiliser cette information pour vérifier son hypothèse du secret partagé. Cette hypothèse confirme la valeur ICV du message GCF, il est probable que cela corresponde au véritable secret partagé; cela peut être confirmé par la séquence d'enregistrement. Si l'hypothèse ne peut pas être utilisée pour reproduire la valeur ICV du message GCF, l'attaquant doit émettre une autre hypothèse et faire une autre tentative. Avec un petit espace pour la clé du secret partagé, le nombre d'hypothèses pour une recherche "force brute" peut ne pas être prohibitif. Cette attaque nécessite la participation active du portier (ou du point d'extrémité si l'attaquant se fait passer pour le portier). La méthode traditionnelle utilisée pour contrecarrer une telle attaque consiste à limiter le nombre de tentatives infructueuses et, lorsqu'un seuil est atteint, à considérer toutes les tentatives ultérieures comme non valables (au moins pendant une période donnée) et à déclencher une alarme; cependant, ces procédures dépendent de l'implémentation.

H.9.5 Demi-clé non chiffrée par le portier

Comme il est mentionné plus haut, l'échange EKE peut rester sûr sous certaines conditions: si le portier répondant ne chiffre pas sa demi-clé Diffie-Hellman. En particulier, le portier doit être le premier participant à prouver qu'il connaît le secret partagé (numéro PIN) au moyen de la valeur ICV. Si tel n'est pas le cas, le portier (ou un intrus se faisant passer pour lui) pourrait simplement essayer toutes les combinaisons possibles de numéros PIN pour déchiffrer la demi-clé Diffie-Hellman du point d'extrémité, calculer le secret partagé Diffie-Hellman qui en résulte,

calculer la clé d'authentification et la tester par rapport à la valeur ICV fournie par le point d'extrémité. Cela n'est pas possible si le point d'extrémité peut d'abord contrôler la valeur ICV fournie par le portier et refuser de passer à l'enregistrement si la valeur ICV n'est pas celle qui est attendue.

L'utilisation d'une demi-clé non chiffrée est un avantage pour le portier dans la mesure où celui-ci peut réutiliser sa clé privée correspondante avec plusieurs points d'extrémité. Cela serait impossible si la même clé était distribuée de façon chiffrée au moyen de plusieurs secrets partagés ou numéros PIN. Un tiers observateur pourrait collecter des exemples de la demi-clé chiffrée au moyen de deux numéros PIN différents, puis chercher les combinaisons possibles de deux numéros PIN pour voir quelle paire a permis d'obtenir la même demi-clé une fois déchiffrée. S'il existe disons 10^8 numéros PIN possibles, il n'existe alors que 10^{16} combinaisons possibles à essayer. Cela équivaut à chercher un nombre aléatoire de 54 bits, ce qui n'est pas du tout infaisable. Même si l'on obtient plusieurs solutions possibles, on peut trouver rapidement la solution correcte grâce à une troisième observation.

Annexe I

Prise en charge des appels à acheminement direct

…

I.5 Symboles et abréviations

La présente annexe utilise les abréviations suivantes:

~~{~~(M~~}~~)~~K; S, IV~~ENC$_{K; S, IV}$ chiffrement EOFB de *M* au moyen de la clé secrète *K* et de la clé secrète de salage *S* et du vecteur initial *IV* (*EOFB encryption of M using secret key K and secret salting key S and initial vector IV*)

CT clearToken

DRC appel à acheminement direct (*direct-routed call*)

EPID identificateur de point d'extrémité (*endpoint identifier*)

GKID identificateur de portier (*gatekeeper identifier*)

K_{AG} secret partagé (Annexe D, Annexe F) entre EP A et GK G (*shared secret (Annex D, Annex F) between EP A and GK G*)

K_{BG}~~G~~$_{H}$ secret partagé (Annexe D, Annexe F) entre EP B et GK G (*shared secret (Annex D, Annex F) between EP B and GK* ~~*G*~~*H*)

K_{GH} secret partagé (Annexe D, Annexe F) entre GK G et GK H (*shared secret*)

KS_{AG} clé de salage partagée secrète entre EP A et GK G (*secret, shared salting key between EP A and GK G*)

KS_{BG}~~G~~$_{H}$ clé de salage partagée secrète entre EP B et GK G (*secret, shared salting key between EP B and GK* ~~*G*~~*H*)

EK'_{AG} clé de chiffrement partagée entre EP A et GK G (*the encryption key shared between EP A and GK G*)

EK'_{BH}~~G~~ clé de chiffrement partagée entre EP B et GK G (*the encryption key shared between EP* ~~*B*~~ *and GK* ~~*G*~~*H*)

K_{AB} clé de chiffrement partagée entre EP A et EP B (*the encryption key shared between EP A and EP B*)

I.6 Références normatives

…

– Recommandation UIT-T H.235, Annexe F (200~~2~~3), Corrigendum 1 du *profil hybride de sécurité*.

…

I.7 Aperçu général

Le profil de sécurité de base décrit dans l'Annexe D (voir le corps principal de la présente Recommandation) ainsi que le profil de sécurité hybride décrit dans l'Annexe F (voir l'Annexe F) (après la première prise de contact) appliquent un secret partagé pour effectuer l'authentification et/ou le contrôle d'intégrité des messages bond par bond utilisant le portier comme hôte intermédiaire de confiance. On utilise le modèle d'appel à acheminement direct, et de ce fait on ne peut pas supposer qu'il y a un secret partagé entre deux points d'extrémité. Il n'est également pas commode d'utiliser un secret partagé préétabli pour sécuriser la communication étant donné que dans ce cas tous les points d'extrémité devront connaître à l'avance quel autre point d'extrémité sera appelé.

La présente annexe traite du scénario représenté à la Figure I.1 dans lequel les points d'extrémité sont associés à un unique portier et utilisent une signalisation d'appel à acheminement direct. Dans ce scénario on suppose un réseau IP non sécurisé dans la zone du portier.

On suppose aussi que chaque point d'extrémité a une relation de communication et une association de sécurité avec le portier et que chaque point d'extrémité s'est enregistré de manière sécurisée auprès du portier en utilisant le profil de sécurité élémentaire ou le profil de sécurité hybride.

Ainsi, un portier du point d'extrémité d'origine peut offrir un secret partagé pour des points d'extrémité en communication directe en utilisant une approche de type Kerberos.

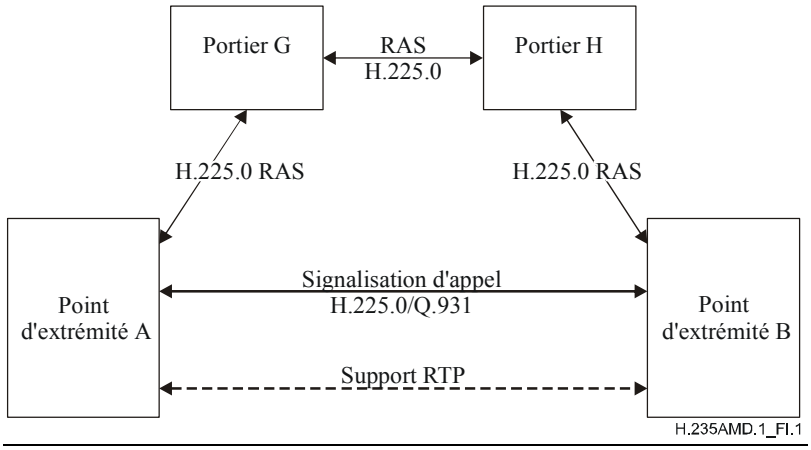

Figure I.1/H.235 – Scénario d'un appel à acheminement direct

I.8 Limitations

~~Actuellement, la présente annexe ne traite pas de scénarios à acheminement direct dans lesquels les points d'extrémité sont associés à des portiers distincts. De plus, l~~La présente annexe ne traite pas non plus des scénarios à acheminement direct sans portier. Ces scénarios appellent un complément d'étude.

I.9 Procédure DRC

Les points d'extrémité en mesure de prendre en charge ce profil de sécurité doivent l'indiquer pendant l'envoi des messages **GRQ** et/ou **RRQ** en incluant un ClearToken distinct dans lequel **tokenOID** est mis à "I0"; les autres champs dans ce ClearToken ne doivent pas être utilisés. Les portiers disposant des capacités spécifiées dans l'Annexe I souhaitant offrir cette fonctionnalité doivent répondre respectivement par un message **GCF** ou **RCF** avec un ClearToken distinct inclus dans un **tokenOID** mis à "I0", tous les autres champs du ClearToken étant inutilisés.

Avant qu'un point d'extrémité A commence à envoyer directement des messages de signalisation d'appel à un autre point d'extrémité B, le point d'extrémité A ou B doit demander son admission au portier G ou H au moyen d'un message **ARQ**. Le point d'extrémité A doit inclure dans le message **ARQ** un ClearToken distinct avec **tokenOID** mis à "I0", tous les autres champs de ClearToken étant inutilisés.

Cette procédure s'applique aussi bien à un seul portier commun à plusieurs points d'extrémité qu'à plusieurs portiers enchaînés. Dans le cas de plusieurs portiers, le portier G – zone de laquelle l'appel provient - devrait localiser le portier H au moyen du mécanisme **LRQ** (multidiffusion) tel que décrit dans le § 8.1.6/H.323 "signalisation facultative par l'extrémité appelée". La communication entre deux portiers doit être sécurisée conformément à l'Annexe D. Pour cela, on part du principe qu'un secret partagé commun K_{GH} est disponible. Etant donné que le message **LRQ** parmi les portiers est généralement un message multidiffusion, le secret partagé K_{GH} ne peut pas en principe être un secret partagé par une paire mais est censé être en fait un secret partagé par un groupe à l'intérieur du nuage potentiel de portier.

NOTE – Cette hypothèse limite l'échelonnabilité dans le cas général et ne permet pas l'authentification de sources. Cependant, on estime que dans les réseaux d'entreprise dont le nombre de portiers bien établis est petit et limité, ces obstacles à la sécurité sont encore acceptables. On pourrait surmonter ces derniers en sécurisant les communications multidiffusion entre portiers au moyen de signatures numériques; cette question appelle toutefois un complément d'étude.

Si le mécanisme **LRQ** est utilisé pour localiser le portier à l'extrémité distante, le message **LRQ** doit alors acheminer un jeton ClearToken distinct dont l'identificateur **tokenOID** est mis à "I0"; tous les autres champs de ce ClearToken ne devraient pas être utilisés. En mode multidiffusion, l'identificateur **generalID** dans le jeton CryptoToken du message **LRQ** ne doit pas être utilisé. La communication entre portiers H.501 et/ou H.510 fera l'objet d'un complément d'étude.

EK_{BH} désigne la clé de chiffrement qui est partagée entre le point d'extrémité B et le portier H. Le portier H doit générer les données de clé de chiffrement EK_{BH} à partir du secret partagé K_{BH} au moyen de la procédure de calcul de clé **fondée** sur la fonction PRF, telle que définie dans le § I.10 où **keyDerivationOID** dans **V3KeySyncMaterial** doit définir "Annexe I-HMAC-SHA1-PRF"; voir le § I.12.

Le portier H doit transmettre la clé EK_{BH} chiffrée au portier G. Le mode de chiffrement OFB amélioré (EOFB) (voir le § B.2.5) doit être utilisé avec le secret, la clé de salage KS_{GH} propre au point d'extrémité. Les algorithmes de chiffrement applicables sont les suivants (voir le § D.11):

- DES (56 bits) dans le mode EOFB au moyen de l'identificateur OID "Y1": optionnel;
- 3DES (168 bits) dans le mode EOFB externe au moyen de l'identificateur OID "Z1": optionnel;
- AES (128 bits) dans le mode EOFB au moyen de l'identificateur OID "Z2": par défaut et recommandé;
- compatible RC2 (56 bits) dans le mode EOFB au moyen de l'identificateur OID "X1": optionnel.

Pour le mode de chiffrement EOFB, le portier H doit générer une valeur initiale aléatoire IV. Pour les OID "X1", OID "Y1"et OID "Z1", le vecteur IV occupe 64 bits et doit être acheminé dans le

champ **iv8** de l'élément **params** dans **V3KeySyncMaterial**; en revanche, le vecteur IV occupe 128 bits pour l'OID "Z2" et doit être acheminé dans le champ **iv16** de **params** dans **V3KeySyncMaterial**.

Le portier H doit inclure $ENC_{K_{GH}, KS_{GH}, IV}(EK_{BH})$ dans le ClearToken CT_{HG} dont l'identificateur **tokenOID** est mis à "I3". Le texte chiffré obtenu $ENC_{K_{GH}, KS_{GH}, IV}(EK_{BH})$ doit être acheminé dans la structure de données **h235key** comme faisant partie de **secureSharedSecret** où il doit être inséré dans le **encryptedSessionKey** de la structure de données **secureSharedSecret**. L'algorithme de chiffrement doit être indiqué dans **algorithmOID** ("X1", "Y1", "Z1" ou "Z2") dans **V3KeySyncMaterial**. La réponse **LCF** doit définir le ClearTokenCT_{HG}.

Le portier G constatant que les points d'extrémité A et B sont compatibles avec la présente annexe doit générer les éléments de clé et les ClearToken comme spécifié ci-dessous.

Le portier est en mesure de calculer le secret partagé K_{AB} associé à l'appel, outre l'opération **ARQ** normale. Ce secret partagé fondé sur l'appel est ensuite propagé aux deux points d'extrémité au moyen de ClearToken. Ces derniers sont acheminés dans le message **ACF** et envoyés à l'appelant.

Deux ClearToken doivent être inclus, un CT_A pour l'appelant A et un autre CT_B pour l'appelé B. Chaque **ClearToken** doit contenir un identificateur OID ("I1" ou "I2") à l'intérieur de **tokenOID** qui indique si le jeton est destiné à l'appelant (OID "I1" pour CT_A) ou à l'appelé (OID "I2" pour CT_B).

Le **ClearToken** tel que défini dans la présente annexe, peut être utilisé en association avec d'autres profils de sécurité tels que ceux décrits dans les Annexes D ou F qui mettent en œuvre des **ClearToken** également. En pareil cas, un ClearToken conforme à l'Annexe I doit utiliser d'autres champs de **ClearToken** également. Par exemple, afin d'utiliser l'Annexe I en association avec l'Annexe D, les champs **timeStamp**, **random**, **generalID**, **sendersID** et **dhkey** doivent être présents et être utilisés tels que décrits par les profils de sécurité définis dans l'Annexe D.

L'identificateur de portier (GKID) doit être inséré dans **sendersID** tandis que **generalID** ne doit pas contenir d'identificateur des points d'extrémité A (CT_A) ou B (CT_B).

EK' désigne la clé de chiffrement qui est partagée entre un point d'extrémité et ~~le~~son portier. Les clés de chiffrement EK'_{AG} et EK'_{H} pour la clé chiffrée de bout en bout K_{AB} doivent être calculées à partir du secret partagé entre le portier et les points d'extrémité (K_{AG} ou $K_{B~~G~~H}$) en utilisant la procédure de calcul de clé **fondée** sur la fonction PRF comme défini au § I.10 où **keyDerivationOID** dans **V3KeySyncMaterial** doit contenir "Annex I-HMAC-SHA1-PRF", voir § I.12.

Le portier G doit générer un secret de session commun partagé K_{AB}, qui est partagé entre le point d'extrémité A et le point B.

Ce secret de session K_{AB} doit être chiffré par EK'_{AG} (pour l'identificateur CT destiné au point d'extrémité A) ou par $EK'_{B~~G~~H}$ (pour l'identificateur CT destiné au point d'extrémité B) au moyen de l'algorithme de chiffrement.

Le mode de chiffrement OFB amélioré (EOFB) (voir § B.2.5) doit être utilisé avec le secret, la clé de salage KS_{AG} ou KS_{BG} propre au point d'extrémité. Les algorithmes de chiffrement applicables sont les suivants (voir § D.11):

- DES (56 bits) dans le mode EOFB utilisant l'OID "Y1": optionnel;
- 3DES (168 bits) dans le mode EOFB externe utilisant l'OID "Z1": optionnel;
- AES (128 bits) dans le mode EOFB utilisant l'OID "Z2": par défaut et recommandé;
- compatible RC2 (56 bits) dans le mode EOFB utilisant l'OID "X1": optionnel.

Pour le mode de chiffrement EOFB, le portier doit générer une valeur initiale aléatoire IV. Pour les OID "X1", OID "Y1" et OID "Z1", le vecteur IV occupe 64 bits et doit être acheminé dans le champ

iv8 de l'élément **paramS** dans **V3KeySyncMaterial**; en revanche le vecteur IV occupe 128 bits pour l'OID "Z2" et doit être acheminé dans le champ **iv16** de **paramS** dans **V3KeySyncMaterial**.

Les textes respectifs chiffrés obtenus $\underline{ENC_{EK_{AG,}\ KS_{AG,}\ IV}(\{K_{AB}\})}$ ~~$_{K'AG,\ KS_{AG,\ IV}}$~~ ou $\underline{ENC_{EK_{BG,}\ KS_{BG,\ IV}}(\{K_{AB}\})}$ ~~$_{K'BG,\ KS_{BG,\ IV}}$~~ doivent être alors acheminés dans la structure de données **h235key** comme faisant partie de **secureShareSecret** où ils doivent être insérés dans le **encryptedSessionKey** de la structure de données **secureSharedSecret**. L'algorithme de chiffrement doit être indiqué dans **algorithmOID** ("X1", "Y1", "Z1" ou "Z2") dans **V3KeySyncMaterial**.

Pour le ClearToken destiné au point d'extrémité A, l'identificateur de point d'extrémité du point d'extrémité B ($EPID_B$) doit être inséré dans **generalID** de **V3KeySyncMaterial**. De même que pour le ClearToken destiné au point d'extrémité B, le point d'identificateur de point d'extrémité du point d'extrémité A ($EPID_A$) doit être inséré dans l'élément **generalID** de **V3KeySyncMaterial**.

Pour les algorithmes de chiffrement EOFB, l'élément **encryptedSaltingKey** ne doit pas être utilisé.

Le portier doit inclure à la fois les identificateurs CT_A et CT_B de ClearToken dans le message **ACF** en direction du point d'extrémité A.

Le point d'extrémité A doit identifier le CT_A par inspection de l'identificateur **tokenOID** "I1" dans ClearToken.

Le point d'extrémité A doit vérifier que l'identificateur CT_A est tout nouveau au moyen du **timeStamp**. Des contrôles de sécurité plus poussés doivent être effectués pour vérifier le **generalID** et le **sendersID** de ClearToken et le **generalID** dans **V3KeySyncMaterial**. Si après vérification, l'identificateur CT_A reçu est tout nouveau, le point d'extrémité A doit récupérer le vecteur IV et calculer $\underline{EK'}_{\sout{A}G}$ et KS_{AG} comme décrit ci-dessus pour le portier G. Le point d'extrémité A doit décrypter l'information **encryptedSessionKey** qui se trouve dans **V3KeySyncMaterial** du CT_A pour obtenir la clé $\underline{EK'}_{AB}$.

Si après vérification, il s'avère que l'identificateur CT_A est tout nouveau, le point d'extrémité A est en mesure d'envoyer un message SETUP au point d'extrémité B. Ce message SETUP inclut l'identificateur CT_B et doit être sécurisé (authentifié et/ou protégé dans son intégrité) au moyen des profils décrits dans les Annexes D ou F en se servant de la clé K_{AB} comme secret partagé. A cette fin, l'élément **generalID** du jeton ClearToken haché de l'Annexe D, (non pas CT_B !) ne doit pas être ~~mis à~~utilisé à moins que le point d'extrémité A ait déjà un $EPID_B$ disponible (par exemple, par configuration ou par mémorisation d'une communication antérieure. S'il utilise une valeur $EPID_B$ pour l'identificateur **generalID** dans le message SETUP, le point d'extrémité A doit accepter la valeur de l'identificateur **sendersID** dans le message de signalisation d'appel renvoyé en tant que l'$EPID_B$ Vrai.

Le point d'extrémité B doit identifier CT_B par inspection de l'identificateur **tokenID** "I2" dans ClearToken.

Le point d'extrémité B doit vérifier que l'identificateur CT_B est tout nouveau en utilisant l'horodate **timeStamp**. Des vérifications plus poussées doivent être effectuées sur ~~**generalID et**~~ **sendersID** de ClearToken et sur **generalID** dans **V3KeySyncMaterial**. Si l'identificateur CT_B reçu est après vérification effectivement tout nouveau, le point d'extrémité B doit récupérer le vecteur IV et calculer $\underline{EK'}_{BG}$ et KS_{BG} tel que décrit ci-dessus pour le portier. Le point d'extrémité B doit décrypter l'information **encryptedSessionKey** se trouvant dans **V3KeySyncMaterial** de l'identificateur CT_B pour obtenir le secret partagé $\underline{EK'}_{AB}$.

Si après vérification, il s'avère que l'identificateur CT_B est tout nouveau, le point d'extrémité B est en utilise la signalisation d'appel en répondant par un message CALL-PROCEEDING, ALERTING ou CONNECT, etc. selon le cas. Si après vérification, il s'avère que l'identificateur n'est pas tout nouveau ou si la vérification de sécurité du message SETUP révèle un problème, le point d'extrémité B doit répondre par un message RELEASE-COMPLETE, l'élément **ReleaseCompleteReason** étant mis à une erreur de sécurité définie dans le § B.2.2.

Lorsque la sécurité de media est appliquée (voir § D.7), le point d'extrémité A et le point d'extrémité B doivent procéder à l'échange des demi-clés de Diffie-Hellman conformément au § D.7.1 et établir une clé maître dynamique de session à partir de laquelle des clés propres au média peuvent être déduites.

Le point d'extrémité B doit inclure generalID mis à $EPID_A$ et sendersID mis à $EPID_B$ pour la protection de tout message de signalisation d'appel H.225.0 destiné à EP A (par exemple, Call Proceeding, Alerting ou Connect).

La Figure I.2 illustre le flux de communications de base.

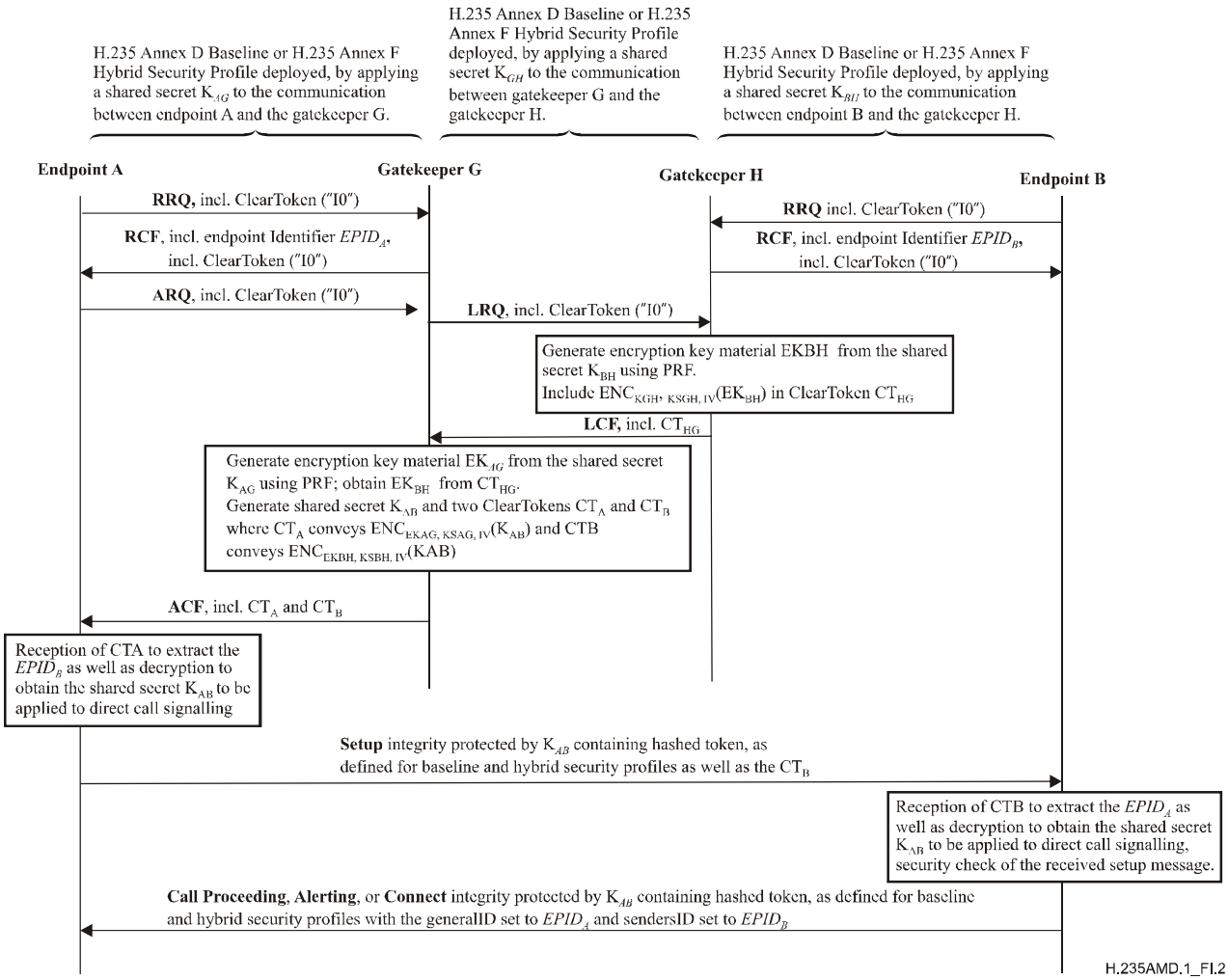

Figure I.2/H.235 – Flux de communications de base

I.10 Procédure d'obtention de la clé au moyen de la fonction PRF

Le présent paragraphe décrit une procédure qui indique comment obtenir les éléments de clé à partir du secret partagé et d'autres paramètres.

La clé de chiffrement EK'_{AG} doit être calculée au moyen de la fonction PRF (voir § B.7), le paramètre *inkey* étant mis à K_{AG} et *label* mis à la constante 0x2AD01C64 || **challenge**.

De même, la clé de chiffrement EK'_{BG} doit être calculée en utilisant cette fonction PRF, le paramètre *inkey* étant mis à K_{BGH} et *label* à la constante 0x1B5C7973 || **challenge**. Dans les deux cas, le paramètre *outkey_len* doit être mis à la longueur requise de la clé de chiffrement pour l'algorithme de chiffrement choisi.

En utilisant cette même fonction, un secret, une clé de salage partagée doivent être générés par le portier et par chaque point d'extrémité. La clé de salage, lorsqu'elle est utilisée dans le mode de chiffrement EOFB, empêche les attaques de texte clair connues du CT_B par le point d'extrémité A dans lequel le point d'extrémité A pourrait dans les autres cas tenter de découvrir $K_{B\text{~~G~~}H}$.

KS_{AG} désigne la clé de salage secrète partagée qui est partagée entre le point d'extrémité A et le portier G. KS_{AG} doit être calculé en utilisant la fonction PRF, le paramètre *inkey* étant mis à K_{AG} et le paramètre *label* à la constante 0x150533E1 || **challenge**. $KS_{B\text{~~G~~}H}$ doit être calculé en utilisant la fonction PRF, le paramètre *inkey* étant mis à $K_{B\text{~~G~~}H}$ et le paramètre *label* à la constante 0x39A2C14B || **challenge**.

...

Tableau I.1/H.235 – Identificateurs d'objet utilisés dans l'Annexe I/H.235

Référence de l'identificateur d'objet	Valeur de l'identificateur d'objet	Description
...	...	...
"I2"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 50}	Utilisé dans la procédure DRC pour le ClearToken tokenOID indiquant que le ClearToken détient une clé de bout en bout pour l'appelé.
"I3"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 52}	Utilisé dans la procédure DRC pour le ClearToken tokenOID entre portiers indiquant que le ClearToken détient une clé de chiffrement pour le portier d'origine.
...	...	...

Appendice I

Détails d'implémentation H.323

…

I.4.5 Sécurité IPSEC

En général, la méthode IPSEC ([IPSEC], [ESP]) et IKE [IKE] peut être utilisée pour assurer l'authentification et, facultativement, la confidentialité (c'est-à-dire le chiffrement) dans la couche IP de façon transparente à tout protocole (applicatif) exploité dans les couches supérieures. Le protocole applicatif n'a pas besoin d'être mis à jour pour permettre cette opération; seule la politique de sécurité à chaque extrémité doit correspondre.

Par exemple, pour tirer le meilleur parti de la sécurité IPSEC pour une simple communication point à point, le scénario ci-après peut être suivi:

1) le point d'extrémité appelant et son portier détermineront par le protocole RAS la politique prescrivant l'utilisation de la sécurité IPSEC (authentification et, facultativement, confidentialité). Avant l'envoi du premier message RAS du point d'extrémité au portier, le démon ISAKMP [ISAKMP]/Oakley [RFC 2412] situé au point d'extrémité négociera les services de sécurité à utiliser pour les paquets à destination et en provenance de l'accès notoire du canal RAS. Une fois la négociation achevée, le canal RAS fonctionne

exactement comme s'il n'avait pas été sécurisé. Au moyen de ce canal sécurisé, le portier informera le point d'extrémité de l'adresse et du numéro d'accès du canal de signalisation d'appel se trouvant au point d'extrémité appelé;

2) après avoir obtenu l'adresse et le numéro d'accès du canal de signalisation d'appel, le point d'extrémité appelant met à jour dynamiquement sa politique de sécurité afin de demander la sécurité IPSEC souhaitée à cette adresse pour cette paire protocole/accès. Ensuite, lorsque le point d'extrémité appelant tentera de se mettre en contact avec cette paire adresse/accès, les paquets seront mis en file d'attente pendant l'exécution d'une négociation par routine ISAKMP [ISAKMP]/Oakley [RFC 2412] entre les points d'extrémité. A l'achèvement de cette négociation, une association de sécurité IPSEC existera pour cette paire adresse/accès et la signalisation Q.931 pourra commencer;

3) lors de l'échange des messages Q.931 SETUP et CONNECT, les points d'extrémité peuvent négocier l'utilisation de la sécurité IPSEC pour le canal H.245. Cela permettra aux points d'extrémité de remettre à jour dynamiquement leurs bases de données pour politique de sécurité IPSEC et d'imposer l'utilisation de cette politique sur cette connexion;

4) comme dans le canal de signalisation d'appel, une négociation ISAKMP [ISAKMP]/Oakley [RFC 2412] transparente se déroulera avant qu'un quelconque paquet H.245 soit émis. L'authentification effectuée par cet échange ISAKMP [ISAKMP]/Oakley [RFC 2412] sera la tentative initiale d'une authentification d'utilisateur à utilisateur. Elle établira un canal (probablement) sécurisé entre les deux utilisateurs, permettant de négocier les caractéristiques du canal audio. Si, à la suite d'un dialogue interpersonnel, l'un des utilisateurs n'est pas satisfait de l'authentification, différents certificats peuvent être choisis et l'échange ISAKMP [ISAKMP]/Oakley [RFC 2412] peut être répété;

5) après chaque authentification ISAKMP [ISAKMP]/Oakley [RFC 2412] H.245, de nouvelles données de clé sont échangées pour le canal audio en protocole RTP. Ces données sont distribuées par le maître sur le canal H.245 sécurisé. Comme le protocole H.245 est défini de façon que le maître distribue les données de clés multimédias sur le canal H.245 (afin de permettre des communications multipoints), il n'est pas recommandé d'utiliser la méthode IPSEC pour le canal RTP.

...

Appendice IV

Bibliographie

[Daemon] DAEMON (J.), Cipher and Hash function design, *Ph.D. Thesis, Katholieke Universiteit Leuven*, March 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-08.txt>*, RFC xxxx, Work in Progress (MSEC WG), IETF, 10/2003.

{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.~~

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	**Systèmes audiovisuels et multimédias**
Série I	Réseau numérique à intégration de services
Série J	Réseaux câblés et transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, circuits téléphoniques, télégraphie, télécopie et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information, protocole Internet et réseaux de nouvelle génération
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication

Imprimé en Suisse
Genève, 2004