Union internationale des télécommunications

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

H.235.6

(09/2005)

SÉRIE H: SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

Infrastructure des services audiovisuels – Aspects système

Cadre de sécurité H.323: profil pour le chiffrement vocal avec gestion de clés native dans les systèmes H.235/H.245

Recommandation UIT-T H.235.6

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.235.6

Cadre de sécurité H.323: profil pour le chiffrement vocal avec gestion de clés native dans les systèmes H.235/H.245

Résumé

La présente Recommandation décrit les procédures de sécurité pour le profil de chiffrement vocal (ex-Annexe D/H.235), y compris la gestion de clés H.235/H.245 native correspondante.

Dans les anciennes versions de la sous-série H.235, ce profil était défini dans le corps de la Rec. UIT-T H.235 et dans son Annexe D. Les Appendices IV, V et VI/H.235.0 donnent le mappage entre tous les paragraphes, toutes les figures et tous les tableaux de la version 3 et tous ceux de la version 4 de la Rec. UIT-T H.235.

Source

La Recommandation UIT-T H.235.6 a été approuvée le 13 septembre 2005 par la Commission d'études 16 (2005-2008) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8.

Mots clés

Authentification, certificat, chiffrement, chiffrement vocal, gestion de clés, intégrité, profil de sécurité, sécurité multimédia, signature numérique.

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 n'avait pas 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.235.6

Cadre de sécurité H.323: profil pour le chiffrement vocal avec gestion de clés native dans les systèmes H.235/H.245

1 Domaine d'application

La présente Recommandation spécifie un profil de sécurité pour le chiffrement vocal qui utilise la gestion de clés H.235/H.245 native. Elle définit des procédures à la fois pour le chiffrement vocal et pour la gestion de clés H.245 native correspondante.

2 Références

2.1 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.

– Recommandation UIT-T H.225.0 (2003), *Protocoles de signalisation d'appel et paquétisation des flux monomédias pour les systèmes de communication multimédias en mode paquet*.

– Recommandation UIT-T H.235 version 1 (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 version 2 (2000), *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 version 3 (2003), *Sécurité et chiffrement pour les terminaux multimédias de la série H (terminaux H.323 et autres terminaux de type H.245)*, plus Corrigendum 1 (2005).

– Recommandation UIT-T H.235.0 (2005), *Cadre de sécurité H.323*: *cadre de sécurité pour les systèmes multimédias de la série H (systèmes H.323 et autres systèmes de type H.245)*.

– Recommandation UIT-T H.235.1 (2005), *Cadre de sécurité H.323*: *profil de sécurité de base*.

– Recommandation UIT-T H.235.2 (2005), *Cadre de sécurité H.323*: *profil de sécurité avec signature*.

– Recommandation UIT-T H.235.3 (2005), *Cadre de sécurité H.323*: *profil de sécurité hybride*.

– Recommandation UIT-T H.245 (2005), *Protocole de commande pour communications multimédias*.

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

– Recommandation UIT-T H.323 Annexe F (1999), *Dispositifs d'extrémité simples*.

– Recommandation UIT-T X.800 (1991), *Architecture de sécurité pour l'interconnexion en systèmes ouverts d'applications du CCITT*.

ISO 7498-2:1989, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Partie 2: Architecture de sécurité*.

– Recommandation UIT-T X.803 (1994) | ISO/CEI 10745:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – Modèle de sécurité pour les couches supérieure*s.

– Recommandation UIT-T X.810 (1995) | ISO/CEI 10181-1:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Cadres de sécurité pour les systèmes ouverts: aperçu général*.

– Recommandation UIT-T X.811 (1995) | ISO/CEI 10181-2:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Cadres de sécurité pour les systèmes ouverts: cadre d'authentification*.

– IETF RFC 2198 (1997), *RTP Payload for Redundant Audio Data*.

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

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

– IETF RFC 2833 (2000), *RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals*.

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

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

– ISO/CEI 9797-1:1999, *Technologies de l'information – Techniques de sécurité – Codes d'authentification de message (MAC) – Partie 1: Mécanismes utilisant un cryptogramme bloc*.

– ISO/CEI 9797-2:2002, *Technologies de l'information – Techniques de sécurité – Codes d'authentification de message (MAC) – Partie 2: Mécanismes utilisant une fonction de hachage*.

– ISO/CEI 10118-3:2004, *Technologies de l'information – Techniques de sécurité – Fonctions de brouillage – Partie 3: Fonctions de brouillage dédiées*.

– ISO/CEI 10116:2006, *Technologies de l'information – Techniques de sécurité – Modes opératoires d'un chiffrement par blocs de n-bits*.

2.2 Références informatives

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

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

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

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

[LI] ETSI TR 101 772 V1.1.2, Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 3; Service Independent requirements definition; Lawful interception – top level requirements.

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

[RFC2268] IETF RFC 2268 (1998), *A Description of the RC2(r) Encryption Algorithm*.

[RFC2405] IETF RFC 2405 (1998), *The ESP DES-CBC Cipher Algorithm With Explicit IV*.

[RFC2412] IETF RFC 2412 (1998), *The OAKLEY Key Determination Protocol*.

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

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

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

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

[ISAKMP] IETF RFC 2408 (1998), *Internet Security Association and Key Management Protocol (ISAKMP)*.

[J.170] Recommandation UIT-T J.170 (2005), *Spécification de la sécurité sur IPCablecom*.

[RTP] IETF RFC 3550 (2003), *RTP: A transport Protocol for Real-Time Applications*.

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

[SRTP] IETF RFC 3711 (2004), *The Secure Real-Time Transport Protocol*.

3 Termes et définitions

Dans la présente Recommandation, les définitions figurant au § 3/H.323, au § 3/H.225.0 et au § 3/H.245 s'appliquent. Certains des termes suivants sont utilisés selon la définition donnée dans la Rec. UIT-T X.800 | ISO 7498-2 et dans les Recommandations UIT-T X.803 | ISO/CEI 10745, X.810 | ISO/CEI 10181-1 et X.811 | ISO/CEI 10181-2.

La **clé de session** utilisée pour le chiffrement des flux de média est produite par le maître pour une session RTP spécifique (dans un élément OLC), au maximum pour la durée de la communication. La clé de session produite est chiffrée au moyen d'une clé obtenue à partir du **secret partagé** Diffie-Hellman convenu que les deux points d'extrémité ont calculé. Dans ce cas, le secret partagé DH joue le rôle de clé maîtresse pour la protection de la ou des clés de session.

4 Symboles et abréviations

La présente Recommandation utilise les abréviations suivantes:

3DES triple DES

AES norme de chiffrement perfectionné (*advanced encryption standard*)

ASN.1	notation de syntaxe abstraite numéro un (*abstract syntax notation one*)
CBC	chaînage de blocs chiffrants (*cipher block chaining*)
CFB	chiffrement avec bouclage (*cipher feedback*)
DES	norme de chiffrement des données (*data encryption standard*)
DH	Diffie-Hellman
DTMF	multifréquence bitonalités (*dual tone multi-frequency*)
ECB	mode dictionnaire (*electronic code book*)
EOFB	mode avec bouclage de sortie amélioré (*enhanced output feedback mode*)
EP	point d'extrémité (*endpoint*)
FEC	correction d'erreur directe (*forward error correction*)
GK	portier (*gatekeeper*)
HMAC	code d'authentification de message haché (*keyed-hash message authentication code*)
IPsec	sécurité du protocole Internet (*Internet protocol security*)
IV	vecteur d'initialisation (*initialization vector*)
KS	clé de salage (*salting key*) en mode EOFB
MAC	code d'authentification de message (*message authentication code*)
MC	contrôleur multipoint (*multipoint controller*)
MCU	unité de commande multipoint, pont de conférence (*multipoint control unit*)
MPS	flux de charge utile multiple (*multiple payload stream*)
OFB	mode avec bouclage de sortie (*output feedback mode*)
OID	identificateur d'objet (*object identifier*)
OLC	ouverture de canal logique (*open logical channel*)
RAS	enregistrement, admission et statut (*registration, admission and status*)
RC	chiffrement de Rivest (*Rivest cipher*)
ROC	compteur de cycles complets (*roll-over counter*)
RSA	Rivest, Shamir et Adleman
RTCP	protocole de commande de transport en temps réel (*real-time control protocol*)
RTP	protocole de transport en temps réel (*real-time protocol*)
SDU	unité de données de service (*service data unit*)
SEQ	numéro de séquence (*sequence number*)
SHA	algorithme de hachage sécurisé (*secure hash algorithm*)
TCP	protocole de commande de transmission (*transmission control protocol*)
TLS	sécurité de la couche de transport (*transport layer security*)
TSAP	point d'accès au service de transport (*transport service access point*)
UDP	protocole de datagramme d'utilisateur (*user datagram protocol*)
UIT	Union internationale des télécommunications
XOR	OU eXclusif (*eXclusive OR*)

5 Conventions

Dans la présente Recommandation, les conventions suivantes s'appliquent:

– la forme "doit/doivent" indique une disposition obligatoire;

– la forme "devrait/devraient" indique une mesure suggérée mais facultative;

– la forme "peut/peuvent" indique une action possible plutôt qu'une action recommandée.

Lorsqu'est mis en œuvre un chiffrement de média en association avec un bourrage de charge utile, le texte indique parfois "la valeur du bourrage devrait être déterminée par la convention normale de l'algorithme de chiffrement", voir par exemple les § 7.6.1, 8.3 et la Figure I.7. Il s'agit d'indiquer que certains algorithmes de chiffrement (par exemple, DES) contiennent des informations d'implémentation supplémentaires qui permettent à l'expéditeur de choisir la valeur du ou des octets de bourrage. On peut citer par exemple des valeurs aléatoires de remplissage, des valeurs statiques ou d'autres séquences générées. Quelle que soit la méthode utilisée, elle n'aura pas d'effet sur l'interopérabilité, mais la qualité de la sécurité pourra être très différente. Ce point est considéré comme relevant de l'implémentation et n'est donc pas traité plus avant dans la présente Recommandation.

6 Aperçu général du système

6.1 Profil de sécurité pour le chiffrement vocal

Le profil de sécurité pour le chiffrement vocal n'est pas un profil indépendant comme c'est le cas du profil de sécurité de base; il s'agit plutôt d'une option du profil de sécurité susmentionné qui peut être utilisée en association avec celui-ci. Ce profil repose également sur certains services de sécurité dans le cadre de la signalisation d'appel et des procédures d'établissement de la connexion, par exemple la concordance de clés Diffie-Hellman et d'autres fonctions de gestion de clés.

Les entités H.323 peuvent implémenter la présente Recommandation pour obtenir la confidentialité vocale. Quatre algorithmes de chiffrement sont proposés (AES, compatible-RC2, DES et triple-DES), les algorithmes à utiliser étant fonction du modèle commercial et des besoins d'exportabilité. En plus du mode de chiffrement CBC, les entités H.323 peuvent implémenter le mode de chiffrement par flux EOFB. Dans les environnements qui offrent déjà un certain degré de confidentialité, le chiffrement vocal n'est peut-être pas nécessaire. Si c'est le cas, la concordance de clés Diffie-Hellman et les autres procédures de gestion de clés sont également superflues.

En ce qui concerne la confidentialité vocale, facultative, l'algorithme de chiffrement suggéré est l'algorithme AES-128, compatible-RC2, DES ou triple-DES, en fonction du modèle commercial et des besoins d'exportabilité. Dans les environnements qui offrent déjà un certain degré de confidentialité, le chiffrement vocal n'est peut-être pas nécessaire. Si c'est le cas, la concordance de clés Diffie-Hellman et les autres procédures de gestion de clés sont également superflues.

Le profil de la présente Recommandation inclut aussi la liste des algorithmes de chiffrement vocal possibles que l'Annexe D de la version 2 de la Rec. UIT-T H.235 et l'Annexe D de la version 3 de la Rec. UIT-T H.235 proposaient.

NOTE 1 – Ce nouveau profil fondé sur des algorithmes de chiffrement tient compte des développements connus dans le domaine de la crypto-analyse et de la sécurité concernant les forces des algorithmes de chiffrement ainsi que de la modification des politiques d'exportation cryptographique. En particulier, il tient compte des exigences d'interopérabilité avec les systèmes conformes à la version 2 ou 3 de la Rec. UIT-T H.235.

Les entités H.323 qui implémentent la présente Recommandation avec la version 4 ou une version ultérieure de la Rec. UIT-T H.235 doivent proposer l'algorithme AES à 128 bits comme algorithme de chiffrement vocal préféré dans leurs capacités de sécurité afin d'offrir la meilleure performance et la meilleure sécurité possibles. En outre, ces entités H.323 peuvent facultativement proposer aussi

l'algorithme triple-DES à 168 bits comme algorithme de chiffrement vocal pour offrir un meilleur interfonctionnement avec les systèmes H.323 qui ont implémenté les fonctionnalités de chiffrement vocal de l'Annexe D des versions 2 et 3 de la Rec. UIT-T H.235. Comme les algorithmes de chiffrement DES à 56 bits et compatible-RC2 à 56 bits (exportable) ne sont plus considérés comme étant suffisamment sûrs, les entités H.323 ne devraient pas proposer ces algorithmes de chiffrement faibles sauf s'ils sont expressément nécessaires, par exemple pour offrir l'interfonctionnement avec des systèmes de chiffrement vocal conformes à l'Annexe D des versions 2 et 3 de la Rec. UIT-T H.235.

Il est préférable que les entités H.323 qui implémentent la présente Recommandation avec la version 4 de la Rec. UIT-T H.235 acceptent l'algorithme AES à 128 bits proposé si leur politique de sécurité le permet. Ces entités H.323 devraient en outre accepter l'algorithme triple-DES à 168 bits si l'algorithme AES n'a pas été proposé ou s'il n'est pas autorisé par leur politique de sécurité. Ces entités H.323 ne devraient pas accepter l'algorithme DES à 56 bits ou l'algorithme compatible-RC2 à 56 bits pour des raisons de sécurité, sauf si ces algorithmes de chiffrement peu sûrs sont autorisés par leur politique de sécurité ou s'ils sont nécessaires pour des raisons d'exportabilité et que d'autres algorithmes plus sûrs (AES à 128 bits ou triple-DES à 168 bits, par exemple) ne sont pas proposés.

Les moyens de contrôle d'accès ne sont pas décrits explicitement; ils peuvent être implémentés localement compte tenu des informations reçues dans les champs de signalisation H.235 (ClearToken, CryptoToken).

La présente Recommandation ne décrit pas les procédures d'attribution des mots de passe/clés secrètes au moment de l'abonnement ni les procédures de gestion et d'administration associées. De telles procédures peuvent être exécutées par des moyens qui ne sont pas traités dans la présente Recommandation.

Les entités de communication concernées ont la possibilité de déterminer implicitement lequel du profil de sécurité de base et du profil de sécurité avec signature est utilisé, en évaluant les identificateurs d'objet de sécurité signalés dans les messages (identificateurs **tokenOID** et **algorithmOID**; voir également le § 11).

Le Tableau 1 récapitule les fonctionnalités de sécurité du profil pour le chiffrement vocal, profil qui est spécifié aux § 7, 8 et 9.

Tableau 1/H.235.6 – Profil pour le chiffrement vocal

<table>
<tr><th rowspan="2">Services de sécurité</th><th colspan="4">Fonctions d'appel</th></tr>
<tr><th>RAS</th><th>H.225.0</th><th>H.245</th><th>RTP</th></tr>
<tr><td>Authentification et intégrité</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Non-répudiation</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Confidentialité</td><td></td><td></td><td></td><td>DES à 56 bits | compatible-RC2 à 56 bits | triple-DES à 168 bits | AES à 128 bits
Mode CBC ou mode EOFB</td></tr>
<tr><td>Contrôle d'accès</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Gestion de clés</td><td></td><td>Echange de clés Diffie-Hellman authentifié</td><td>Gestion de clés de session H.235 intégrée (échange de clés Diffie-Hellman authentifié, mise à jour de clé)</td><td></td></tr>
</table>

Selon la procédure générale, un secret partagé est établi (échange Diffie-Hellman) entre les deux parties en communication au lancement de la connexion. Ce secret partagé est ensuite utilisé pour protéger (un ensemble de) des clés de média qui sont utilisées pour chiffrer les sessions de média (RTP).

Le profil de sécurité pour le chiffrement vocal est une amélioration facultative du profil de sécurité de base et du profil de sécurité avec signature; son utilisation peut être négociée dans le contexte de la négociation des capacités de sécurité des terminaux. Dans les environnements où la confidentialité vocale est assurée par d'autres moyens, il n'est pas nécessaire d'implémenter le chiffrement de média et les procédures de gestion de clés correspondantes (concordance de clés Diffie-Hellman, mise à jour et synchronisation de clés).

Les algorithmes de chiffrement choisis sont les suivantes: AES, compatible-RC2, DES et triple-DES.

NOTE 2 – Comme une implémentation de l'algorithme triple-DES peut aussi être utilisée pour l'algorithme DES, cela permet d'obtenir une implémentation compacte.

Indépendamment du choix de l'algorithme de chiffrement de média spécifique, les options ci-après doivent être suivies explicitement:

- si nécessaire, génération d'un vecteur d'initialisation (IV) comme spécifié au § 9.3.1;
- si nécessaire, remplissage comme indiqué au § 9.3.2.

La charge utile audio doit être chiffrée au moyen de l'algorithme de chiffrement négocié ("X", "Y", "Z3" ou "Z") conformément aux procédures décrites aux § 9 et 9.3 et aux méthodes de bourrage cryptographique du § I.1. La charge utile audio peut être chiffrée au moyen de l'algorithme de chiffrement négocié ("X1", "Y1", "Z1" ou "Z2") fonctionnant dans un mode de chiffrement par flux (EOFB).

7 Signalisation et procédures H.245

En général, les aspects relatifs au secret des communications des canaux de média sont commandés de la même façon que tout autre paramètre de codage: chaque terminal indique ses capacités, l'émetteur des données choisit un format à utiliser et le récepteur acquitte ou refuse le mode. Tous les aspects du mécanisme qui sont indépendants du transport, comme la sélection de l'algorithme, sont indiqués par des éléments génériques de canal logique. Les caractéristiques de transport telles que la synchronisation des algorithmes de chiffrement ou des clés sont acheminées dans des structures propres à la couche Transport.

7.1 Fonctionnement avec canal H.245 sécurisé

En supposant que les procédures de connexion indiquent un mode de fonctionnement sécurisé, la prise de contact avec négociation et l'authentification doivent être effectuées pour le canal de commande H.245 avant l'échange d'éventuels autres messages H.245. S'il a été négocié, l'échange de certificats doit se faire au moyen d'un mécanisme approprié pour les terminaux conformes à la série H. Après sécurisation du canal H.245, les terminaux utilisent le protocole H.245 comme ils le feraient en mode non sécurisé.

7.2 Fonctionnement avec canal H.245 non sécurisé

En variante, le canal H.245 peut fonctionner en mode non sécurisé et les deux entités ouvrent un canal logique sécurisé avec lequel l'authentification et le calcul du secret partagé sont effectués. Par exemple, le protocole TLS (RFC 2246, RFC 3546) ou IPsec (RFC 2401) peut être utilisé par l'ouverture d'un canal logique dont le champ **dataType** contenant une valeur pour le paramètre **h235Control**. Ce canal pourra ensuite être utilisé pour calculer un secret partagé protégeant d'éventuelles clés de session de média ou pour transporter le champ **EncryptionSync**.

7.3 Echange de capacités

Conformément aux procédures indiquées au § 5.2/H.245 (Procédures d'échange de capacités) et conformément à la Recommandation de la série H applicable au système, les points d'extrémité échangent leurs capacités au moyen de messages H.245. Ces ensembles de capacités peuvent maintenant contenir des définitions indiquant des paramètres de sécurité et de chiffrement. Par exemple, un point d'extrémité peut signaler des capacités d'émission et de réception de signaux vidéo H.261, normales ou chiffrées.

Chaque algorithme de chiffrement utilisé avec un codec de média particulier implique une nouvelle définition de capacité. Comme pour toute autre capacité, les points d'extrémité peuvent indiquer, au cours de leur échange de capacités, des codecs chiffrés aussi bien indépendants que dépendants. Cela permettra aux points d'extrémité de dimensionner leurs capacités de sécurité en fonction des préfixes et des ressources disponibles.

Une fois l'échange de capacités effectué, les points d'extrémité peuvent ouvrir des canaux logiques sécurisés pour média de la même façon qu'ils le feraient en mode non sécurisé.

7.4 Rôle de maître

La relation maître-esclave H.245 est utilisée pour établir l'entité maîtresse en vue du fonctionnement en canal bidirectionnel et de la résolution d'autres conflits. Ce rôle de maître est également utilisé dans les méthodes de sécurité. Bien que le ou les modes de sécurité d'un flux de média soient fixés par l'émetteur (en fonction des capacités du récepteur), le maître est le point d'extrémité qui produit la clé de chiffrement. Cette production est effectuée sans tenir compte du fait que le maître est le récepteur ou l'émetteur du média chiffré. Pour permettre le fonctionnement de canaux à destinations multiples avec clés partagées, le pont (qui est également le maître) doit normalement produire les clés.

7.5 Signalisation par canal logique

Les points d'extrémité ouvrent des canaux logiques de média en mode sécurisé de la même façon qu'ils le feraient pour des canaux logiques de média en mode non sécurisé. Chaque canal peut fonctionner de manière totalement indépendante des autres canaux – en particulier pour ce qui est de la sécurité. Le mode particulier doit être défini dans le champ **dataType** du message **OpenLogicalChannel**. La clé de chiffrement initiale doit être transmise dans le message **OpenLogicalChannel** ou **OpenLogicalChannelAck** selon la relation maître/esclave de l'expéditeur du message **OpenLogicalChannel**.

Le message **OpenLogicalChannelAck** doit faire office de confirmation du mode de chiffrement. Si le message **openLogicalChannel** n'est pas acceptable par le destinataire, la valeur **dataTypeNotSupported** ou **dataTypeNotAvailable** (condition transitoire) doit être renvoyée dans le champ de cause du message **OpenLogicalChannelReject**.

Au cours de l'échange protocolaire qui établit le canal logique, la clé de chiffrement doit être transmise du maître à l'esclave (sans tenir compte de l'expéditeur du message **OpenLogicalChannel**). Pour les canaux de média ouverts par un point d'extrémité (autre que le maître), le maître doit renvoyer la clé de chiffrement initiale et le point de synchronisation initial dans le message **OpenLogicalChannelAck** (dans le champ **encryptionSync**). Pour les canaux de média ouverts par le maître, le message **OpenLogicalChannel** doit comporter la clé de chiffrement initiale et le point de synchronisation dans le champ **encryptionSync**.

7.6 Sécurité avec connexion rapide

Les points d'extrémité peuvent mettre en œuvre la procédure de connexion rapide (voir les § 8.1.7 et 8.1.7.1/H.323) en utilisant l'élément de démarrage rapide pour échanger en toute sécurité les données de clé (clé maîtresse et clés de chiffrement de session). Les procédures définies au § 7.6.1

décrivent le démarrage rapide "simple" ne reposant pas sur plusieurs algorithmes de chiffrement proposés; quant au § 7.6.1.1, il décrit le cas particulier du démarrage rapide avec plusieurs algorithmes de chiffrement proposés, permettant un codage plus condensé des messages.

7.6.1 Sécurité unidirectionnelle avec démarrage rapide

Cette procédure décrit la façon d'établir un canal logique de sécurité unidirectionnel (semi-duplex) entre l'appelant et l'appelé.

Procédure au niveau de l'appelant

L'appelant (émetteur du message SETUP) présente à la fois son jeton DH et les structures FastStart qu'il prend en charge. Le jeton DH doit être acheminé dans un champ ClearToken imbriqué dans un champ CryptoToken, ou sous la forme d'un champ ClearToken séparé (voir aussi le § 7.8). Pendant la phase SETUP à CONNECT doit avoir lieu un échange Diffie-Hellman (DH), qui confère aux deux points d'extrémité un secret partagé. Le champ **ClearToken** des champs **CryptoToken** doit contenir un champ **dhkey**, utilisé pour transmettre les paramètres comme indiqué dans la présente Recommandation. Le champ **halfkey** contient la clé publique aléatoire d'un participant, le champ **modsize** contient DH-prime et le champ **generator** contient DH-group. Les paramètres DH à utiliser sont indiqués dans le Tableau 4. Pour plus de détails, se référer à [RFC2412], Appendice E2.

NOTE 1 – Les messages H.225.0 étant authentifiés (comme décrit précédemment dans la procédure I), l'échange DH est un échange authentifié.

Dans l'un ou l'autre sens avec un message de signalisation d'appel H.225.0 acheminant une demi-clé Diffie-Hellman, lorsque l'information d'identification est disponible, l'appelant ou l'appelé doit également, au moment de son enregistrement, inclure un jeton **ClearToken** de bout en bout séparé avec le champ **sendersID** mis à l'identificateur de point d'extrémité de l'expéditeur et le champ **tokenOID** mis à "E". Toute entité de signalisation H.323 intermédiaire doit retransmettre ce jeton particulier de bout en bout sans modification.

Les structures FastStart indiquent les canaux logiques ouverts offerts avec les capacités de sécurité proposées. Les deux canaux H235Cap et nonH235Cap devraient être proposés. Au cours de l'échange de capacités H.245, les points d'extrémité présentent des entrées **H235SecurityCapability** pour les codecs qu'ils prennent en charge. Chaque codec est associé à une capacité de sécurité H.235 individuelle. Conformément au Tableau 6, ces capacités devraient indiquer la prise en charge de l'algorithme AES à 128 bits en mode CBC (OID – "Z3"), de l'algorithme RC2 à 56 bits compatible en mode CBC (OID – "X") ou de l'algorithme DES à 56 bits en mode CBC (OID – "Y") et peuvent indiquer la prise en charge de l'algorithme triple-DES à 168 bits en mode CBC (OID – "Z"), de l'algorithme triple-DES à 168 bits en mode EOFB (OID – "Z1"), de l'algorithme RC2 compatible en mode EOFB (OID – "X1"), de l'algorithme DES en mode EOFB (OID – "Y1") ou de l'algorithme AES en mode EOFB (OID – "Z2").

OpenLogicalChannel achemine à la fois **forwardLogicalChannelParameters** et **reverseLogicalChannelParameters** avec **dataType** fournissant **h235Media** avec **encryptionAuthenticationAndIntegrity** où un **MediaEncryptionAlgorithm** au maximum doit être présent dans **encryptionCapability**.

Pour les besoins de la relation de sécurité, l'appelé est le maître *a priori*, voir également le § 7.4.

L'appelant devrait mettre l'élément **mediaWaitForConnect** à "true", pour s'assurer que les données de clé de session sont disponibles et que le média chiffré reçu peut être déchiffré. Dans les scénarios où l'on souhaite un "média sans délai", c'est-à-dire où l'appelé transmet un média chiffré ou non chiffré en même temps qu'il envoie le message de réponse et les données de clé de chiffrement, l'appelant doit en principe savoir qu'il ne pourra déchiffrer le contenu à moins de disposer des données de clé.

NOTE 2 – Dans ce cas, si l'appelé transmet un média chiffré à l'appelant, ce qu'il peut faire théoriquement étant donné qu'il dispose des adresses RTP/RTCP de l'appelant, l'appelant ne sera pas en mesure de le déchiffrer sans le secret partagé fourni par le message Connect (*Alerting, Call Proceeding*).

Procédure au niveau de l'appelé

Pendant la phase de démarrage rapide (FastStart), l'appelé présente son jeton DH (voir également le § 7.8) et les structures FastStart qu'il prend en charge. Si la procédure Diffie-Hellman est appliquée, il est recommandé à l'appelé de renvoyer son jeton DH dans le message de réponse dans les plus brefs délais; c'est-à-dire, dans le message de réponse qui suit immédiatement le message SETUP. Ceci permet à l'appelant de calculer la clé maîtresse à partir du secret partagé DH et d'être prêt à recevoir la clé de session et le média chiffré.

NOTE 3 – Dans le cas où il n'y a pas d'algorithme de chiffrement disponible des deux côtés, le flux de média peut rester non chiffré ou la connexion peut être interrompue, en fonction de la politique de sécurité.

Chaque entité doit prendre les bits de faible poids appropriés provenant du secret Diffie-Hellman partagé commun pour la clé de chiffrement principale (clé maîtresse); c'est-à-dire les 56 bits de plus faible poids du secret Diffie-Hellman pour les identificateurs OID "X", "X1", "Y1" ou "Y" et les 168 bits de plus faible poids provenant du secret Diffie-Hellman pour les identificateurs OID "Z", "Z1" ou "Z2" et les 128 bits de plus faible poids du secret Diffie-Hellman pour les identificateurs OID "Z3" ou "Z2", voir également le Tableau 6.

Les réponses **OpenLogicalChannel(Ack)** sont émises avec la clé de session (maîtresse) créée et introduite dans le champ **encryptionSync**. Ce champ contient la clé de session pour le canal logique appelant→appelé. Le transport de la clé doit s'effectuer conformément à la procédure décrite au § 8.3 en utilisant les éléments **KeySyncMaterial** ou **V3KeySyncMaterial** (voir le § 8.3.1). La clé de session doit être chiffrée au moyen du secret partagé DH de la manière décrite ci-dessous.

NOTE 4 – Il n'y a pas de méthode prescrite pour produire les clés de session qui sont utilisées pour chiffrer le média. La production de ces valeurs est une question d'implémentation qui dépend des ressources locales, de la politique et de l'algorithme de chiffrement à utiliser. Il convient de prendre garde d'éviter de produire des clés faibles.

La clé de session chiffrée doit être acheminée dans le champ **H.235Key**/**sharedSecret** du champ **encryptionSync** au moyen de la procédure définie au § 8.3. La clé de session doit être acheminée dans le champ **keyMaterial** de **KeySyncMaterial**. Si sa taille n'est pas un multiple de la taille de bloc, elle doit être complétée par bourrage pour que sa taille soit un multiple de la taille de bloc avant chiffrement. La valeur du bourrage devrait être déterminée par la convention normale de l'algorithme de chiffrement. L'élément **KeySyncMaterial** (qui a fait l'objet d'un bourrage) sera chiffré en utilisant:

– 56 bits du secret partagé, en commençant par les bits de faible poids du secret Diffie-Hellman pour l'identificateur OID "X", "X1", "Y1" ou "Y";
– tous les bits du secret partagé pour l'identificateur OID "Z2", "Z" ou "Z1", en commençant par les bits de faible poids du secret DH.

Mais lorsque c'est possible, il est préférable d'utiliser le mécanisme de transport de clé amélioré conformément au § 8.3.1, compte tenu du résultat de la procédure d'indication définie pour la version 3 (voir le § 8.2).

Dans le cas où un canal de média sécurisé en duplex intégral pris parmi deux canaux unidirectionnels doit être établi en utilisant la procédure de démarrage rapide, l'appelé doit ouvrir un deuxième canal logique vers l'appelant. Ce canal logique sera signalé dans un élément fastStart séparé. En utilisant le secret partagé DH disponible comme clé maîtresse, l'appelé inclut une clé de session différente pour ce canal logique dans **encryptionSync**.

7.6.1.1 Utilisation d'algorithmes de chiffrement multiples dans la procédure de connexion rapide

La négociation du chiffrement de média dans le cadre de procédures de connexion rapide conduit à une expansion inefficace du nombre d'éléments **OpenLogicalChannel** dans l'élément **fastConnect** d'un message SETUP. Cela se produit car un élément **OLC** (**OpenLogicalChannel**) distinct est requis pour chaque combinaison de codec (**dataType**) et d'algorithme de chiffrement (y compris "aucun").

L'algorithme de chiffrement à appliquer à un flux de média est spécifié par l'inclusion du champ **dataType.h235Media.encryptionAuthenticationAndIntegrity.encryptionCapability dataType** dans l'élément **OLC**. La pratique définie dans la Rec. UIT-T H.235v2 consiste à n'inclure qu'un seul élément **MediaEncryptionAlgorithm** dans l'élément **encryptionCapability**, bien que ce dernier élément soit défini comme une séquence des éléments précédents. Cette procédure permet l'inclusion d'une séquence par ordre de préférence de capacités de chiffrement dans chaque élément **OLC** proposé. Le récepteur de l'élément **OLC** doit alors sélectionner un seul algorithme parmi ceux qui lui sont proposés et renvoyer l'élément **OLC** contenant le seul algorithme choisi (avec les adresses de transport appropriées et les données de clé de chiffrement).

Afin d'obtenir une efficacité maximale, l'identificateur d'objet "NULL-ENCR" (voir Tableau 2) représente l'algorithme de chiffrement "néant" (Null), ce qui signifie qu'aucune opération de chiffrement ne doit avoir lieu. L'utilisation de cette méthode particulière nécessite seulement un élément **OLC** par codec proposé et par sens.

Tableau 2/H.235.6 – Identificateur d'objet pour le chiffrement NULL

Désignation de l'identificateur d'objet	Valeur de l'identificateur d'objet	Description
"NULL-ENCR"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 26}	Indique "l'algorithme de chiffrement NULL"

Procédure applicable à l'appelant (voir le § 8.1.7.1/H.323)

Si un élément **dataType** proposé spécifie un chiffrement via le choix **h235Media**, l'élément **encryptionAuthenticationAndIntegrity** inclus peut inclure un élément **encryptionCapability** contenant plusieurs algorithmes de chiffrement (y compris l'algorithme NULL). Cette structure doit être utilisée pour offrir un choix parmi les algorithmes spécifiés pour le chiffrement de la capacité de média associée.

Procédure applicable à l'appelé (voir le § 8.1.7.1/H.323)

Si plusieurs algorithmes de chiffrement sont proposés pour un canal donné, le point d'extrémité appelé doit en choisir un et modifier l'élément **OpenLogicalChannel** pour éliminer les autres.

7.6.2 Sécurité bidirectionnelle avec démarrage rapide

La sécurité des canaux de données T.120 bidirectionnels appelle un complément d'étude.

7.7 Signaux DTMF H.245 chiffrés

Les points d'extrémité peuvent choisir d'envoyer des signaux DTMF (RFC 2833) chiffrés pour obtenir une certaine confidentialité. Au moyen de la clé de chiffrement de session, les points d'extrémité peuvent chiffrer les signaux DTMF (RFC 2833) dans **UserInputIndication** sous forme de:

- chaîne de base chiffrée: **encryptedAlphanumeric**;

- chaîne iA5 chiffrée: **encryptedSignalType** dans **signal**;
- chaîne générale chiffrée: **encryptedAlphanumeric** dans **extendedAlphanumeric**.

NOTE 1 – Les paramètres additionnels pour le protocole RTP dans la chaîne iA5 avec des horodates et des numéros de canaux logiques ou la mise à jour de signal avec la durée des tonalités ne sont pas chiffrés, étant donné qu'ils sont considérés comme n'acheminant pas des informations sensibles.

La capacité négociée **secureDTMF** se rapporte à une chaîne iA5 chiffrée.

Il convient d'appliquer la gestion de clés telle que spécifiée dans le § 6.1 pour obtenir une clé de chiffrement de session. Cette clé doit être utilisée pour chiffrer les signaux DTMF (RFC 2833) H.245.

NOTE 2 – Cela n'implique pas nécessairement l'utilisation de la clé de session également pour le chiffrement de charge utile RTP.

Toutefois, lorsqu'on utilise également la DTMF (RFC 2833) via le protocole RTP en mettant à 1 le fanion **rtpPayloadIndication**, il est fortement recommandé de sécuriser la charge utile RTP en utilisant le profil pour le chiffrement vocal décrit au § 6.1.

Le Tableau 3 indique les algorithmes de chiffrement disponibles (DES, 3DES ou AES) qui devraient mettre en œuvre le mode EOFB (incluant le mode OFB comme cas particulier, voir le § 8.4). Pour éviter le bourrage potentiel des caractères DTMF (RFC 2833), les modes CBC, CFB et autres modes de chaînage de blocs qui peuvent rendre le bourrage nécessaire ne sont pas recommandés pour le chiffrement des signaux DTMF (RFC 2833).

7.7.1 Chaîne de base chiffrée

Si **encryptedBasicString** dans **UserInputCapability** a été sélectionné, **encryptedAlphanumeric** doit indiquer l'algorithme de chiffrement appliqué dans le champ **algorithmOID**, l'élément **paramS** contient la valeur initiale pour l'opération de chiffrement. La chaîne alphanumérique chiffrée doit être placée dans **encrypted**.

7.7.2 Chaîne iA5 chiffrée

Si **encryptedIA5String** dans **UserInputCapability** a été sélectionné, **encryptedSignalType** doit contenir le type **ClearSignalType** chiffré dans lequel **sig** contient le caractère **signalType** en clair. **signalType** doit contenir un "!" fictif qui doit être supprimé par le destinataire.

Le champ **algorithmOID** doit indiquer l'algorithme de chiffrement appliqué, **paramS** contient la valeur initiale pour l'opération de chiffrement.

7.7.3 Chaîne générale chiffrée

Si **encryptedGeneralString** dans **UserInputCapability** a été sélectionné, l'élément **encryptedAlphanumeric** contenu dans **extendedAlphanumeric** doit indiquer l'algorithme de chiffrement appliqué dans le champ **algorithmOID** tandis que **alphanumeric** doit contenir une chaîne vide et **paramS** contient la valeur initiale pour l'opération de chiffrement.

7.7.4 Liste des identificateurs d'objet

Tableau 3/H.235.6 – Identificateurs d'objet pour le chiffrement de signaux DTMF H.245

Désignation de l'identificateur d'objet	Valeur de l'identificateur d'objet	Description
"DES-EOFB-DTMF"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 12}	Chiffrement de signaux DTMF H.245 avec l'algorithme DES-56 en mode EOFB
"3DES-EOFB-DTMF"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 13}	Chiffrement de signaux DTMF H.245 avec l'algorithme 3DES-168 en mode EOFB
"AES-EOFB-DTMF"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 14}	Chiffrement de signaux DTMF H.245 avec l'algorithme AES-128 en mode EOFB

7.8 Fonctionnement en mode Diffie-Hellman

La présente Recommandation utilise le protocole Diffie-Hellman pour la concordance de clés de bout en bout. En fonction de la situation, la clé Diffie-Hellman négociée peut jouer le rôle de clé maîtresse (§ 6.1) ou de clé dynamique de session (Recommandations UIT-T H.235.3 et H.530).

Le système Diffie-Hellman est caractérisé par les paramètres de système g et p où p est un grand nombre premier et g désigne le générateur du groupe multiplicatif modulo p ou d'un sous-groupe fort modulo p. $g^x \bmod p$ désigne la demi-clé (publique) Diffie-Hellman de l'appelant tandis que $g^y \bmod p$ définit la demi-clé (publique) Diffie-Hellman de l'appelé. La norme RFC 2412 donne d'autres informations contextuelles et des conseils sur la manière de choisir des paramètres de Diffie-Hellman sécurisés.

La Rec. UIT-T H.235.0 achemine une instance de Diffie-Hellman (g, p, g^x) codée dans un jeton **ClearToken** où **dhkey** contient la demi-clé **halfkey** $g^x \bmod p$ (respectivement $g^y \bmod p$) pour un certain secret x aléatoire (respectivement y), le nombre premier p dans **modsize** et le générateur **generator** g. Un cas particulier est le triplet (0, 0, 0) ou un champ **dhkey** vide qui ne représente pas d'instance DH mais qui doit être utilisé pour signaler que le profil pour le chiffrement vocal n'est pas utilisé.

Bien souvent, les paramètres p et g du système DH sont fixes pour un ensemble d'applications avec des valeurs bien définies, encore que les systèmes d'extrémité peuvent également choisir leur propre ensemble de paramètres. L'appelé devrait savoir que des paramètres DH non standards peuvent offrir une sécurité inférieure à ce que les paramètres semblent montrer à première vue; par exemple, l'appelant peut avoir choisi un nombre qui n'est pas premier, ou g génère juste un sous-groupe plus petit. Alors qu'un test exhaustif des paramètres n'est pas réalisable dans la pratique, c'est la politique de sécurité de l'appelé qui détermine l'acceptation ou le rejet de telles offres.

Pour les paramètres de système DH fixes, une caractérisation simplifiée via un identificateur d'objet peut conduire à des messages codés plus condensés que ceux qui incluent des valeurs littérales. Un jeton **ClearToken** qui achemine une instance DH avec des paramètres DH fixes, normalisés, peut faire référence à l'instance DH avec un identificateur DH-OID dans le champ **tokenOID**, à moins que le champ **tokenOID** ne soit utilisé à d'autres fins (comme décrit au § 7/H.235.1 pour un jeton **CryptoToken** distinctif). L'expéditeur a la possibilité d'inclure en plus les valeurs DH littérales mais cela n'est pas nécessaire.

Dans le cas où plusieurs instances DH doivent être indiquées, chacune via un identificateur DH-OID, il faut omettre les paramètres DH figurant dans le jeton **CryptoToken** distinctif (qui est

occupé par le profil H.235.1) en ne faisant pas figurer le champ **dhkey** et toutes les instances DH doivent être acheminées dans des jetons **ClearTokens** séparés où le champ **tokenOID** contient l'identificateur DH-OID et le champ **dhkey** peut être absent; tous les autres champs de ce jeton **ClearToken** ne doivent pas être utilisés.

NOTE 1 – Cela n'exclut pas la possibilité d'acheminer une instance DH dans un jeton **CryptoToken** distinct ou autres jetons **ClearTokens** disponibles en incluant littéralement les valeurs des paramètres DH.

Dans le cas où une instance DH non standard doit être indiquée, l'identificateur DH-OID "DHdummy" doit être utilisé et des paramètres de groupe DH non standards doivent être explicitement fournis dans le **ClearToken**.

L'appelant peut soumettre un ou plusieurs **ClearTokens** acheminant chacun une instance différente Diffie-Hellman. L'appelant est incité à fournir le plus grand nombre d'instances DH possible permis par sa politique de sécurité. L'appelé peut ainsi choisir une instance appropriée pour la réponse, ce qui accroît la probabilité de trouver un ensemble de paramètres communs applicables.

L'appelé doit choisir et accepter une seule instance DH (ou aucune) à partir de l'ensemble non ordonné d'instances DH fourni par l'appelant dans le message SETUP. Au cas où il est en mesure de choisir une instance DH qui correspond à ses propres besoins de sécurité, l'appelé ne doit pas modifier une instance DH proposée ou en renvoyer une qui n'a pas été envoyée par l'appelant. La force des algorithmes de chiffrement dont les deux points d'extrémité disposent pendant l'appel devrait correspondre à la force fournie par l'instance DH retenue qui est renvoyée par l'appelé, voir le Tableau 4. L'appelé doit indiquer l'instance DH retenue dans le message de réponse.

Dans le cas où l'appelé rejette toutes les propositions pour des raisons de sécurité ou en raison de l'insuffisance des capacités de traitement, il ne doit pas faire figurer le champ **dhkey** dans le message de réponse.

L'appelé doit inclure son jeton DH dans la réponse entre SETUP et CONNECT. L'appelé peut inclure son jeton DH dans le message de réponse qui suit immédiatement SETUP, ou peut l'inclure à un stade ultérieur, mais au plus tard dans le message CONNECT.

NOTE 2 – Il y a plusieurs aspects à prendre en considération concernant le moment où l'appelé peut inclure le ou les jetons DH pendant les réponses entre SETUP et CONNECT: le temps de réponse, la charge de traitement à laquelle est soumis l'appelé, la capacité de média sans délai, etc. Ces questions sont considérées comme dépendantes de l'implémentation.

Pour certaines raisons cependant, certains portiers routeurs peuvent ne pas délivrer toutes les réponses entre SETUP et CONNECT à l'appelant. Ainsi, un ou plusieurs messages de réponse de signalisation d'appel H.225.0 incluant un éventuel jeton DH peuvent être éliminés et ne parviendront pas à l'appelant. L'appelant ne sera pas alors en mesure de calculer la clé maîtresse DH et la ou les clés de session de média. Pour éviter cela, l'appelé devrait toujours inclure le même jeton DH dans chaque message de réponse entre SETUP et CONNECT.

Dans les cas où l'identificateur DH-OID indique une instance DH différente de celle qui est actuellement acheminée dans **modsize** et **generator**, les valeurs littérales acheminées dans **modsize** et **generator** doivent avoir la priorité sur l'identificateur DH-OID dans le jeton. Pour la réponse, l'appelé devrait remplacer l'identificateur DH-OID qui pose problème par l'identificateur DH-OID statique, par exemple "DH1024", qui correspond aux champs **modsize** et **generator** ou "DHdummy" s'il n'existe pas d'identificateur DH-OID correspondant.

7.8.1 Demande de renégociation des paramètres DH en milieu d'appel

Un portier H.323 peut demander la renégociation des paramètres DH en milieu d'appel au moyen des procédures définies dans le présent paragraphe. Cette renégociation peut être nécessaire pour établir la concordance de clés DH entre un point d'extrémité déjà connecté au portier et un point d'extrémité à connecter (voir la Figure 1). La renégociation des paramètres Diffie-Hellman est nécessaire pour la prise en charge de plusieurs services supplémentaires. Toutes les procédures

définies dans le présent paragraphe doivent être effectuées uniquement lorsque les points d'extrémité H.323 sont dans l'état de "pause côté émetteur", défini au § 8.4.6/H.323.

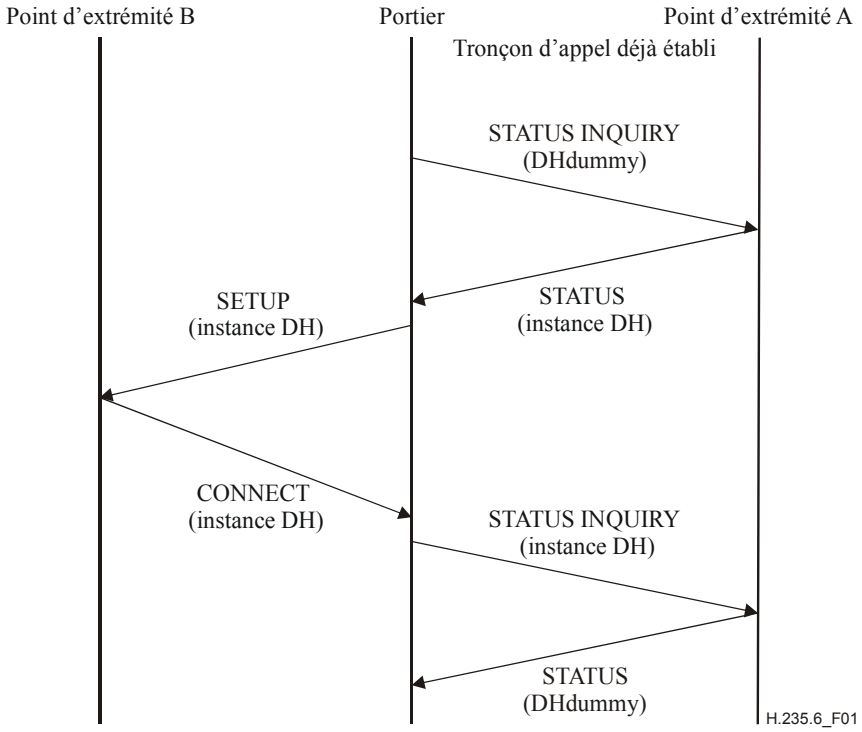


Figure 1/H.235.6 – Utilisation de la "demande de paramètres DH en milieu d'appel" pour les services supplémentaires

Pour demander des paramètres DH en milieu d'appel, une entité H.323 doit envoyer un message STATUS INQUIRY contenant un champ **ClearToken** avec le champ **tokenOID** mis à l'identificateur DH-OID "DHdummy", les autres champs étant omis.

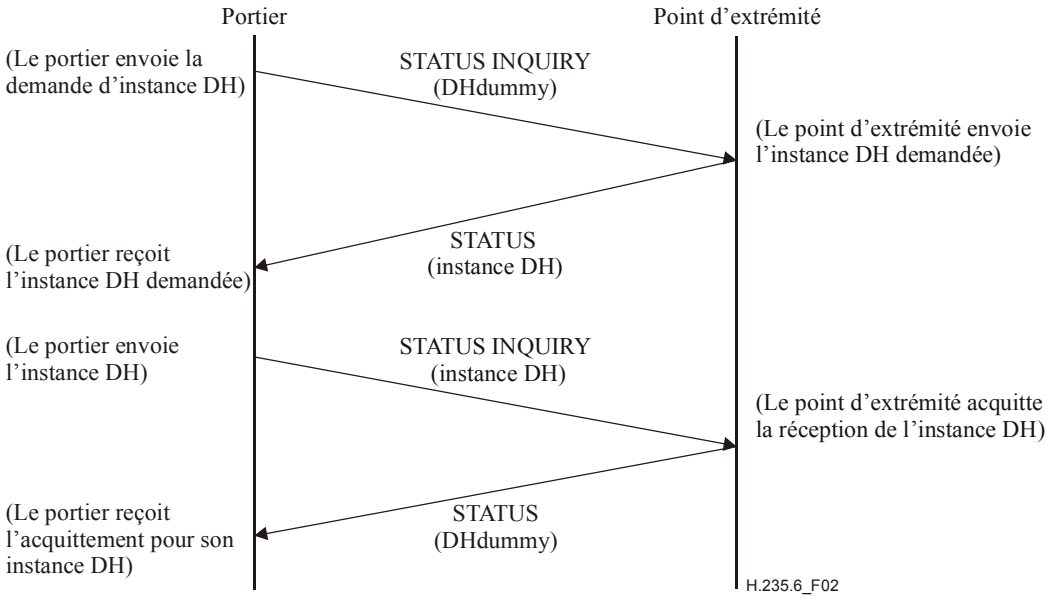


Figure 2/H.235.6 – Demande de paramètres DH en milieu d'appel

Si un point d'extrémité H.323 reçoit le message STATUS INQUIRY contenant un champ **ClearToken** avec le champ **tokenOID** mis à l'identificateur DH-OID "DHdummy", il doit répondre par un message STATUS contenant l'ensemble des instances DH (voir la Figure 2). Pour spécifier les instances DH dans ce message STATUS, il faut suivre les règles définies au § 7.8 pour le message SETUP.

NOTE 1 – Une entité H.323 qui ne prend pas en charge cette procédure est supposée répondre au message STATUS INQUIRY par un message STATUS sans instance DH.

Pour acheminer l'instance DH acceptée en milieu d'appel, l'entité H.323 doit envoyer un message STATUS INQUIRY contenant l'instance DH acceptée (voir la Figure 2). Pour spécifier les instances DH dans ce message STATUS INQUIRY, il faut suivre les règles définies ci-dessus au § 7.8 pour la réponse au message SETUP.

Si un point d'extrémité H.323 reçoit un tel message STATUS INQUIRY contenant un champ **ClearToken** avec une instance DH, il doit répondre par un message STATUS contenant un champ **ClearToken** avec le champ **tokenOID** mis à l'identificateur DH-OID "DHdummy", les autres champs étant omis.

NOTE 2 – Une entité H.323 qui ne prend pas en charge cette procédure est supposée répondre au message STATUS INQUIRY par un message STATUS sans instance DH.

Un point d'extrémité H.323 recevant un message STATUS INQUIRY avec une instance DH doit recalculer le secret partagé DH à partir de cette instance DH et du dernier ensemble d'instance(s) DH qu'il a envoyé au cours de l'appel considéré.

Si un portier H.323 reçoit un message STATUS INQUIRY contenant un champ **ClearToken** avec une instance DH ou avec un champ **tokenOID** mis à l'identificateur DH-OID "DHdummy", il doit – sauf dans les cas décrits ci-dessous – retransmettre le message au deuxième tronçon de l'appel dans le contexte duquel le message a été reçu.

Si un portier H.323 reçoit une réponse STATUS au message STATUS INQUIRY qu'il a retransmis, il doit retransmettre le message STATUS au tronçon d'appel sur lequel le message STATUS INQUIRY a été reçu.

Si un portier H.323 qui attend une réponse au message STATUS INQUIRY contenant un champ **ClearToken** avec le champ **tokenOID** mis à l'identificateur DH-OID "DHdummy" qu'il a envoyé reçoit un message STATUS INQUIRY contenant un champ **ClearToken** avec le champ **tokenOID** mis à l'identificateur DH-OID "DHdummy" et avec le fanion CRV mis à 1, il doit répondre par un message STATUS contenant un champ **ClearToken** avec le champ **tokenOID** mis à l'identificateur DH-OID "DHdummy" (voir la Figure 3).

Si un portier H.323 reçoit un message STATUS INQUIRY contenant un champ **ClearToken** avec une instance DH ou avec le champ **tokenOID** mis à l'identificateur DH-OID "DHdummy" alors que le deuxième tronçon de l'appel n'est pas établi, il doit attendre l'établissement du deuxième tronçon de l'appel, envoyer un ensemble de capacités vide sur ce tronçon puis retransmettre le message STATUS INQUIRY reçu (voir la Figure 3).

Un portier H.323 ne doit pas lancer les procédures définies dans le présent paragraphe après avoir envoyé un message STATUS contenant une instance DH et avant d'avoir reçu le message STATUS INQUIRY contenant une instance DH.

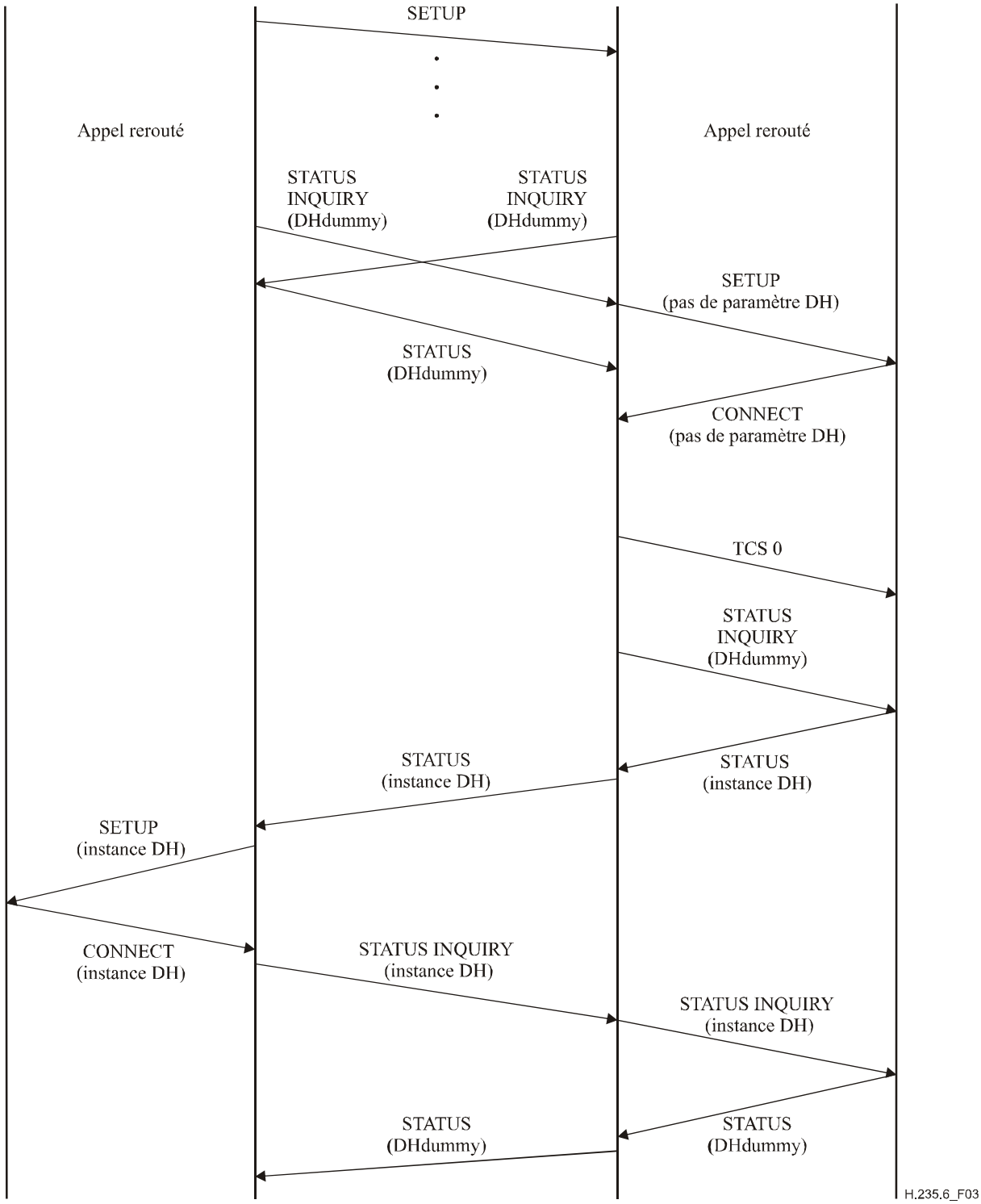


Figure 3/H.235.6 – Utilisation de la "demande de paramètres DH en milieu d'appel" pour un reroutage d'appel simultané par deux portiers

8 Signalisation et procédures

Les procédures décrites au § 8/H.323 (Procédures de signalisation d'appel) doivent être suivies. Les points d'extrémité H.323 doivent avoir la capacité de coder et de reconnaître la présence (ou l'absence) de prescriptions de sécurité (pour le canal H.245) signalées dans les messages H.225.0.

Si le canal H.225.0 lui-même doit être sécurisé, les mêmes procédures qu'au § 8/H.323 doivent être suivies. La différence de fonctionnement est que les communications ne doivent avoir lieu qu'après

connexion à l'identificateur de point TSAP sécurisé et au moyen des modes de sécurité prédéterminés (TLS RFC 2246, RFC 3546, par exemple). Etant donné que les messages H.225.0 sont les premiers échangés lors de l'établissement de communications H.323, il ne peut pas y avoir de négociation de sécurité "dans la bande" pour les messages H.225.0. En d'autres termes, les deux parties doivent savoir *a priori* qu'elles vont utiliser un mode de sécurité particulier. Pour les flux H.323 sur IP, un autre port bien connu (1300) est utilisé pour les communications sécurisées par la méthode TLS (RFC 2246, RFC 3546).

Un des objectifs des échanges H.225.0, en ce qui concerne la sécurité H.323, est d'offrir un mécanisme permettant d'établir le canal H.245 sécurisé. Facultativement, l'authentification peut se produire pendant l'échange de messages H.225.0. Cette authentification peut être fondée sur des certificats ou sur des mots de passe, avec chiffrement et/ou hachage (c'est-à-dire signature). Les particularités de ces modes de fonctionnement sont décrites aux § 8.1 à 8.2.3/H.235.0.

Un point d'extrémité H.323 qui reçoit un message SETUP avec la capacité **h245SecurityCapability** activée doit répondre en indiquant le mode acceptable correspondant (**h245SecurityMode**) dans le message CONNECT. Lorsqu'il n'y a pas de capacités correspondantes, le terminal appelé peut refuser la connexion en envoyant un message **Release Complete** avec le code de motif mis à ***SecurityDenied***. Cette erreur est destinée à n'acheminer aucune information sur une éventuelle discordance de sécurité: le terminal appelant devra déterminer le problème par un autre moyen. Lorsque le terminal appelant reçoit un message CONNECT sans mode de sécurité suffisant ou acceptable, il peut mettre fin à l'appel par un message **Release Complete** avec le motif ***SecurityDenied***. Lorsque le terminal appelant reçoit un message CONNECT sans aucune capacité de sécurité, il peut mettre fin à l'appel par un message **Release Complete** avec le motif ***undefinedReason***.

Si le terminal appelant reçoit un mode **h245Security** acceptable, il doit ouvrir et exploiter le canal H.245 dans le mode de sécurité indiqué. L'échec d'établissement du canal H.245 dans le mode de sécurité déterminé ici devrait être considéré comme une erreur de protocole et il devrait être mis fin à la connexion.

8.1 Compatibilité avec la Révision 1

Un point d'extrémité possédant des capacités de sécurité ne doit pas retourner de champs, d'indications ou d'états liés à la sécurité, à un point d'extrémité ne possédant pas de capacités de sécurité. Si un appelé reçoit un message SETUP qui ne contient pas de capacités de sécurité **H245Security** et/ou de jeton d'authentification, il peut retourner un message **Release Complete** afin de refuser la connexion; mais dans ce cas il doit utiliser le code de motif ***undefinedReason***. De manière analogue, si un appelant reçoit un message CONNECT sans indication **H245SecurityMode** ni/ou un jeton d'authentification alors qu'il a envoyé un message "d'établissement" SETUP avec **H245Security** et/ou un jeton d'authentification, il peut également mettre fin à la connexion en émettant un message **Release Complete** avec le code de motif ***undefinedReason***.

8.2 Indication de capacité de version 3

Les points d'extrémité de version 3 ou d'une version ultérieure de la Rec. UIT-T H.235 peuvent offrir des procédures de sécurité améliorées sur le trajet de média que n'offrent pas les versions 1 et 2. Ces procédures de sécurité améliorées sont les suivantes:

- transport de clé amélioré (**V3KeySyncMaterial,** voir le § 8.3.1);
- mise à jour de clé améliorée, voir le § 8.6.2.

Etant donné qu'en général, les points d'extrémité ne savent rien au sujet de leur prise en charge réciproque de la version 3 ou d'une version ultérieure de la Rec. UIT-T H.235, une indication explicite de la version est ajoutée pendant l'établissement d'appel.

Les points d'extrémité de version 3 ou d'une version ultérieure de la Rec. UIT-T H.235 devraient toujours utiliser la procédure décrite dans le présent paragraphe pour déterminer si les procédures de version 3 peuvent être utilisées (transport de clé amélioré, synchronisation de chiffrement amélioré). En fonction du résultat de la procédure de signalisation logique, les points d'extrémité peuvent utiliser les procédures (voir le § 8.3) assurant la rétrocompatibilité avec des points d'extrémité de version 1 ou 2 de la Rec. UIT-T H.235.

Pour indiquer si les procédures améliorées décrites dans la version 3 de la Rec. UIT-T H.235 peuvent être utilisées, les points d'extrémité appelant et appelé doivent inclure un jeton **ClearToken** additionnel indiquant la capacité de version 3 pendant la signalisation d'appel (SETUP, CONNECT, etc.). L'absence d'un tel jeton indiquera la prise en charge uniquement de la version 1 ou de la version 2 de la Rec. UIT-T H.235. Dans ce cas, les points d'extrémité doivent utiliser la procédure décrite au § 8.3. Dans les autres cas, ils peuvent utiliser les procédures améliorées décrites au § 8.3.1 ou utiliser la procédure de la version 1 ou de la version 2 de la Rec. UIT-T H.235 décrite au § 8.3.

Ce jeton **ClearToken** devra utiliser le champ **tokenOID** mis à "V3", dont la valeur est la suivante:

"V3"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 24}	Indicateur de capacité de version 3 dans un jeton ClearToken pendant la signalisation d'appel.

Tous les autres champs de ce jeton **ClearToken** devront rester inutilisés sauf s'ils sont utilisés pour acheminer les paramètres DH.

8.3 Transport de la clé

Le maître générera les données de clé de session et les distribuera à son ou à ses homologues. Deux procédures sont proposées pour le transport de la clé:

- une procédure principalement conçue pour les points d'extrémité version 1 ou version 2 de la Rec. UIT-T H.235 décrite dans le présent paragraphe;
- une procédure améliorée pour les points d'extrémité version 3 ou version ultérieure de la Rec. UIT-T H.235, décrite au § 8.3.1.

Les points d'extrémité version 1 ou version 2 de la Rec. UIT-T H.235 appliquent la procédure suivante pour le transport de la clé de session:

L'élément **KeySyncMaterial** contient l'identificateur de point d'extrémité du maître dans le champ **generalID** et achemine les données de clé de session dans **keyMaterial**. La valeur de **generalID** devrait être incluse pour offrir un niveau minimal d'authentification de l'émetteur de la clé de session (voir également le § 8.6). Le destinataire devrait vérifier que le champ **generalID** reçu est correct.

NOTE – Dans la présente Recommandation, on suppose que chaque point d'extrémité s'est enregistré auprès d'un portier et dispose d'un identificateur de point d'extrémité qui peut être acheminé dans le champ **generalID**. La présente Recommandation ne prend pas en charge les scénarios sans portier; ces scénarios appellent un complément d'étude.

L'élément **KeySyncMaterial** doit être chiffré en utilisant la clé maîtresse négociée. L'élément **KeySyncMaterial** doit toujours être complété par des informations de bourrage pour que sa longueur soit égale à un multiple de la longueur de bloc avant le chiffrement, le dernier octet devant contenir le nombre d'octets de bourrage (y compris le dernier). La valeur du bourrage devrait être déterminée par la convention normale de l'algorithme de chiffrement. Le résultat du chiffrement doit être stocké dans l'élément **sharedSecret** du champ **H235Key**.

8.3.1 Transport de clé amélioré dans la version 3 de la Rec. UIT-T H.235

On a observé que la définition en syntaxe ASN.1 de l'élément **KeySyncMaterial** et la façon dont l'opération ENCRYPTED{} est appliquée aux données dans les versions 1 et 2 de la Rec. UIT-T H.235 révèle un grand nombre de textes en clair connus, dont l'identificateur **generalID** du maître, mais également certains bits de codage connus pour la structure. L'identificateur **generalID**, même s'il est chiffré, est connu grâce à d'autres parties non chiffrées du message de signalisation (par exemple, **senderID**). On estime que la présence d'un tel texte en clair connu affaiblit considérablement le système de sécurité, de sorte qu'un attaquant peut très facilement casser la clé de session par la "force brute", en particulier pour un chiffrement par bloc qui utilise une petite longueur de bloc (DES-56 ou compatible-RC2, par exemple).

De plus, la version 3 de la Rec. UIT-T H.235 doit pouvoir permettre de transporter des données de clé additionnelles, à savoir:

- transport sécurisé d'une clé de salage vers le ou les homologues. Cette clé de salage est introduite pour le mode OFB amélioré; voir le § 8.4.

La version 3 de la Rec. UIT-T H.235 étend le champ **H235key** avec **secureSharedSecret** contenant l'élément **V3KeySyncMaterial** qui contient les paramètres suivants:

generalID contient l'identificateur de point d'extrémité de l'expéditeur s'il est disponible, dans les autres cas ce champ reste inutilisé.

algorithmOID indique l'algorithme de chiffrement appliqué et le mode de fonctionnement.

paramS contient la valeur d'initialisation qui est appliquée au chiffrement de la ou des clés acheminées.

NOTE 1 – Le vecteur IV à l'intérieur de **paramS** ne doit pas être confondu avec le paquet IV RTP qui n'est pas signalé. L'élément **ClearSalt** contient optionnellement une clé de salage non chiffrée pour le chiffrement de la clé de session (par exemple, pour le mode EOFB).

encryptedSessionKey contient le texte chiffré de la clé de session brute chiffrée.

encryptedSaltingKey contient le texte chiffré de l'éventuelle clé de salage de média brute chiffrée. La clé de salage est nécessaire pour le mode OFB amélioré.

clearSaltingKey peut contenir la clé de salage de média brute non chiffrée. Dans les implémentations, il faudra s'assurer que les éléments **encryptedSaltingKey** et **clearSaltingKey** ne sont pas utilisés simultanément.

paramSsalt contient la valeur initiale pour le chiffrement de la clé de salage. L'élément **ClearSalt** contient optionnellement une clé de salage non chiffrée pour le chiffrement de la clé de salage (par exemple, pour le mode EOFB).

NOTE 2 – **generalID**, **algorithmOID** et **paramS** sont toujours transmis en clair, tandis que **encryptedSessionKey**, **encryptedSaltingKey** contiennent le texte chiffré des données de clé chiffrée.

Le maître génère la ou les clés, conformément aux capacités négociées entre les terminaux, et envoie la ou les clés en utilisant **V3KeySyncMaterial** vers les points d'extrémité homologues. Ainsi, l'élément **V3KeySyncMaterial** doit être retransmis sans modification par les portiers intermédiaires lorsqu'il y en a.

Les points d'extrémité conformes à la version 3 ou à une version ultérieure de la Rec. UIT-T H.235 devraient toujours utiliser l'élément **secureSharedSecret** dans l'élément **H235Key**, mais en fonction du résultat de la procédure de signalisation logique du § 8.2, utilisant le jeton **ClearToken** de version 3, ces points peuvent utiliser **sharedSecret** pour assurer la rétrocompatibilité avec des points d'extrémité conformes à la version 1 ou 2 de la Rec. UIT-T H.235.

8.4 Mode OFB amélioré

Le mode OFB (ISO/CEI 10116) est un mode de fonctionnement qui met en œuvre un chiffrement par flux au moyen d'algorithmes de chiffrement par bloc. Le mode OFB offre:

- de bonnes performances grâce à un temps de traitement de chiffrement court;
- un traitement facile et peu complexe des blocs incomplets;
- une bonne résistance aux erreurs binaires.

Le mode OFB amélioré est un mode OFB légèrement modifié appelé ici mode EOFB (*enhanced output feedback*) qui présente les mêmes caractéristiques que le mode OFB avec en plus:

1) l'utilisation d'une clé de salage KS qui vient s'ajouter à la clé de chiffrement KE;
2) l'introduction d'un indice implicite de paquet.

L'utilisation d'une clé de salage secrète supplémentaire KS, qui est soumise à l'opération OU exclusif pour le bouclage, ajoute une sécurité supplémentaire par rapport à l'analyse de texte en clair connu. Il s'agit d'un avantage majeur du point de vue de la sécurité que les autres modes de fonctionnement normalisés (CBC, OFB, etc.) n'offrent pas. L'utilisation du mode EOFB conduit donc à une plus grande sécurité par rapport au texte en clair à forte redondance et également par rapport à l'analyse de texte en clair connu.

Le mode EOFB est défini par $C_i = P_i \oplus S_i$ avec $S_i = E_{KE}(KS \oplus S_{i-1})$ pour $i = 1 \ldots n$ et $S_0 = IV$ où C_i est le ième bloc de texte chiffré, P_i est le ième bloc de texte en clair; S_i est le ième bloc de flux de clés; KE la clé de chiffrement et $\oplus$ est un OU exclusif binaire. Le mode EOFB est illustré à la Figure I.6.

Le mode EOFB peut également fonctionner en mode OFB standard, ce qui rend le mode EOFB rétrocompatible avec le mode OFB. Dans ce cas où l'on souhaite une rétrocompatibilité avec le mode OFB standard, la clé de salage KS doit être constituée entièrement de 0 ou le champ **encryptedSaltingKey** doit être laissé vide dans l'élément **V3KeySyncMaterial**, ce qui revient au même. Toutefois, l'utilisation d'une clé de salage réelle est fortement recommandée lorsqu'il s'agit de chiffrer des charges utiles RTP avec un chiffrement par bloc utilisant une longueur de bloc courte (DES-56 ou compatible-RC2, par exemple).

Après qu'au plus 2^{48} paquets ont été traités, une nouvelle clé de chiffrement de session KE et une nouvelle clé de salage KS doivent être utilisées; dans le cas contraire, il y aura réutilisation du flux de clés, ce qui compromettra la sécurité.

Le paragraphe 11 définit les identificateurs d'objet pour l'algorithme DES-56 en mode EOFB, l'algorithme compatible-RC2 en mode EOFB, l'algorithme 3-DES en mode EOFB et l'algorithme AES en mode EOFB.

8.5 Gestion de clés

Les points d'extrémité conformes à la présente Recommandation devraient utiliser la procédure de connexion rapide conformément au § 7.6.1. Si le démarrage rapide n'est pas appliqué, il faut utiliser la tunnellisation H.245 pour sécuriser les messages de commande d'appel H.245 selon la présente Recommandation. Les procédures de démarrage rapide permettent d'établir un ou deux canaux logiques unidirectionnels, de négocier les capacités de sécurité, de distribuer un secret partagé commun (secret DH partagé) qui joue le rôle de clé maîtresse et de distribuer de façon sécurisée une clé de chiffrement.

Le Tableau 4 définit les identificateurs OID attribués pour divers algorithmes de chiffrement et établit une correspondance entre ces identificateurs et les identificateurs OID attribués au groupe Diffie-Hellman. Trois groupes DH sont identifiés par un OID:

- "DHdummy": une instance de ce groupe DH devrait être appliquée chaque fois qu'il s'agit d'une sécurité exportable (512 bits) ou qu'un groupe quelconque ou non standard est utilisé.

  NOTE 1 – Aucun groupe DH particulier n'est défini; l'OID désigne tout groupe DH non standard.
- Une instance de groupe DH à 512 bits doit être utilisée pour générer une clé maîtresse pour la distribution d'une ou de plusieurs clés de session pour les algorithmes de chiffrement compatible-RC2 ("X") ou DES à 56 bits ("Y").
- "DH1024": ce groupe DH est à utiliser en cas de sécurité élevée (1024 bits). L'OID désigne un groupe DH fixe et normalisé. Ce groupe DH doit être utilisé pour produire une clé maîtresse pour la distribution d'une ou de plusieurs clés de session pour les algorithmes de chiffrement 3-DES ("Z").
- "DH1536": ce groupe est proposé en option pour les points d'extrémité de version 3 avec des besoins de sécurité très élevés qui vont au-delà de la sécurité d'un groupe DH à 1024 bits. L'OID désigne un groupe DH fixe. Ce groupe DH doit être utilisé pour produire une clé maîtresse pour la distribution d'une ou de plusieurs clés de session pour les algorithmes de chiffrement 3-DES ("Z", "Z1") ou AES-128 ("Z2", "Z3").

Il est recommandé d'utiliser le groupe DH1024 et optionnellement le groupe DH1536 sauf si d'autres besoins de sécurité font que d'autres paramètres de Diffie-Hellman sont préférés. De plus, il est recommandé d'envisager l'utilisation des identificateurs OID identifiant les groupes DH (voir le § 7.8). Néanmoins, les implémentations devraient être prêtes à recevoir des paramètres de groupe DH littéralement sans indication OID explicite. Dans ce cas, il convient de s'assurer dans les implémentations, que le groupe DH correct est acheminé conformément au Tableau 4.

Les points d'extrémité peuvent utiliser des paramètres de groupe DH non standards. L'utilisation de l'identificateur OID "DHdummy" devrait permettre d'indiquer des groupes DH non standards. C'est à l'appelé qu'il appartient d'accepter ou de refuser des groupes DH de ce type.

NOTE 2 – Le choix du groupe DH ne supprime pas la nécessité de négocier l'algorithme réel de chiffrement de média. Cette négociation doit s'effectuer avec la procédure de négociation de capacité de terminal H.245.

NOTE 3 – Pendant la phase d'établissement de la connexion (SETUP à CONNECT), les identificateurs OID d'algorithme de chiffrement ne doivent pas être utilisés pour indiquer une instance Diffie-Hellman.

Tableau 4/H.235.6 – Groupes de Diffie-Hellman

OID de l'algorithme de chiffrement	DH-OID	Description de groupe D-H
"X", "X1" (compatible-RC2), "Y", "Y1" (DES)	"DHdummy"	Mod-P, tout nombre premier de 512 bits approprié
"Z", "Z1" (3-DES), "Z2", "Z3" (AES)	"DH1024"	Mod-P, nombre premier de 1024 bits Nombre premier = $2^{1024} - 2^{960} - 1 + 2^{64} \times \{ [2^{894} \text{ pi}] + 129093 \}$ = (179769313486231590770839156793787453197860296048756011706444423684197180216158519368947833795864925541502180565485980503646440548199239100050792877003355816639229553136239076508735759914822574862575007425302077447712589550957937778424442426617334727629299387668709205606050270810842907692932019128194467627007)$_{10}$ Générateur (Note) = 2

Tableau 4/H.235.6 – Groupes de Diffie-Hellman

OID de l'algorithme de chiffrement	DH-OID	Description de groupe D-H
"Z", "Z1" (3-DES), "Z2", "Z3" (AES)	"DH1536"	Mod-P, nombre premier de 1536 bits Nombre premier = $2^{1536} – 2^{1472} – 1 + 2^{64} \times \{ [2^{1406}$ pi] + 741804 } = (2410312426921032588552076022197566074856950548502459942654116941958108831682612228890093858261341614673227141477904012196503648957050582631942730706805009223062734745341073406696246014589361659774041027169249453200378729434170325843778659198143763193776859869524088940195577346119843545301547043747207749969763750084308926339295559968882457872412993810129130294592999947926365264059284647209730384947211681434464714438488520940127459844288859336526896320919633919)$_{10}$ Générateur (Note) = 2
NOTE – Le générateur est utilisé pour produire le jeton DH.		

8.6 Mise à jour et synchronisation des clés

Pour les chiffrements par blocs de 64 bits, la fréquence de rafraîchissement des clés *doit* être telle que pas plus de 2^{32} blocs soient chiffrés au moyen de la même clé. Il *convient* que les implémentations rafraîchissent les clés avant que 2^{30} blocs aient été chiffrés au moyen de la même clé (voir le § 9.1). Pour les chiffrements par blocs de 128 bits, la fréquence de rafraîchissement des clés *doit* être telle qu'au plus 2^{64} blocs soient chiffrés au moyen de la même clé. Il *convient* que les implémentations rafraîchissent les clés avant que 2^{62} blocs aient été chiffrés au moyen de la même clé (voir le § 9.1). Les deux entités concernées sont libres de changer la clé de session média aussi souvent qu'elles le jugent nécessaire compte tenu de leur politique de sécurité. Par exemple, le maître peut distribuer une nouvelle clé de session au moyen de **encryptionUpdate** ou de **encryptionUpdateCommand** du message **miscellaneousCommand**. Par ailleurs, l'esclave peut demander au maître une nouvelle clé de session au moyen de **encryptionUpdateRequest** du message **miscellaneousCommand**.

Le message **MiscellaneousCommand** contient **encryptionUpdate** et **encryptionUpdateCommand** dont le champ **encryptionSynch** comprend les paramètres suivants:

- **synchFlag**: le nouveau numéro de charge utile RTP dynamique indiquant un changement de clé.
- **h235key**: achemine la nouvelle clé de session chiffrée. Il s'agit de la clé **H235Key** codée en ASN.1 transmise sous forme de chaîne d'octets.

Le champ **sharedSecret** dans la structure **H235Key** utilise les champs suivants:

- **algorithmOID**: mis à "X", "X1" pour l'algorithme compatible-RC2 à 56 bits, à "Y", "Y1" pour la norme DES à 56 bits ou "Z", "Z1" pour l'algorithme 3-DES à 168 bits ou mis à "Z3" pour l'algorithme AES à 128 bits.

  NOTE 1 – L'algorithme de chiffrement de clé de session est le même que l'algorithme de chiffrement de média négocié.

- **paramS**: mis à la valeur initiale. Pour les chiffrements de flux par blocs de 64 bits, le champ **iv8** contient une séquence binaire formée d'un bloc de 64 bits aléatoires produit par l'initiateur. Pour les chiffrements de flux par blocs de 128 bits, le champ **iv16** contient une séquence binaire formée d'un bloc de 128 bits aléatoires produit par l'initiateur. Ce champ

ne doit pas être utilisé pour le mode CBC et sa valeur est NULL, ce qui signifie que le vecteur CBC-IV pour le chiffrement de la clé de session doit être mis à 0; il doit être uniquement utilisé pour acheminer le vecteur IV dans le mode EOFB.

- **encryptedData**: mis au résultat de **KeySynchMaterial** chiffré.

En tant que partie de **KeySyncMaterial**:

- **generalID**: identificateur de l'émetteur distribuant la clé.

  NOTE 2 – Dans la présente Recommandation, on suppose que chaque point d'extrémité s'est enregistré auprès d'un portier et a obtenu un identificateur de point d'extrémité qui peut être acheminé dans le champ **generalID**. Dans la présente Recommandation, on ne considère pas les scénarios sans portier, ces scénarios appellent un complément d'étude.

- **keyMaterial**: mis à la nouvelle clé de session. Pour les algorithmes DES et compatible-RC2, il s'agit d'une clé de 56 bits, pour l'algorithme 3-DES, d'une clé de 168 bits et pour l'algorithme AES, d'une clé de 128 bits. Le maître doit produire une nouvelle clé de session qui répond au moins aux critères de sécurité suivants: il ne s'agira pas d'une clé DES faible ou semi-faible et elle doit utiliser une source aléatoire suffisamment sûre.

Le message **MiscellaneousCommand** contient le champ **encryptionUpdateRequest** contenant **keyProtectionMethod** où le fanion **sharedSecret** est mis à TRUE.

NOTE 3 – Etant donné que la mise à jour et la synchronisation de la clé reposent sur des messages H.245 qui ne sont pas portés dans un autre message au cours de la connexion rapide, il faut utiliser une tunnellisation H.245 pour les entités H.323 sécurisées.

Les clés de session de média n'ont pas une durée de vie éternelle. A un certain instant, chaque clé de session expire. Une nouvelle clé de session doit alors être utilisée pour protéger une session sécurisée en cours. Dans des environnements de conférence, une nouvelle clé de session de groupe devrait être définie et distribuée lorsque des membres d'un groupe rejoignent ou quittent une conférence sécurisée, ce qui les empêche d'accéder aux données passées ou futures.

- Une mise à jour et une synchronisation de clé fondée sur le type de charge utile définit un nouveau type de charge utile dynamique pour cette nouvelle clé de session; voir les § 8.6.1, § 8.6.2 et § 8.6.3.

Pour la mise à jour des clés, la présente Recommandation propose une prise de contact sans accusé de réception qui est applicable également aux points d'extrémité des versions 1 et 2 de la Rec. UIT-T H.235 et aussi une prise de contact avec accusé de réception particulièrement fiable pour les points d'extrémité version 3 et versions ultérieures de la Rec. UIT-T H.235.

8.6.1 Mise à jour de clé sans accusé de réception

La Figure 4 illustre la phase de prise de contact sans accusé de réception pour la distribution/mise à jour de clé de session. Si l'esclave souhaite une clé de session mise à jour, il peut demander une nouvelle clé de session au maître en émettant une demande **encryptionUpdateRequest** au maître. Le maître envoie une nouvelle clé de session (avec ou sans demande préalable **encryptionUpdateRequest** de la part de l'esclave) à l'esclave dans un message **EncryptionUpdate**.

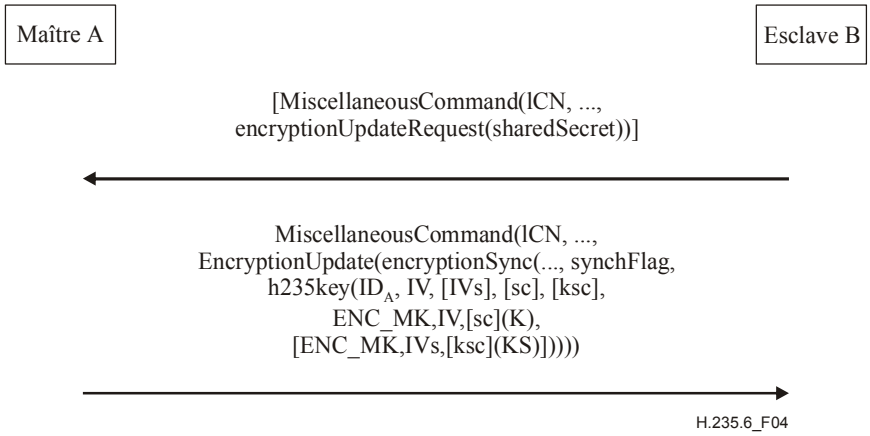


Figure 4/H.235.6 – Distribution/mise à jour de clé de session sans accusé de réception du maître vers le ou les esclaves

où:

ICN	est le numéro de canal logique;
synchFlag	est le numéro de charge utile RTP dynamique;
ID_A	est l'identificateur **generalID** de la source;
IV	est la valeur/le vecteur initial pour le chiffrement de la clé de session;
IVs	est la valeur/le vecteur initial pour le chiffrement de la clé de salage;
ENC_MK, IV, sc(K)	signifie le chiffrement d'un texte en clair *K* au moyen de la clé *M*, du vecteur initial *IV* [et d'une clé de salage *sc*, seulement pour le mode EOFB];
KS	est la clé de salage pour le média (pour le mode EOFB seulement);
K	est la clé de session en clair;
sc	est la clé de salage non chiffrée lorsque le mode EOFB est utilisé pour le chiffrement de la clé de session;
ksc	est la clé de salage non chiffrée lorsque le mode EOFB est utilisé pour le chiffrement de la clé de salage;
s2M/m2S	est le fanion **direction** (version 3 ou version ultérieure de la Rec. UIT-T H.235 seulement) entre (s2m = esclave vers maître, m2s = maître vers esclave);
[]	représente une partie optionnelle.

Les méthodes de mise à jour de clé telles que décrites dans les paragraphes qui suivent peuvent faire appel au mode de chiffrement EOFB pour la protection des données de clé transmises. Pour mettre en œuvre le mode EOFB pour la protection des données de clé de la même manière que pour la protection de charge utile de média, une clé de salage additionnelle (sc ou ksc) doit être utilisée.

8.6.2 Mise à jour de clé améliorée

Les points d'extrémité de version 3 ou d'une version ultérieure de la Rec. UIT-T H.235 doivent exécuter une procédure explicite/implicite de mise à jour de clé avec accusé de réception. Il s'agit d'appliquer des techniques de mise à jour de clés fiables qui sont fondées sur la méthode de mise à jour de clés sans accusé de réception telle que définie dans les versions antérieures à la version 3 de la Rec. UIT-T H.235. La capacité d'exécution d'une telle procédure doit être négociée au moyen de l'indication de capacité de version 3 conformément au § 8.2.

La Figure 5 illustre les procédures de mise à jour de clé pour un canal logique détenu par l'esclave. Si c'est l'esclave qui déclenche une mise à jour de clé et demande une nouvelle clé de session au maître, il doit envoyer une commande **MiscellaneousCommand** au maître dans laquelle l'élément **logicalChannelNumber** doit contenir le numéro de canal logique (tel que défini par l'esclave), l'élément **sharedSecret** doit être mis à "true", le fanion **direction** doit être mis à **slaveToMaster** et le nouveau numéro de charge utile dynamique doit être demandé dans l'élément **synchFlag** à l'intérieur de **EncryptionUpdateRequest**. Si c'est le maître qui déclenche une mise à jour de clé, ce message **EncryptionUpdateRequest** ne doit pas être envoyé.

Le maître – s'il répond à une demande de l'esclave ou si c'est de sa propre initiative – doit émettre une commande **EncryptionUpdateCommand** dans laquelle l'élément **logicalChannelNumber** doit contenir le numéro de canal logique, le fanion **direction** doit être mis à **slaveToMaster** à l'intérieur de **MiscellaneousCommand** et **synchFlag** dans **encryptionSync** reflète le nouveau numéro de charge utile dynamique.

L'élément **h235key** doit acheminer la nouvelle clé de session. Il doit contenir l'identité du maître dans **generalID** et le vecteur initial *IV* appliqué dans **paramS**. La clé de session de média chiffrée doit être acheminée à l'intérieur de **encryptedSessionKey**, la fonction de chiffrement s'appliquant à la clé de session maîtresse et la valeur initiale de **paramS** à la clé de session *K*. Dans le mode EOFB, une clé de salage non chiffrée est acheminée dans **ClearSalt** à l'intérieur de **paramS** (*sc*). L'élément **encryptedSaltingKey** doit acheminer la clé de salage de média chiffrée, la fonction de chiffrement s'appliquant à la clé de session maîtresse et la valeur initiale **paramSsaltIV** à la clé de salage de média *KS*. Pour le mode EOFB, une clé de salage non chiffrée (*ksc*) est acheminée dans **ClearSalt** à l'intérieur de **paramSsalt**. **clearSaltingKey** peut contenir une clé de salage de média non chiffrée auquel cas **encryptedSaltingKey** doit rester vide et inversement. Une clé de salage non chiffrée doit être transmise uniquement si la sécurité n'en souffre pas, si tel n'est pas le cas on recommande de chiffrer la clé de salage de média.

Le maître doit être prêt à recevoir un média chiffré avec la nouvelle clé de session dès la soumission de la commande **EncryptionUpdateCommand**, mais devrait continuer à utiliser l'ancienne clé de session jusqu'à réception de l'élément **EncryptionUpdateAck**. Le maître peut lancer la nouvelle clé de session après la réception de **EncryptionUpdateAck** alors que l'esclave peut appliquer la nouvelle clé de session après la réception de la commande **EncryptionUpdateCommand**.

NOTE 1 – Le maître peut choisir une valeur de type de charge utile dynamique quelconque pour l'esclave puisque le type de charge utile n'est lié qu'au port du canal de média.

NOTE 2 – Il n'est pas nécessaire que l'esclave accuse explicitement la réception de la nouvelle clé. Le maître est en mesure de déduire que l'esclave a bien reçu la clé émise, lorsqu'il reçoit le média chiffré avec le nouveau type de charge utile.

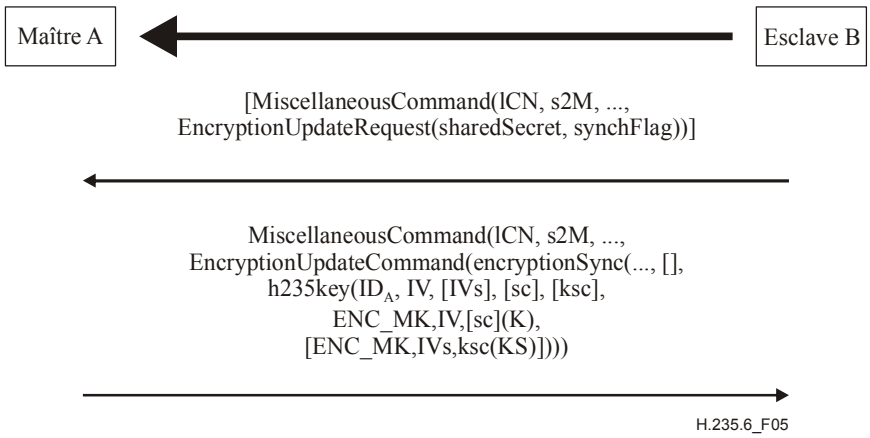


Figure 5/H.235.6 – Mise à jour de la clé de session sur le canal logique de l'esclave

La Figure 6 montre les procédures de mise à jour de clé pour un canal logique détenu par le maître. Si c'est l'esclave qui déclenche la mise à jour de clé et demande une nouvelle clé de session au maître, il doit envoyer une commande **MiscellaneousCommand** au maître dans laquelle **logicalChannelNumber** doit contenir le numéro de canal logique (tel que défini par le maître), **sharedSecret** doit être mis à "true" et le fanion **direction** doit être mis à **masterToSlave**. Si c'est le maître qui déclenche la mise à jour de la clé, ce message **EncryptionUpdateRequest** ne doit pas être envoyé.

Le maître – s'il répond à une demande de l'esclave ou si c'est de sa propre initiative – doit émettre une commande **EncryptionUpdateCommand** dans laquelle **logicalChannelNumber** doit contenir le numéro de canal logique, **direction** doit être mis à **masterToSlave**, **encryptionSync** doit contenir **synchFlag** avec le nouveau numéro de charge utile dynamique. **h235key** doit acheminer la nouvelle clé de session. **h235key** doit contenir l'identité du maître dans **generalID** et le vecteur initial *IV* appliqué dans **paramS**. La clé de session de média chiffrée doit être acheminée dans **encryptedSessionKey**, la fonction de chiffrement s'appliquant à la clé maîtresse et la valeur initiale de **paramS** à la clé de session *K*. Dans le mode EOFB, une clé de salage non chiffrée est acheminée dans **ClearSalt** à l'intérieur de **paramS** (*sc*). Dans le mode EOFB, **encryptedSaltingKey** doit acheminer la clé de salage de média non chiffrée, la fonction de chiffrement s'appliquant à la clé de session maîtresse et la valeur initiale **paramSsaltIV** à la clé de salage *KS*. Dans le mode EOFB, une clé de salage non chiffrée (*ksc*) est acheminée dans **ClearSalt** à l'intérieur de **paramSsalt**. **clearSaltingKey** peut contenir une clé de salage de média non chiffrée auquel cas **encryptedSaltingKey** doit rester vide et inversement. Une clé de salage non chiffrée doit uniquement être transmise si la sécurité n'en souffre pas, si tel n'est pas le cas il est recommandé que la clé de salage de média soit chiffrée.

L'esclave doit accuser réception de la nouvelle clé de session en répondant par **MiscellaneousCommand** dans laquelle le **logicalChannelNumber** contient le numéro de canal logique et **encryptionUpdateAck** reflète le nouveau numéro de charge utile dynamique dans **synchFlag**.

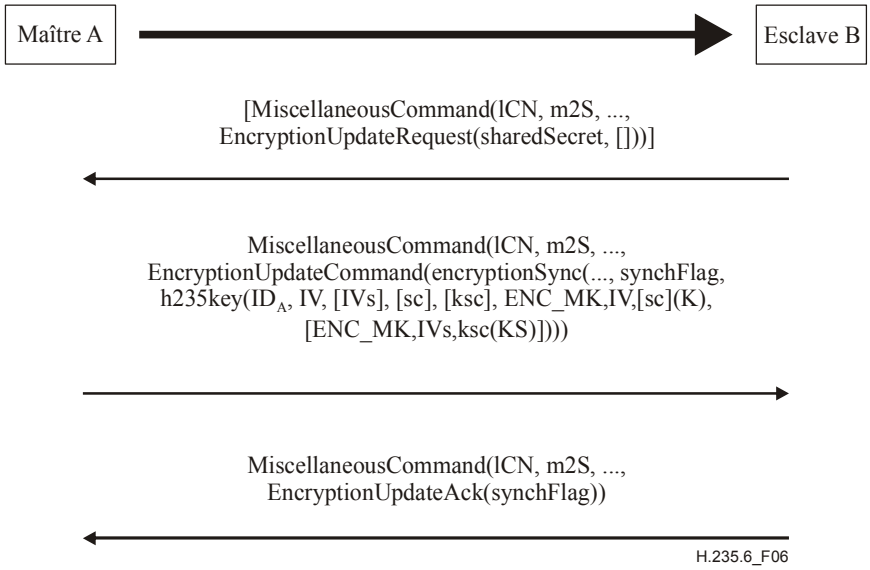


Figure 6/H.235.6 – Mise à jour de la clé de session sur le canal logique du maître

8.6.3 Mise à jour et synchronisation de clé sur la base du type de charge utile

La clé de chiffrement initiale est présentée par le maître en même temps que le numéro de charge utile dynamique dans **synchFlag** (par le biais de **EncryptionSync** de la Rec. UIT-T H.245). Le ou les récepteurs du flux de média doivent commencer à utiliser la clé dès la réception de ce numéro de charge utile dans l'en-tête RTP.

Si le canal logique négocié n'achemine qu'un seul type de charge utile, la valeur de **synchFlag** peut remplacer le type de charge utile négocié dans l'en-tête RTP. Si, en revanche, le canal logique négocié peut acheminer plusieurs types de charge utile (même si c'est dans des paquets RTP distincts), les paquets RTP doivent être formatés comme décrit dans la RFC 2198, la valeur de **synchFlag** correspondant au type de charge utile encapsulant et le ou les types de charge utile réels résidant dans le ou les blocs d'en-tête additionnels comme spécifié dans la RFC 2198.

Une ou plusieurs nouvelles clés peuvent être distribuées à tout moment par le point d'extrémité maître. La synchronisation de la toute nouvelle clé avec le flux de média doit être indiquée par le passage du type de charge utile à une nouvelle valeur dynamique.

NOTE – On notera que les valeurs spécifiques n'ont pas d'importance du moment qu'elles changent à chaque distribution d'une nouvelle clé.

8.7 Interactions non terminales

8.7.1 Passerelle

Comme indiqué au § 6.6/H.235.0, une passerelle H.323 devrait être considérée comme un élément de confiance. Cela comprend les passerelles entre protocoles (H.323-H.320, etc.) et les passerelles de sécurité (proxy/pare-feu). Le secret des communications de média peut être assuré entre le point d'extrémité communicant et la passerelle. Mais ce qui se produit au-delà de la passerelle devrait être considéré *a priori* comme non sécurisé.

8.7.2 Nouvelles clés

Les procédures décrites au § 8.5/H.323 sont appliquées par un pont de conférence afin d'éjecter un participant d'une conférence. Le maître peut produire de nouvelles clés de chiffrement pour les canaux logiques (et ne pas les distribuer au correspondant éjecté); cette méthode peut être utilisée afin d'empêcher le correspondant éjecté de surveiller les flux de média.

8.7.3 Eléments H.323 de confiance

En général, les ponts de conférence, les passerelles et les portiers (s'ils implémentent le modèle à routage par portier) sont des éléments de confiance pour ce qui est du secret des communications par le canal de commande. Si le canal d'établissement des connexions (H.225.0) est sécurisé *et* à routage par le portier, l'on doit également s'y fier. Si l'un de ces éléments H.323 doit fonctionner avec des flux de média (c'est-à-dire pour un mixage, un transcodage), ils doivent alors, par définition, être aussi considérés comme fiables pour le secret des communications de média.

On peut aussi faire confiance aux proxys ou pare-feu (bien que ne constituant pas des éléments spécifiques de la Rec. UIT-T H.323), car ils terminent des connexions et peuvent tout à fait avoir à manipuler les messages et les flux de média.

8.8 Procédures multipoint

8.8.1 Authentification

L'authentification doit être effectuée entre un point d'extrémité et de pont de conférence de la même façon qu'elle le serait dans une conférence point à point. Le pont doit déterminer la politique concernant le niveau et la sévérité de l'authentification. Comme indiqué au § 6.6/H.235.0, le pont est un élément de confiance. Les points d'extrémité d'une conférence peuvent être limités par le niveau d'authentification employé par le pont de conférence. De nouvelles commandes **ConferenceRequest/ConferenceResponse** permettent aux points d'extrémité d'obtenir du pont les certificats des autres participants à la conférence. Comme indiqué dans les procédures H.245, les points d'extrémité d'une conférence multipoint peuvent demander les certificats des autres points d'extrémité via le pont mais ne sont pas toujours en mesure d'effectuer une authentification cryptographique directe à l'intérieur du canal H.245.

8.8.2 Secret des communications

Un pont de conférence doit remporter tous les échanges maître/esclave et doit donc fournir la ou les clés de chiffrement aux participants à une conférence multipoint. Le secret des communications pour les différents émetteurs dans une session commune (dans l'hypothèse de destinations multiples) peut être obtenu avec des clés individuelles ou avec des clés communes. Ces deux modes peuvent être arbitrairement choisis par le pont. Ils ne doivent pas pouvoir être commandés à partir

d'un point d'extrémité particulier, sauf dans les modes autorisés par la politique des ponts de conférence. En d'autres termes, une clé commune peut être utilisée sur de multiples canaux logiques qui ont été ouverts par des émetteurs différents.

9 Procédures de chiffrement de flux de média

Les flux de média doivent être chiffrés au moyen de l'algorithme et de la clé qui sont présentés dans le canal H.245. Les Figures 7 et 8 montrent le flux général. On notera que l'en-tête de transport est attaché à l'unité SDU de transport une fois que cette unité a été chiffrée. Les segments opaques indiquent le secret des communications. Au fur et à mesure que de nouvelles clés sont reçues par l'émetteur et utilisées dans le chiffrement, l'en-tête d'unité SDU doit indiquer d'une façon ou d'une autre au récepteur que la nouvelle clé est désormais en usage. Par exemple, selon la Rec. UIT-T H.323, l'en-tête RTP (SDU) modifiera son type de charge utile pour indiquer le passage à la nouvelle clé.

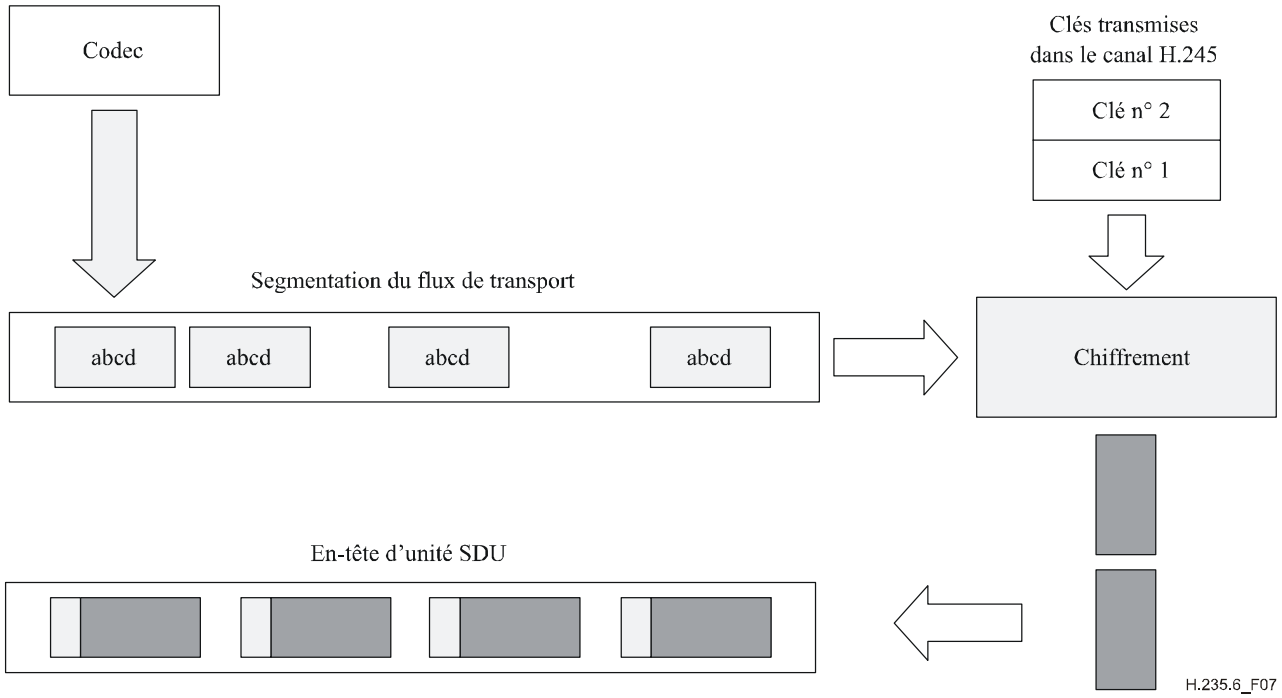


Figure 7/H.235.6 – Chiffrement du média

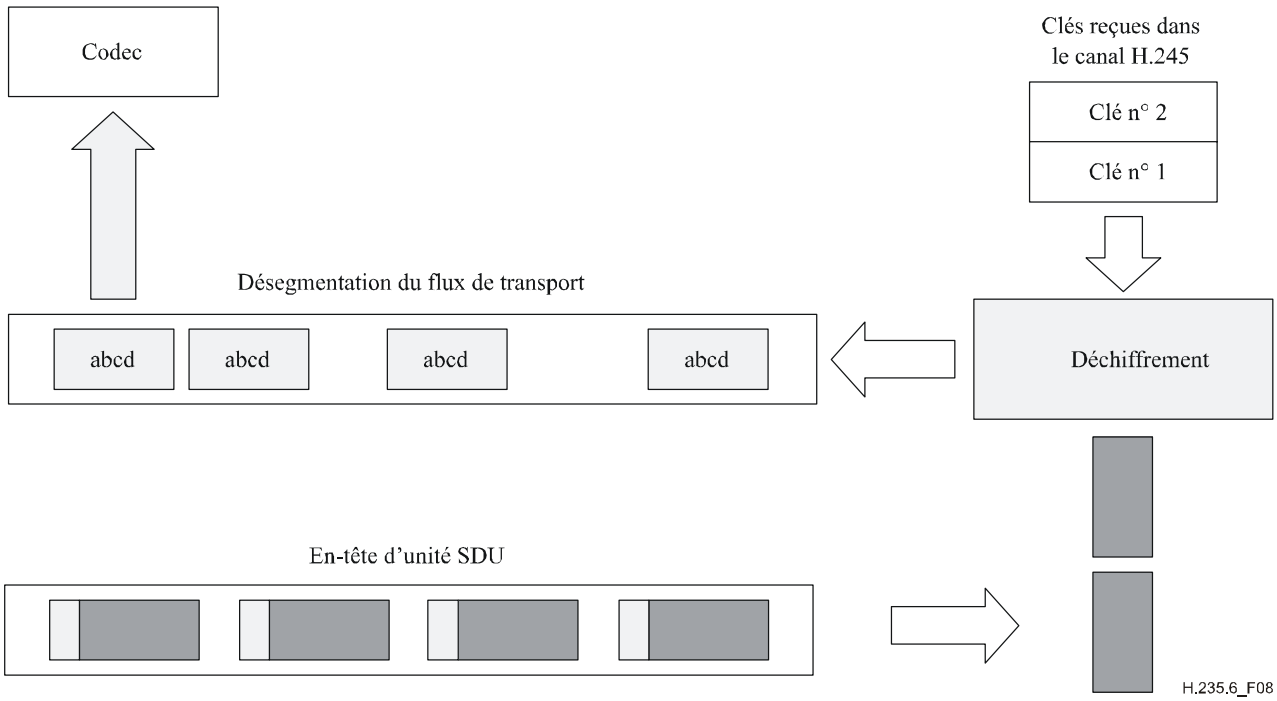


Figure 8/H.235.6 – Déchiffrement du média

9.1 Clés de session de média

Le message **encryptionUpdate** comporte le champ de clé **h235key**, qui est codé en notation ASN.1 dans le contexte de l'arbre ASN.1 de la Rec. UIT-T H.235 et qui est transmis sous forme d'une chaîne d'octets opaque par rapport au flux H.245. Cette clé peut être protégée au moyen d'un des trois mécanismes possibles en vue de sa transmission entre deux points d'extrémité.

- Si le canal H.245 est sécurisé, aucune protection additionnelle n'est appliquée aux données de clé. Celle-ci est transmise "en clair" dans ce champ; la valeur de choix ASN.1 **secureChannel** est alors utilisée.
- Si une clé secrète et un algorithme ont été établis en dehors du canal H.245 dans son ensemble (c'est-à-dire hors du flux H.323 ou sur un canal logique de type **h235Control**), le secret partagé est utilisé pour chiffrer les données de clé et la clé chiffrée résultante est insérée dans ce champ. Dans ce cas, la valeur de choix ASN.1 **sharedSecret** est utilisée.
- Des certificats peuvent être utilisés lorsque le canal H.245 n'est pas sécurisé; mais ils peuvent aussi être employés en complément d'un canal H.245 sécurisé. Lorsque des certificats sont utilisés, les données de clé sont chiffrées au moyen de la clé publique du certificat et de la structure ASN.1 **certProtectedKey**.

A tout moment au cours d'une conférence, un récepteur (ou un émetteur) peut demander une nouvelle clé (**encryptionUpdateRequest**), par exemple parce qu'il suspecte avoir perdu la synchronisation de l'un des canaux logiques. Le maître qui reçoit cette demande doit produire une ou des nouvelles clés en réponse à cette commande. Le maître peut également décider, de manière asynchrone, de distribuer une ou des nouvelles clés: il doit dans ce cas utiliser le message **encryptionUpdate**.

Après avoir reçu une demande **encryptionUpdateRequest**, un maître doit envoyer une mise à jour **encryptionUpdate**. Si la conférence est de type multipoint, le pont (en tant que maître) devrait distribuer la nouvelle clé à tous les récepteurs avant de la donner à l'émetteur. L'émetteur des données sur le canal logique doit utiliser la nouvelle clé le plus tôt possible après avoir reçu le message.

Un émetteur (supposé ne pas être le maître) peut également demander une nouvelle clé. Si l'émetteur fait partie d'une conférence multipoint, la procédure doit être la suivante:

- l'émetteur doit envoyer au pont (maître) la demande **encryptionUpdateRequest**;
- le pont devrait produire une ou des nouvelles clés et envoyer un message **encryptionUpdate** à tous les participants à la conférence, sauf à l'émetteur;
- après avoir distribué les nouvelles clés à tous les autres participants, le pont doit envoyer le message **encryptionUpdate** à l'émetteur. Celui-ci doit alors utiliser la nouvelle clé.

9.2 Mécanisme antispam pour les médias

Le destinataire d'un flux de média RTP voudra peut-être s'opposer aux attaques de type déni de service et aux attaques par inondation au niveau des ports RTP/UDP découverts. Lorsqu'il a implémenté la capacité antispam, le destinataire peut rapidement déterminer si un paquet RTP reçu provient d'une source non autorisée et le rejeter.

Lorsqu'elle est active, la capacité antispam signale que le mécanisme antispam est utilisé:

- pour des données de média en clair sans chiffrement (voir le cas 1 ci-dessous);
- en combinaison avec des données de média chiffrées lorsque **EncryptionCapability** contient un algorithme de chiffrement (voir le cas 2 ci-dessous).

Les deux possibilités offrent une authentification **RTP packet authentication** sommaire de champs sélectionnés au moyen d'un code d'identification de message calculé (MAC, *message authentication code*). Ce code peut être calculé au moyen des identificateurs d'objets définis au § 9.2.1. Le chiffrement est obtenu:

- par un algorithme de chiffrement (tel que DES en mode MAC; voir l'ISO/CEI 9797-1 et 9797-2). Un code DES-MAC est signalé au moyen de l'identificateur OID "N", alors qu'un code triple-DES-MAC est signalé au moyen de l'identificateur OID "O";
- par l'emploi d'une fonction cryptographique unidirectionnelle (telle que SHA1). L'identificateur OID qu'il convient d'utiliser est "M".

L'algorithme MAC est indiqué dans l'identificateur d'objet de l'algorithme **antiSpamAlgorithm**. L'identificateur OID de l'algorithme indique en outre, implicitement, la taille du code MAC (par exemple, 1 bloc = 64 bits pour un code DES-MAC). Pour économiser de la largeur de bande, le code MAC peut être tronqué moyennant une légère diminution de la sécurité, de manière à former un code MAC à 32 bits par exemple; cela nécessite un identificateur d'objet différent. La méthode antispam est indépendante de tout chiffrement additionnel de la charge utile (voir les cas 1 et 2 ci-dessous).

Le mécanisme antispam utilise le format de paquet RTP ci-après (voir la Figure 9) lorsque la séquence de bourrage RTP est interprétée de la manière suivante (voir le § 5/RFC 3550).

- Le bit P de l'en-tête RTP doit être mis à 1.
- Les octets de bourrage doivent être ajoutés à la fin de la charge utile avec la signification suivante:

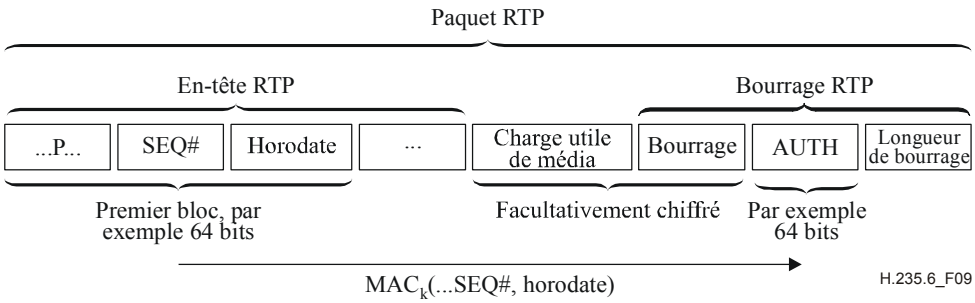


Figure 9/H.235.6 – Format de paquet RTP pour le mécanisme antispam pour les médias

NOTE 1 – Lorsque le mécanisme antispam n'est pas utilisé, les champs "AUTH" et "*longueur de bourrage*" ne sont pas utilisés eux non plus et le format de paquet RTP usuel s'applique.

1) *Cas du mécanisme antispam utilisé seul*

Ce cas s'applique lorsque les données de média ne sont pas chiffrées et que les champs de bourrage restent vides. Le dernier octet de bourrage RTP contient le décompte des octets de remplissage qu'il convient d'ignorer à la fin du paquet RTP. Les autres octets de bourrage acheminent le code MAC. Celui-ci doit être calculé sur le premier bloc chiffré de l'en-tête RTP comprenant l'horodate et le numéro de séquence variable au moyen de l'algorithme MAC négocié de **antiSpamAlgorithm** et du secret symétrique. Un secret partagé statique ou configuré manuellement ou un secret partagé négocié dynamiquement *k* peut être utilisé conformément aux procédures de la Rec. UIT-T H.235.0. Pour des tailles de bloc plus importantes (supérieures à 64 bits), il faudra prendre un nombre de bits additionnels suffisant de l'en-tête RTP, voire de la première charge utile de média.

Pour le calcul du code MAC, il est recommandé d'utiliser la clé obtenue lors de la distribution de clé de session de média H.235, bien que la clé de session appliquée ne soit pas utilisée pour le chiffrement de la charge utile. On peut utiliser, pour la gestion des clés, la connexion rapide sécurisée avec établissement de clé (voir l'Annexe J/H.323) ou le mode manuel. L'expéditeur calcule le code MAC comme indiqué ci-dessus et inclut le résultat dans le champ MAC du champ AUTH de bourrage RTP. L'expéditeur et le destinataire connaissent la taille du champ AUTH et la longueur du code MAC par **antiSpamAlgorithm**.

La vérification du code MAC du côté destinataire devrait être faite le plus tôt possible, éventuellement dans la pile RTP ou, au plus tard, avant le déchiffrement ou la décompression de la charge utile. Le destinataire recalcule d'abord le code MAC de la même manière que l'a fait l'expéditeur et compare le code MAC calculé avec le code MAC remis dans le bourrage RTP. Si les codes MAC ne concordent pas, l'en-tête RTP a été modifié pendant le transit ou a été envoyé par une entité non autorisée qui ne possède pas la clé. Donc, le paquet RTP ne pouvant être authentifié doit être éliminé et l'événement peut être journalisé; cela indique probablement une tentative d'attaque de type déni de service. Dans les autres cas, le traitement du paquet RTP authentifié peut se poursuivre, le bourrage RTP est supprimé et la charge utile est envoyée dans le codec.

NOTE 2 – Le calcul/vérification sommaire du code MAC avec chiffrement DES fait intervenir une seule opération de chiffrement; quant au code MAC avec hachage SHA1, il est calculé sur une partie réduite des paquets de longueur fixe; les opérations de chiffrement consomment donc un minimum de ressources de traitement.

2) *Cas de la méthode antispam avec chiffrement de la charge utile*

Ce cas s'applique lorsque les données de média sont chiffrées et que la méthode antispam est invoquée. Lorsque la charge utile ne correspond pas à des limites de blocs paires, certains octets de bourrage additionnels doivent être ajoutés à la charge utile, devant le code MAC. La charge utile de média est chiffrée conformément au § 9.

EncryptionCapability définit l'algorithme de chiffrement de la charge utile alors que **antiSpamAlgorithm** définit la méthode antispam. Pour des raisons de sécurité, le chiffrement de média et le code MAC doivent utiliser des clés de session différentes. La clé *k* du code MAC est calculée en introduisant la clé de chiffrement *K* dans la fonction de hachage unidirectionnelle SHA1.

k = SHA1(*K*); il convient de prendre suffisamment de bits sur le résultat du hachage, selon l'ordre des octets dans le réseau. Lorsque **antiSpamAlgorithm** indique un algorithme de chiffrement, les bits collectés doivent être transformés en clé de chiffrement correcte; par exemple, en remplissant les bits de parité DES.

Lorsque le destinataire a correctement vérifié l'authenticité du paquet RTP, la charge utile est déchiffrée et le bourrage RTP est ignoré. La procédure générale est conforme au cas 1 ci-dessus.

9.2.1 Liste des identificateurs d'objet

Le Tableau 5 énumère tous les identificateurs d'objet OID mentionnés.

Tableau 5/H.235.6 – Identificateurs d'objet utilisés pour le mécanisme antispam

Désignation de l'identificateur d'objet	Valeur de l'identificateur d'objet	Description
"M"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 8}	Mécanisme antispam utilisant l'algorithme HMAC-SHA1-96.
"N"	{iso(1) identified-organization(3) oiw(14), secsig(3) algorithm(2) desMAC(10)}	Mécanisme antispam utilisant un code MAC à 64 bits (voir l'ISO/CEI 9797-1 et 9797-2) avec l'algorithme DES (56 bits).
"O"	{iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) desEDE(17)}	Mécanisme antispam utilisant un code MAC (voir l'ISO/CEI 9797-1 et 9797-2) avec l'algorithme triple-DES (168 bits).

9.3 Considérations liées aux protocoles RTP/RTCP

L'utilisation du chiffrement dans un flux RTP suivra la méthode générale qui a été recommandée dans le document [RTP]. Le média doit être chiffré de manière indépendante, paquet par paquet.

NOTE – Il convient de noter que, si la longueur d'un paquet RTP est supérieure à celle d'une unité MTU, une perte partielle (de fragment) provoquera l'indéchiffrabilité de l'ensemble du paquet RTP.

L'en-tête RTP ne doit pas être chiffré. Pour les codecs audio/vidéo, la totalité de la charge utile du codec audio/vidéo, y compris les éventuels en-têtes de charge utile audio/vidéo doivent être chiffrés. La synchronisation de nouvelles clés et du texte chiffré est fondée sur le type de charge utile dynamique (voir le § 8.6.3).

On part du principe que le chiffrement n'est appliqué qu'à la charge utile de chaque paquet RTP; les en-têtes RTP restent en clair. On suppose que tous les paquets RTP contiennent un nombre entier d'octets. La façon dont les paquets RTP sont encapsulés dans la couche Transport ou Réseau est hors du domaine d'application de la présente Recommandation. Tous les modes doivent prévoir la perte (ou le déclassement) de paquets, ainsi que le bourrage de paquets pour qu'ils comportent un nombre d'octets approprié.

Le déchiffrement du flux doit être effectué sans tenir compte des états afin de tenir compte du fait que des paquets peuvent être perdus; chaque paquet devrait être déchiffré isolément. Deux exigences du mode algorithmique par blocs doivent s'appliquer comme suit:

9.3.1 Vecteurs d'initialisation

La plupart des modes par blocs impliquent un certain "chaînage"; chaque cycle de chiffrement dépend d'une certaine manière de la sortie du cycle précédent. Au début d'un paquet, une certaine valeur initiale de bloc (généralement appelée vecteur d'initialisation (IV, *initialization vector*)) doit donc être fournie afin de commencer le processus de chiffrement. Quel que soit le nombre d'octets de flux qui sont traités à chaque cycle de chiffrement, la longueur du vecteur d'initialisation est toujours égale à celle d'un bloc. Tous les modes, sauf le mode dictionnaire (ECB, *electronic code book*), nécessitent un vecteur d'initialisation.

9.3.1.1 Vecteurs d'initialisation CBC

Un vecteur d'initialisation (IV) est nécessaire lorsqu'on utilise un chiffrement par blocs dans le mode CBC pour chiffrer les charges utiles des paquets RTP. La taille d'un vecteur IV est la même que la taille du bloc pour un chiffrement par blocs particulier. Par exemple, la taille du vecteur d'initialisation pour les algorithmes DES et 3-DES est de 64 bits, alors que pour l'algorithme AES, elle est de 128 bits.

Dans le mode CBC, un vecteur d'initialisation doit être construit à partir des *B* premiers octets (où *B* est la longueur de bloc) de Seq# concaténé avec l'horodate. Cela forme la séquence, *SSTTTT*, dans laquelle *SS* est Seq# RTP sur deux octets et *TTTT* est l'horodate RTP à quatre octets. Cette séquence doit être répétée jusqu'à ce que *B* octets aient été produits, avec troncage si nécessaire. Par exemple, les vecteurs d'initialisation de 64 et 128 bits contiendraient les séquences *SSTTTTSS* et *SSTTTTSSTTTTSSTT*, respectivement. Il convient de noter que le vecteur d'initialisation produit de cette façon peut faire apparaître une séquence de clé qui est considérée comme "faible" pour un algorithme particulier.

9.3.1.2 Vecteurs d'initialisation EOFB

Le vecteur initial unique IV pour chaque paquet RTP dans le mode EOFB doit être calculé comme suit:

on associe à chaque paquet RTP un indice implicite *i* de paquet à 48 bits tel que défini dans [SRTP] où $i = 2^{16} \times \text{ROC} + \text{SEQ}$, SEQ étant le numéro de séquence extrait de l'en-tête RTP et ROC le compteur de cycles complets de 32 bits qui compte le nombre de fois où le numéro de séquence (SEQ) passe par 65535.

Initialement, le compteur ROC doit être mis à zéro. Chaque fois que le numéro de séquence termine un cycle modulo 2^{16}, l'expéditeur doit incrémenter le compteur ROC de un modulo 2^{32}.

Le vecteur initial IV est calculé comme ($i \parallel T$ [$\parallel i \parallel T \parallel \ldots$]), où l'indice *i* à 48 bits et l'horodate *T* de 32 bits provenant de l'en-tête RTP sont concaténés plusieurs fois jusqu'à ce que la taille de bloc soit remplie. Le symbole || représente la concaténation.

NOTE – Le compteur ROC et le vecteur IV sont conservés et calculés localement de chaque côté homologue et ne sont pas transmis.

Lorsque des paquets sont perdus ou réordonnés, le récepteur doit calculer un indice *i* estimé:

$i = 2^{16} \times v + \text{SEQ}$, où *v* est pris dans l'ensemble {ROC-1, ROC, ROC+1} modulo 2^{32} de sorte que *v* soit très proche (dans le sens 2^{48}) de la valeur $2^{16} \times \text{ROC} + s_l$, s_l étant le numéro de séquence conservé au niveau du récepteur. Après avoir traité le paquet en utilisant l'indice estimé, le récepteur doit décider si s_l et ROC doivent être mis à jour. Par exemple, une méthode simple (mais qui n'est pas insensible aux erreurs) consiste simplement à fixer s_l à SEQ (si SEQ > s_l) et si la valeur $v = \text{ROC} + 1$ a été utilisée, à actualiser ROC à *v*; voir également pour plus d'informations [SRTP] section 3.2.1.

9.3.2 Bourrage

Les modes ECB et CBC traitent toujours le flux d'entrée bloc par bloc. Les modes CFB et OFB peuvent traiter un nombre N ($\leq B$) quelconque d'octets du flux d'entrée, il est recommandé que $N = B$.

Deux méthodes permettent de traiter les paquets dont la charge utile ne correspond pas à un nombre entier de blocs:

1) l'extraction cryptographique en cas de blocs incomplets pour les modes ECB et CBC; pas de bourrage pour les modes CFB et EOFB;

2) le bourrage de la façon prescrite dans [RTP], section 5.1.

La référence [RTP], section 5.1 décrit une méthode de bourrage dans laquelle la charge utile est bourrée jusqu'à former un nombre entier de blocs. Le dernier octet doit indiquer le nombre d'octets de bourrage (y compris ce dernier octet) et le bit *P* doit être mis à 1 dans l'en-tête RTP. La valeur du bourrage devrait être déterminée par la convention normale de l'algorithme de chiffrement.

Toutes les implémentations conformes à la Rec. UIT-T H.235 doivent prendre en charge les deux méthodes. La méthode utilisée peut être déduite comme suit: si le bit *P* est mis à 1 dans l'en-tête RTP, le paquet est bourré; si la longueur du paquet n'est pas un multiple de *B* et que le bit *P* n'est pas mis à 1, la méthode d'extraction cryptographique s'applique. Sinon, la longueur du paquet est un multiple de *B* et le bourrage ne s'applique pas.

9.3.3 Protection RTCP

L'application de techniques de chiffrement aux éléments RTCP appelle un complément d'étude.

9.3.4 Flux de charge utile sécurisée

Les réseaux H.323, lorsqu'ils sont utilisés par exemple pour la transmission modem sur IP, mettent en œuvre la signalisation H.245 pour établir et négocier le canal de données en bande vocale et le protocole RTP pour la paquétisation d'un flux de charge utile multiple (MPS, *multiple payload stream*).

Pour un seul flux de média avec un seul type de charge utile ou FEC pour un autre canal, le type de charge utile dynamique dans **encryptionSync** doit remplacer le type de charge utile par défaut.

Pour les flux encapsulants (c'est-à-dire assurant le codage avec redondance ou avec FEC codé RFC 2198), le type de charge utile dynamique dans **encryptionSync** doit remplacer le type de charge utile encapsulant.

Pour les flux de charge utile multiple, le type de charge utile dynamique dans **syncFlag** de **encryptionSync** doit être ignoré et on doit utiliser à sa place les types de charge utile (optionnels) dans le ou les éléments **multiplePayloadStreamElement**.

La commande **EncryptionUpdateCommand** doit être utilisée pour la procédure de mise à jour de clé améliorée pour distribuer les données de la nouvelle clé de session (voir le § 8.6.2). **multiplePayloadStream** est uniquement utilisé lorsqu'un flux de charge utile multiple doit se voir attribuer une nouvelle clé, auquel cas le type de charge utile dynamique dans **EncryptionSync** doit être ignoré.

9.3.5 Interfonctionnement avec la Rec. UIT-T J.170

A étudier.

9.4 Algorithme 3-DES en mode CBC externe

Il *convient* d'utiliser, dans le présent profil de sécurité, l'algorithme 3-DES à 168 bits en mode CBC externe, présenté sur la Figure 10. Sur cette figure, chaque k_i désigne une clé de 56 bits. Il *faut* utiliser une clé de 56 bits différente dans chaque bloc de chiffrement (E) et de déchiffrement (D).

Aucune des 64 clés faibles de l'algorithme DES n'est réputée entraîner de faiblesse dans l'algorithme 3-DES. Néanmoins, les implémentations qui sont conformes à ce profil devraient rejeter la clé lorsqu'une clé DES faible intervient (voir RFC 2405).

De plus amples informations sur l'algorithme 3-DES sont données dans les documents [Schneier] et [RFC2405].

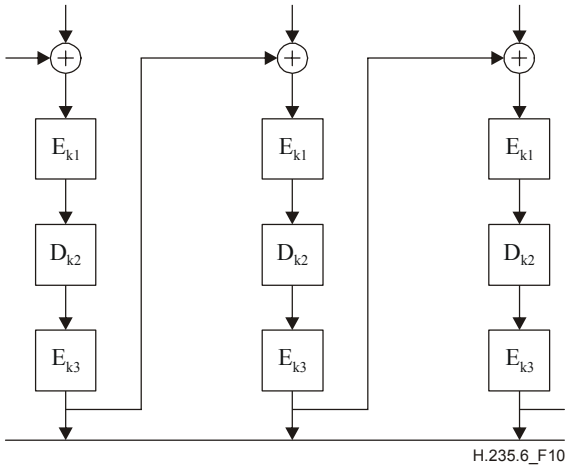


Figure 10/H.235.6 – Chiffrement 3-DES en mode CBC externe

9.5 Algorithme DES en mode EOFB

La voix peut être chiffrée au moyen de l'algorithme DES fonctionnant dans le mode EOFB. Le mode EOFB permet d'utiliser le parallélisme dans les implémentations. Lorsqu'on utilise le mode EOFB, il est recommandé, pour des raisons de performance et de sécurité, d'appliquer un bouclage sur la totalité du bloc de chiffrement (c'est-à-dire sur les 64 bits pour l'algorithme DES dans le cas par exemple où $n = j = 64$). Toutefois, comme il n'assure pas le chaînage sur les blocs et les bits, l'EOFB peut être sensible à des attaques spécifiques dépendant des propriétés statistiques des données d'entrée en clair. Ainsi, les clés (voir § 8.6) devraient être mises à jour régulièrement et au moins avant le début d'un nouveau cycle de la valeur initiale. Voir le § 9.3.1.2 pour le calcul de la valeur initiale.

9.6 Algorithme 3-DES en mode EOFB externe

L'algorithme 3-DES à 168 bits en mode EOFB externe, illustré à la Figure 11, peut être utilisé dans le présent profil de sécurité. Sur cette figure, chaque k_i désigne une clé de 56 bits. Une clé différente de 56 bits *doit* être utilisée dans chaque bloc de chiffrement (E) et de déchiffrement (D). Aucune des 64 clés faibles de l'algorithme DES n'est réputée occasionner de faiblesse dans l'algorithme 3-DES. Toutefois, les implémentations conformes à ce profil devraient rejeter la clé lorsqu'une clé DES faible intervient [RFC2405].

Pour de plus amples informations sur l'algorithme 3-DES, voir [Schneier], [RFC2405].

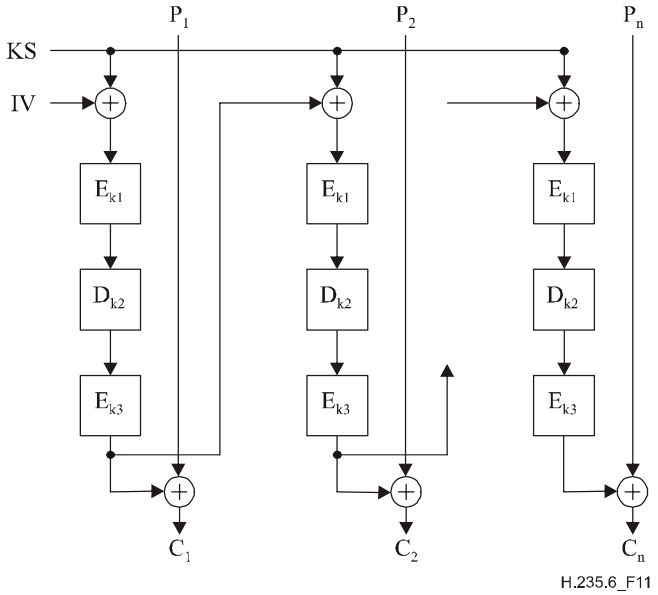


Figure 11/H.235.6 – Chiffrement 3-DES en mode EOFB externe

10 Interception licite

A étudier (voir [LI]).

11 Liste des identificateurs d'objet

Le Tableau 6 énumère tous les identificateurs OID mentionnés (voir également [OIW] et [WEBOIDs]). Il y a des identificateurs d'objet pour H.235v1 (Rec. UIT-T H.235v1) et pour H.235v2 (Rec. UIT-T H.235 v2).

Tableau 6/H.235.6 – Identificateurs d'objet

Désignation de l'identificateur d'objet	Valeur(s) de l'identificateur d'objet	Description
"DHdummy"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 40} {itu-t (0) recommendation (0) h (8) 235 version (0) 3 40}	Groupe DH non standard fourni explicitement
"DH1024"	{itu-t (0) recommendation (0) h (8) 235 version (0) 2 43} {itu-t (0) recommendation (0) h (8) 235 version (0) 3 43}	Groupe DH à 1024 bits
"DH1536"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 44}	Groupe DH à 1536 bits
"X"	{iso(1) member-body(2) us(840) rsadsi(113549) encryptionalgorithm(3) 2}	Chiffrement vocal utilisant l'algorithme compatible-RC2 (56 bits) ou compatible-RC2 en mode CBC et le groupe DH à 512 bits
"X1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 27}	Chiffrement vocal utilisant l'algorithme compatible-RC2 (56 bits) ou compatible-RC2 en mode EOFB et le groupe DH à 512 bits

Tableau 6/H.235.6 – Identificateurs d'objet

Désignation de l'identificateur d'objet	Valeur(s) de l'identificateur d'objet	Description
"Y"	{iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) descbc(7)}	Chiffrement vocal utilisant l'algorithme DES (56 bits) en mode CBC et le groupe DH à 512 bits
"Y1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 28}	Chiffrement vocal utilisant l'algorithme DES (56 bits) en mode EOFB et le groupe DH à 512 bits avec bouclage à 64 bits
"Z1"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 29}	Chiffrement vocal utilisant l'algorithme 3-DES (168 bits) en mode EOFB externe et le groupe DH à 1024 bits avec bouclage à 64 bits
"Z2"	{itu-t (0) recommendation (0) h (8) 235 version (0) 3 30}	Chiffrement vocal utilisant l'algorithme AES (128 bits) en mode EOFB et le groupe DH à 1024 bits
"Z3"	{joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) 3 nistAlgorithm(4) aes(1) cbc(2)}	Chiffrement vocal utilisant l'algorithme AES (128 bits) en mode CBC et le groupe DH à 1024 bits
"Z"	{iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) desEDE(17)}	Chiffrement vocal utilisant l'algorithme 3-DES (168 bits) en mode CBC externe et le groupe DH à 1024 bits

Appendice I

Détails d'implémentation H.323

I.1 Méthodes de bourrage cryptographique

L'extraction cryptographique est décrite dans [Schneier], p. 191 et 196. Les Figures I.1 à I.5 illustrent cette technique.

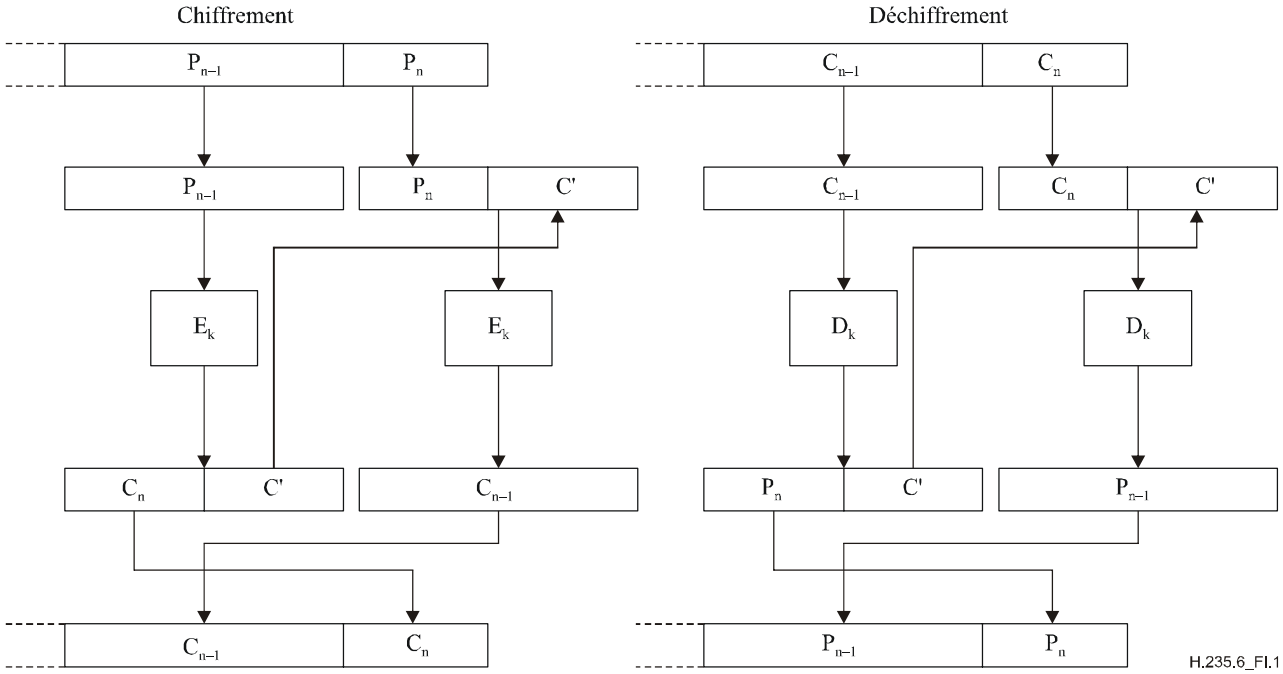


Figure I.1/H.235.6 – Extraction d'un texte chiffré en mode ECB

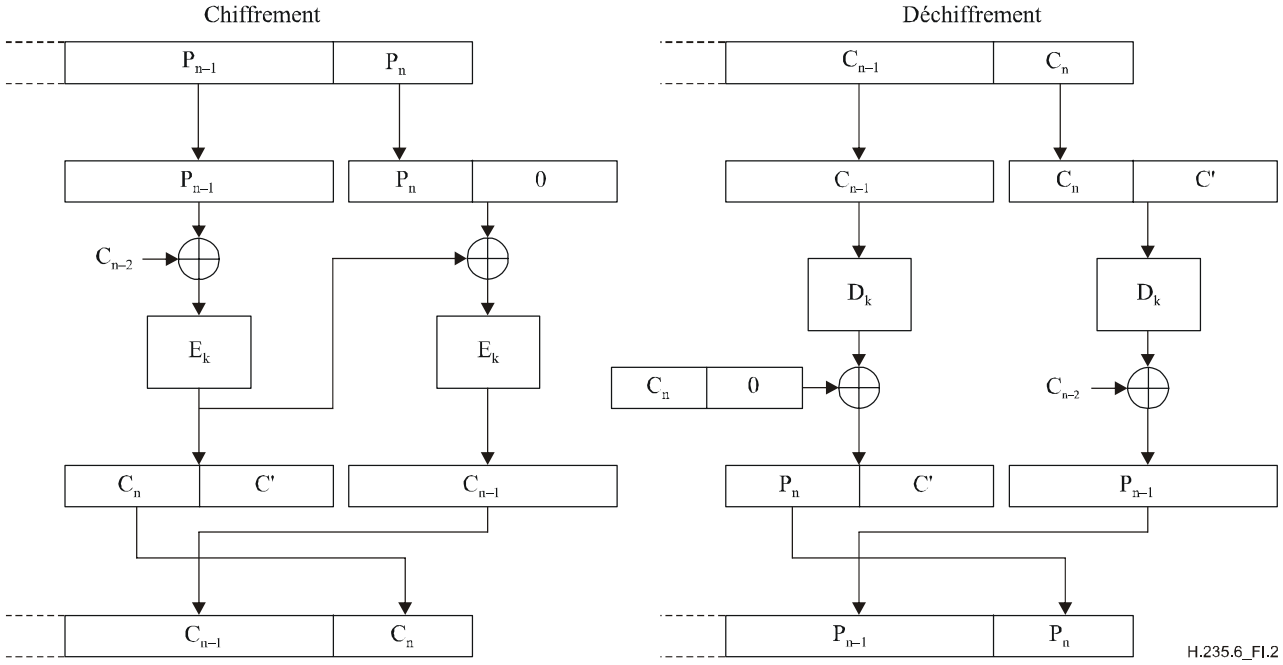


Figure I.2/H.235.6 – Extraction d'un texte chiffré en mode CBC

NOTE – Pour pouvoir effectuer une extraction de texte chiffré dans les modes ECB ou CBC, il faut que la charge utile achemine au moins un bloc complet. Les implémentations qui mettent en œuvre l'extraction de textes chiffrés dans les modes ECB ou CBC devraient s'assurer que la charge utile achemine toujours au moins un bloc chiffré, par exemple par un choix approprié du rythme d'échantillonnage ou de paquétisation ou le choix approprié de l'algorithme de chiffrement.

Au cas où la charge utile occupe moins d'un bloc, le vecteur initial IV doit être utilisé comme bloc de texte chiffré précédent lorsque l'extraction de texte chiffré est appliquée dans le mode CBC.

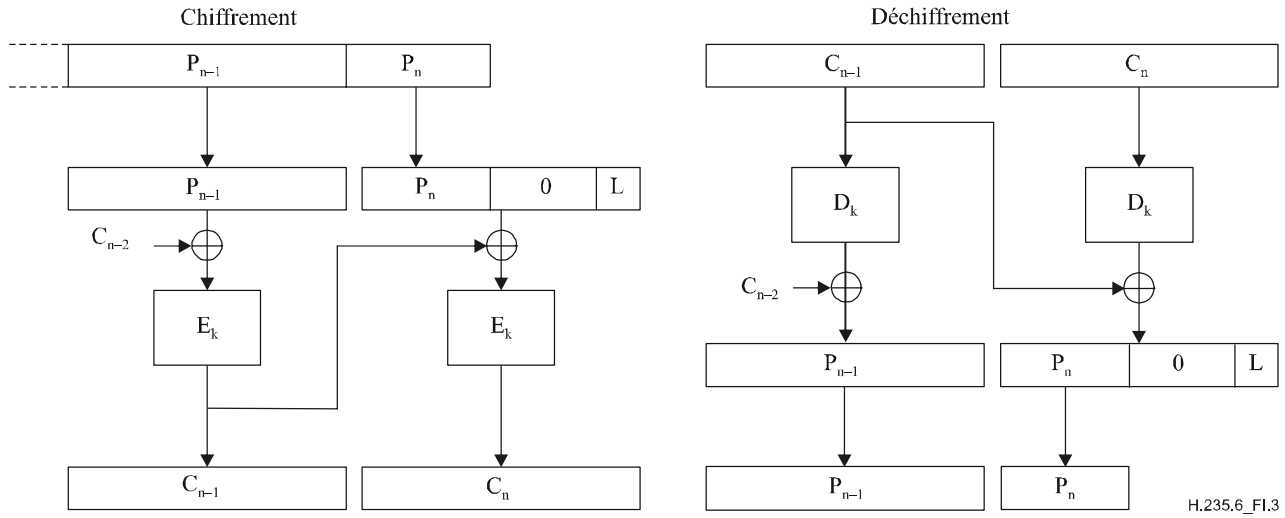


Figure I.3/H.235.6 – Bourrage de zéros en mode CBC

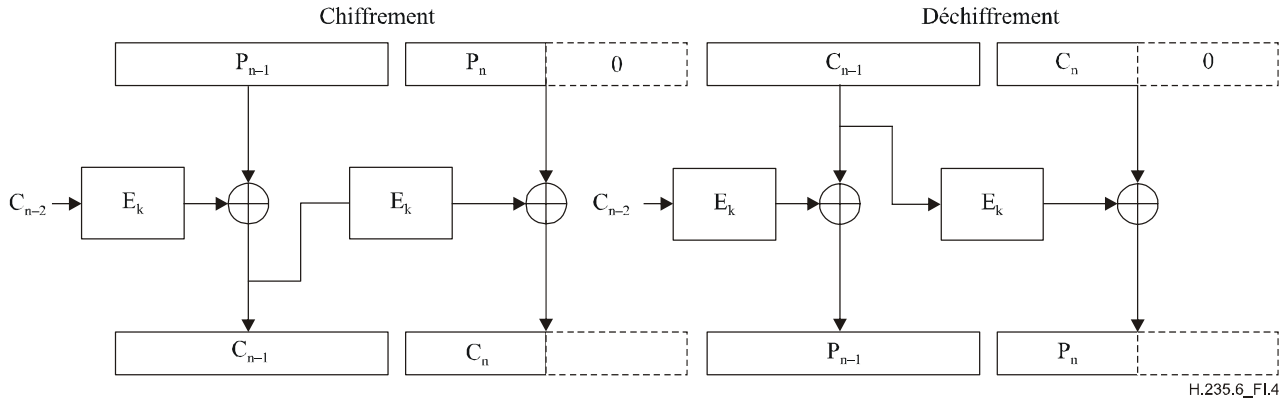
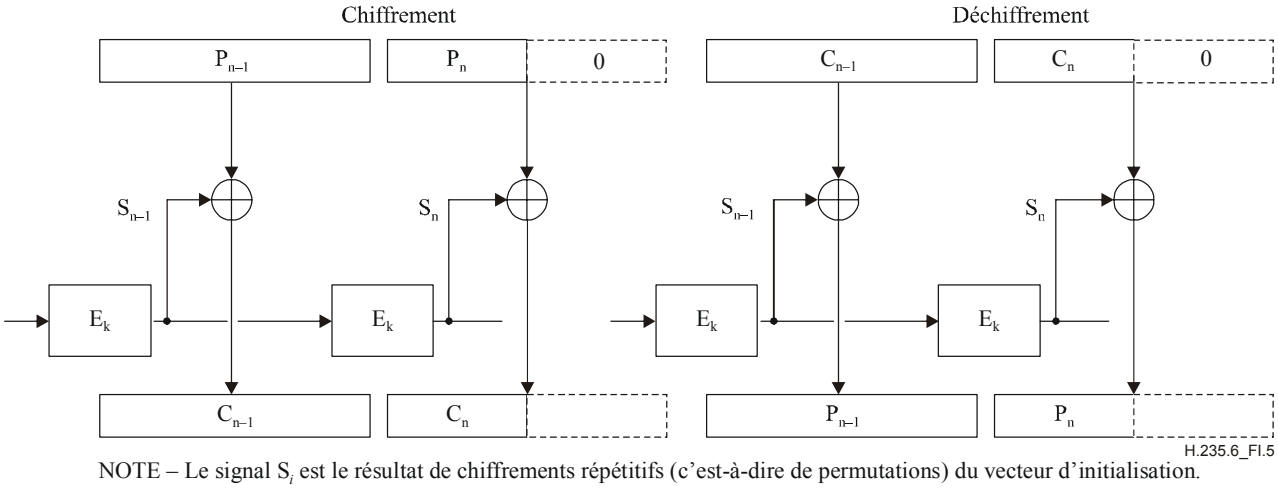


Figure I.4/H.235.6 – Bourrage de zéros en mode CFB



NOTE – Le signal S_i est le résultat de chiffrements répétitifs (c'est-à-dire de permutations) du vecteur d'initialisation.

Figure I.5/H.235.6 – Bourrage de zéros en mode OFB

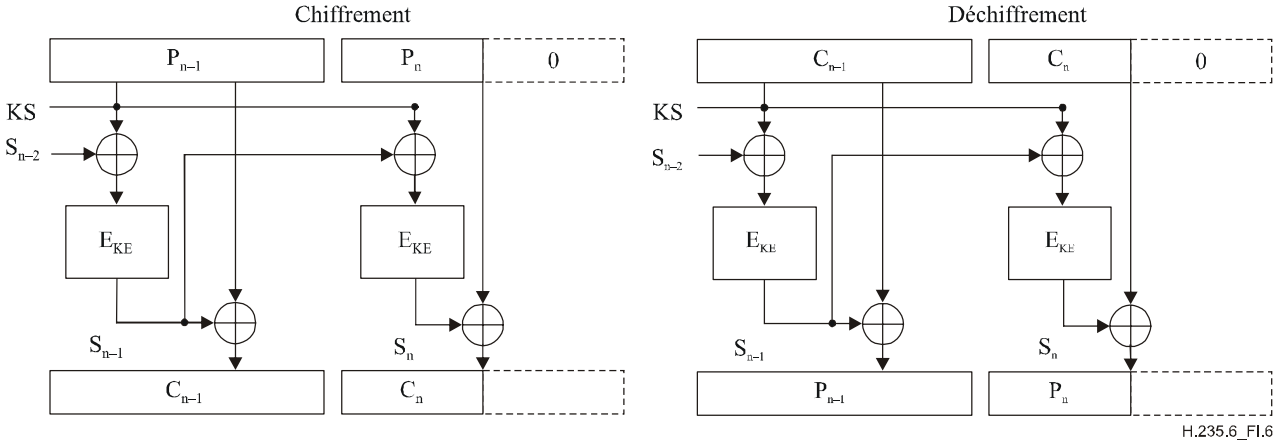


Figure I.6/H.235.6 – Mode EOFB avec bourrage de zéros

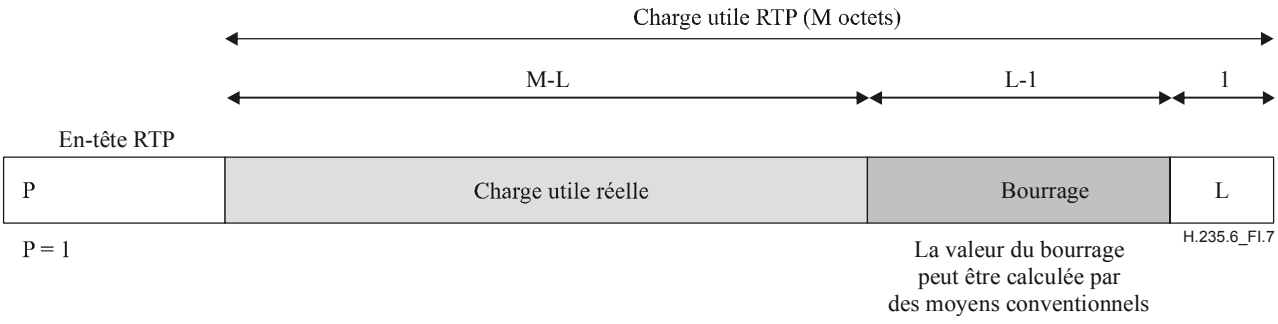


Figure I.7/H.235.6 – Bourrage tel que prescrit par le protocole RTP

I.2 Nouvelles clés

Les procédures décrites au § 8.5/H.323 sont appliquées par un pont de conférence afin d'éjecter un participant d'une conférence. Le maître peut produire de nouvelles clés de chiffrement pour les canaux logiques (et ne pas les distribuer au correspondant éjecté); cette méthode peut être utilisée afin d'empêcher le correspondant éjecté de surveiller les flux de média.

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
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	Gestion des télécommunications y compris le RGT et maintenance des réseaux
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, communication entre systèmes ouverts et sécurité
Série Y	Infrastructure mondiale de l'information, protocole Internet et réseaux de prochaine génération
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication

Imprimé en Suisse
Genève, 2006