Union internationale des télécommunications

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

H.235.8

(09/2005)

SÉRIE H: SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

Infrastructure des services audiovisuels – Aspects système

Cadre de sécurité H.323: échange de clés dans le protocole SRTP au moyen de canaux de signalisation sécurisés

Recommandation UIT-T H.235.8

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.235.8

Cadre de sécurité H.323: échange de clés dans le protocole SRTP au moyen de canaux de signalisation sécurisés

Résumé

La présente Recommandation a pour objet de décrire les procédures de sécurité applicables à l'échange de clés dans le protocole SRTP au moyen de canaux de signalisation sécurisés dans des réseaux H.323/H.235.

La présente Recommandation est fondée sur les Recommandations UIT-T H.323 et H.225.0 (version 4 ou version ultérieure).

Source

La Recommandation UIT-T H.235.8 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.

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.235.8

Cadre de sécurité H.323: échange de clés dans le protocole SRTP au moyen de canaux de signalisation sécurisés

1 Domaine d'application

La présente Recommandation a pour objet de définir des orientations concernant les procédures de sécurité permettant de prendre en charge le protocole de transport sécurisé en temps réel (SRTP, *secure real time protocol*) entre deux points d'extrémité H.323 dans les cas où les données cryptographiques associées au canal de média sont acheminées dans un canal de signalisation sécurisé (par exemple, IPsec (RFC 2401), TLS (RFC 2246) ou un autre mécanisme H.235). Ces procédures de sécurité sont proposées en remplacement d'autres procédures de sécurité H.235 prenant en charge le protocole SRTP.

La présente Recommandation décrit les procédures visant à prendre en charge le protocole de transport sécurisé en temps réel (SRTP) de l'IETF dans les systèmes H.323. Le protocole SRTP assure des services de sécurité pour les médias RTP et est tributaire de protocoles distincts pour assurer des services de gestion de clés ainsi que la négociation des paramètres cryptographiques. Ces procédures ne devraient pas être utilisées lorsque le canal de signalisation sécurisé aboutit à un système intermédiaire, auquel cas les données cryptographiques SRTP devraient être acheminées par un mécanisme sécurisé de bout en bout.

Ces procédures prennent en charge la signalisation, la négociation et le transport des clés cryptographiques SRTP, des identificateurs d'algorithmes, d'authentification et de chiffrement ainsi que d'autres paramètres de session entre deux points d'extrémité H.323.

Un aspect fondamental de ces procédures réside dans le fait que l'esclave H.245 aussi bien que le maître H.245 doivent être capables de générer et de distribuer des clés cryptographiques.

Il est possible d'échanger des capacités de sécurité SRTP par l'échange de capacités entre deux terminaux au moyen d'entrées h235SecurityCapability du tableau capabilityTable du message TerminalCapabilitySet H.245. Le champ genericH235SecurityCapability contenu dans le champ encryptionAuthenticationAndIntegrity de l'entrée h235SecurityCapability contient le champ SrtpCryptoCapability qui spécifiera les suites cryptographiques SRTP.

Un paramètre "crypto" SRTP est spécifié pour signaler et négocier les paramètres cryptographiques SRTP. La définition du paramètre "crypto" dans la présente Recommandation se limite aux flux de média unidiffusés entre deux entités, chaque source possédant une clé cryptographique unique; la prise en charge des flux de média multidiffusés ou des flux multipoint unidiffusés appelle un complément d'étude.

Le paramètre "crypto" SRTP est destiné à établir les paramètres cryptographiques SRTP lors de l'échange d'un seul message ou lors de l'échange d'un message dans chaque sens. Dans le cas de l'échange d'un message dans chaque sens, les paramètres cryptographiques peuvent être négociés. Par exemple, dans la procédure de connexion rapide, le point d'extrémité H.323 offrant envoie un ensemble de paramètres "crypto" SRTP offerts au point d'extrémité H.323 répondant, chaque offre étant encapsulée dans un message OpenLogicalChannel H.245 distinct. Le point d'extrémité H.323 répondant peut ensuite accepter un des paramètres offerts et répondre avec une réponse qui comprend le sous-ensemble de paramètres sélectionnés encapsulé dans un message OpenLogicalChannel H.245.

Dans le cas de l'échange d'un seul message, aucune négociation n'est prévue. Le point d'extrémité H.323 offrant envoie les paramètres "crypto" SRTP au point d'extrémité H.323 répondant, lequel soit accepte les paramètres offerts, soit refuse l'appel.

Des procédures cryptographiques à clé publique peuvent être appliquées en complément afin d'assurer la confidentialité et l'authentification de bout en bout des données de clé de session SRTP échangées entre deux points d'extrémité H.323 en chiffrant puis en signant les données de clé SRTP dans le cas où le protocole de sécurité d'encapsulation (par exemple, IPsec, TLS) n'est mis en œuvre que jusqu'à un dispositif intermédiaire et, par conséquent, n'assure pas la sécurité de bout en bout.

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.235.0 (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.323 (2003), *Systèmes de communication multimédia en mode paquet*.

– Recommandation UIT-T H.460.11 (2004), *Etablissement d'appel différé dans les systèmes H.323*.

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

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

– IETF RFC 2733 (1999), *An RTP Payload Format for Generic Forward Error Correction*.

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

– IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications*.

– IETF RFC 3711 (2004), *The Secure Real-time Transport Protocol (SRTP)*.

– IETF RFC 3852 (2004), *Cryptographic Message Syntax (CMS)*.

2.2 Références informatives

– IETF Draft, F. Andreasen, M. Baugher, D. Wing: *Session Description Protocol Security Descriptions for Media Streams,* <draft-ietf-mmusic-sdescriptions-11.txt>.

3 Symboles et abréviations

La présente Recommandation utilise les abréviations suivantes:

AES norme de chiffrement perfectionnée (*advanced encryption algorithm*)

ASN.1 notation de syntaxe abstraite numéro un (*abstract syntax notation one*)

CA autorité de certification (*certificate authority*)

CEK clé de chiffrement de contenu (*content encryption key*)

CMS syntaxe de message cryptographique (*cryptographic message syntax*)

EP point d'extrémité (*endpoint*)

FEC correction d'erreur directe (*forward error correction*)

FFS à étudier (*for further study*)

F8 algorithme de chiffrement UMTS (*UMTS encryption algorithm*)

GK portier (*gatekeeper*)

GW passerelle (*gateway*)

HMAC code d'authentification de message par hachage avec clé (*keyed-hash message authentication code*)

IETF Groupe de travail d'ingénierie Internet (*Internet engineering task force*)

KDR fréquence de calcul de clé (*key derivation rate*)

MAC code d'authentification de message (*message authentication code*)

MKI identificateur de clé maîtresse (*master key identifier*)

OID identificateur d'objet (*object identifier*)

OLC ouverture de canal logique (*open logical channel*)

PKI infrastructure de clé publique (*public key infrastructure*)

RAS enregistrement, admission et statut (*registration, admission, status*)

ROC compteur de cycles complets (*roll-over counter*)

RTCP protocole de commande de transport en temps réel (*real-time transport control protocol*)

RTP protocole de transport en temps réel (*real-time transport protocol*)

SHA1 algorithme de hachage sécurisé n^{o} 1 (*secure hash algorithm 1*)

SRTCP protocole de commande de transport sécurisé en temps réel (*secure real-time transport control protocol*)

SRTP protocole de transport en temps réel sécurisé (*secure real-time transport protocol*)

SSRC source de synchronisation (*synchronization source*)

TLS sécurité de la couche de transport (*transport level security*)

WSH indication de taille de fenêtre (*window size hint*)

4 Description des paramètres

L'échange des capacités cryptographiques et des données de clé dans le protocole SRTP s'effectue au moyen de deux paramètres:

– **SrtpCryptoInfo** dans **StrpCryptoCapability** doit contenir la suite cryptographique ainsi que les paramètres de session. Le paramètre **SrtpCryptoInfo** doit être transporté dans le paramètre **genericH235SecurityCapability** H.245 pour signaler et négocier les paramètres cryptographiques SRTP.

– **SrtpKeyParameters** dans **SrtpKeys** doit contenir les données de clé SRTP. Le conteneur **SrtpKeys** dans le paramètre **h235Key** H.245 doit transporter un ou plusieurs paramètres **SrtpKeyParameters** ainsi que les clés SRTP.

L'emploi des paramètres cryptographiques SRTP dans la présente Recommandation se limite aux flux de média unidiffusés entre deux entités, chaque source possédant une clé cryptographique unique; la prise en charge des flux de média multidiffusés ou des flux multipoint unidiffusés appelle un complément d'étude.

4.1 Transport des paramètres SRTP

Une connexion de média SRTP en duplex intégral fait intervenir deux canaux unidirectionnels, un dans chaque sens. Chaque offre cryptographique est transportée dans un message **OpenLogicalChannel** H.245 distinct.

4.1.1 Transport du paramètre SrtpKeys

Le paramètre **SrtpKeys** contenant les données de clé cryptographique SRTP doit être transporté dans le champ **genericKeyMaterial** du paramètre **secureSharedSecret** (**V3KeySyncMaterial**) figurant dans le conteneur **h235Key** dans le paramètre **encryptionSync** des messages **OpenLogicalChannel** H.245.

Le contenu de clé cryptographique SRTP figurant dans le conteneur **genericKeyMaterial** doit être identifié au moyen de la valeur d'identificateur d'objet H.235.8 (voir Tableau 1) dans le champ **standard** de **capabilityIdentifier** dans le champ **genericH235SecurityCapability** de **encryptionAuthenticationAndIntegrity** dans **h235Media** du paramètre **dataType** de message OLC.

D'autres propositions **OpenLogicalChannel** pour le même canal contenant la même valeur **sessionID** dans **H2250LogicalChannelParameters** peuvent utiliser la même offre cryptographique. Etant donné qu'une seule de ces différentes sessions sera acceptée, l'univocité de clé sera garantie.

4.1.2 Transport du paramètre SrtpCryptoCapability

Le paramètre **SrtpCryptoCapability** doit être transporté dans le champ **genericH235SecurityCapability** de **encryptionAuthenticationAndIntegrity** dans **h235Media** du paramètre **dataType** des messages **OpenLogicalChannel**.

Le message **TerminalCapabilitySet** H.245 peut inclure une ou plusieurs entrées **h235SecurityCapability** dans le tableau **capabilityTable**. Afin d'indiquer la prise en charge de ces procédures, le point d'extrémité H.323 doit définir **genericH235SecurityCapability** dans **encryptionAuthenticationAndIntegrity** dans une entrée **h235SecurityCapability** comme suit:

- **capabilityIdentifier** doit contenir l'identificateur d'objet H.235.8 (voir Tableau 1) dans le champ **standard**;
- **maxbitRate, collapsing**, **nonCollapsing** et **transport** doivent rester inutilisés;
- **nonCollapsingRaw** doit contenir le paramètre **SrtpCryptoCapability**.

Tableau 1/H.235.8 – Identificateur d'objet H.235.8

Valeur de l'identificateur d'objet
{ itu-t (0) recommendation (0) h (8) 235 version (0) 4 90 }

4.2 Description du paramètre SrtpCryptoCapability

Le paramètre **SrtpCryptoCapability** peut contenir un ou plusieurs paramètres **SrtpCryptoInfo** pouvant servir à spécifier des capacités pour la session SRTP. Les éléments **BOOLEAN OPTIONAL** doivent être interprétés de la façon suivante:

1) si l'élément vaut FALSE, la capacité n'est pas prise en charge;

2) si l'élément vaut TRUE, la capacité est prise en charge et requise;

3) si l'élément est absent, la capacité est prise en charge mais n'est pas requise.

Lorsque le paramètre **SrtpCryptoCapability** est utilisé dans un échange de capacités, il est possible d'indiquer toutes les options acceptables à l'intérieur d'une seule capacité générique. Dans ce cas, l'omission d'un élément **BOOLEAN OPTIONAL** signifiera que la capacité est prise en charge mais n'est pas requise.

Lorsque ce paramètre est utilisé dans une expression **dataType** de message OLC, une seule option peut être utilisée. A cette fin, les règles suivantes doivent être observées:

- **FecOrder** ne peut contenir qu'une des valeurs optionnelles;
- dans **SrtpSessionParameters**, les valeurs **BOOLEAN OPTIONAL** doivent être soit TRUE soit FALSE;
- **SrtpCryptoCapability** doit contenir un seul élément **SrtpCryptoInfo**.

Le paramètre **SrtpCryptoInfo** se compose du champ obligatoire **cryptoSuite** et des champs optionnels **sessionParams** et **allowMKI** qui sont décrits ci-après.

Tableau 2/H.235.8 – Identificateurs d'objet des suites cryptographiques H.235.8

Suite cryptographique	Valeur de l'identificateur d'objet
AES_CM_128_HMAC_SHA1_80	{ itu-t (0) recommendation (0) h (8) 235 version (0) 4 91 }
AES_CM_128_HMAC_SHA1_32	{ itu-t (0) recommendation (0) h (8) 235 version (0) 4 92 }
F8_128_HMAC_SHA1_80	{ itu-t (0) recommendation (0) h (8) 235 version (0) 4 93 }

4.2.1 cryptoSuite

L'identificateur d'objet (voir Tableau 2) contenu dans le champ **cryptoSuite** indique les algorithmes de chiffrement et d'authentification à utiliser dans la session SRTP. La spécification SRTP comporte de nombreux paramètres qui sont regroupés en trois options, appelées "suites cryptographiques". Ces options peuvent être élargies en ce sens que de nouvelles suites cryptographiques peuvent être ajoutées. Les trois suites cryptographiques qui ont été définies sont les suivantes: AES_CM_128_HMAC_SHA1_80, AES_CM_128_HMAC_SHA1_32 et F8_128_HMAC_SHA1_80. Les paramètres SRTP qui sont associés à chacune de ces suites sont présentés dans les lignes du Tableau 3.

Tableau 3/H.235.8 – Valeurs par défaut des suites cryptographiques

Paramètre SRTP	AES_CM_128_ HMAC_SHA1_80	AES_CM_128_ HMAC_SHA1_32	F8_128_ HMAC_SHA1_80
Longueur de la clé maîtresse	128 bits	128 bits	128 bits
Valeur du sel	112 bits	112 bits	112 bits
Durée de vie	2^31 paquets	2^31 paquets	2^31 paquets
Chiffre	Compteur AES	Compteur AES	F8
Clé de chiffrement	128 bits	128 bits	128 bits
MAC	HMAC-SHA1	HMAC-SHA1	HMAC-SHA1
Longueur d'étiquette d'authentification	80 bits	32 bits	80 bits
Longueur de clé d'authentification SRTP	160 bits	160 bits	160 bits
Longueur de clé d'authentification SRTCP	160 bits	160 bits	160 bits

Le champ **cryptoSuite** est un paramètre négocié.

4.2.2 sessionParams

Les paramètres de session peuvent être soit négociés, soit déclaratifs; la définition d'un paramètre de session particulier doit indiquer si ce dernier est négocié ou déclaratif. Les paramètres négociés s'appliquent aux données envoyées dans les deux sens, alors que les paramètres déclaratifs ne s'appliquent qu'aux médias envoyés par l'entité qui a produit la description de session. Par conséquent, un paramètre déclaratif dans une offre s'applique aux médias envoyés par l'entité offrante, alors qu'un paramètre déclaratif dans une réponse s'applique aux médias envoyés par l'entité répondante.

Le champ optionnel **sessionParams** contient les paramètres de session SRTP.

4.2.2.1 kdr

Le paramètre KDR spécifie la fréquence de calcul de clé, tel que décrit dans le § 4.3.1 de la norme RFC 3711. La valeur de ce paramètre doit être un entier compris dans l'ensemble {1, 2, ..., 24}, qui représente une puissance de 2, de 2^1 à 2^24 inclus. La fréquence de calcul de clé SRTP permet de déterminer la fréquence d'obtention d'une nouvelle clé de session à partir d'une clé maîtresse SRTP (RFC 3711). Lorsque la fréquence de calcul de clé n'est pas spécifiée (c'est-à-dire lorsque le paramètre KDR est omis), un calcul de clé initial unique est effectué (RFC 3711). KDR est un paramètre déclaratif.

4.2.2.2 unencryptedSrtp

Il s'agit d'un champ booléen optionnel; s'il est présent, il signale que les charges utiles des paquets SRTP ne sont pas chiffrées. unencryptedSrtp est un paramètre négocié.

4.2.2.3 unencryptedSrtcp

Il s'agit d'un champ booléen optionnel; s'il est présent, il signale que les charges utiles des paquets SRTCP ne sont pas chiffrées. unencryptedSrtcp est un paramètre négocié.

4.2.2.4 unauthenticatedSrtp

Les charges utiles des paquets SRTP et SRTCP sont authentifiées par défaut. unauthenticatedSrtp est un champ booléen optionnel; s'il est présent, il signale que les charges utiles des paquets SRTP

ne sont pas authentifiées. Selon la spécification du protocole SRTP, l'authentification des messages est requise pour SRTCP, mais pas pour SRTP (RFC 3711). unauthenticatedSrtp est un paramètre négocié.

4.2.2.5 fecOrder

Le paramètre **fecOrder** indique l'ordre du traitement de correction d'erreur directe (FEC) des paquets RTP (RFC 3550, RFC 2733) par rapport au chiffrement SRTP au niveau de l'émetteur. La valeur **fecBeforeSrtp** du paramètre **fecOrder** indique que la fonction FEC est appliquée avant le traitement SRTP effectué par l'émetteur des médias SRTP et après le traitement SRTP effectué par le récepteur des médias SRTP; **fecBeforeSrtp** est la valeur par défaut. **fecAfterSrtp** indique l'ordre inverse du traitement. **fecOrder** est un paramètre déclaratif.

4.2.2.6 windowSizeHint

Le protocole SRTP définit le paramètre SRTP-WINDOW-SIZE (taille de fenêtre SRTP) (RFC 3711, section 3.3.2) visant à assurer la protection contre les attaques par réexécution. La valeur minimale est 64 (RFC 3711), mais cette valeur peut être considérée comme étant trop petite pour certaines applications (par exemple pour la vidéo).

Le paramètre de session d'indication de taille de fenêtre (WSH, *window size hint*) donne, à titre indicatif, la taille appropriée de cette fenêtre (par exemple, sur la base du nombre de paquets par seconde connu par l'émetteur). Toutefois, les informations données par les descripteurs de mise en paquets des médias peuvent suffire à un récepteur pour déterminer le paramètre de façon satisfaisante. En conséquence, la valeur de ce paramètre n'est donnée qu'à titre indicatif au récepteur qui peut choisir de l'ignorer.

windowSizeHint est un paramètre déclaratif.

4.2.2.7 Définition de nouveaux paramètres de session SRTP

De nouveaux paramètres de session SRTP sont par défaut obligatoires. Le champ **newParameter** est destiné à ajouter de nouveaux paramètres de session. Si un ancien point d'extrémité H.323 reçoit un paramètre **SrtpCryptoInfo** avec un paramètre de session inconnu dans le champ **newParameter**, ce nouveau paramètre **SrtpCryptoInfo** sera considéré comme étant non valide.

4.3 Description des paramètres du champ SrtpKeys

Le champ **SrtpKeys** contient un ou plusieurs paramètres de clé **SrtpKeyParameter** devant être utilisés pour la session SRTP. Chaque paramètre **SrtpKeyParameter** contient les données de clé (clé maîtresse et sel) ainsi que toutes les politiques relatives à la clé maîtresse, y compris la durée pendant laquelle celle-ci peut être utilisée (durée de vie) et la question de savoir si elle utilise un identificateur de clé maîtresse (MKI, *master key identifier*) pour associer un paquet SRTP entrant à une clé maîtresse particulière. Les implémentations conformes respecteront les politiques associées à une clé maîtresse et n'accepteront pas les paquets entrants qui violent ces politiques (par exemple, après expiration de la durée de vie de la clé maîtresse).

4.3.1 masterKey

Il s'agit de la clé maîtresse cryptographique à utiliser pour la session SRTP. La longueur de cette clé est déterminée par la suite cryptographique à laquelle s'applique la clé. Si la longueur ne correspond pas à celle spécifiée pour la suite cryptographique, le paramètre "crypto" en question sera considéré comme étant non valide. Chaque clé maîtresse doit représenter un nombre aléatoire sur le plan cryptographique et doit être unique pour chaque flux de média proposé.

4.3.2 masterSalt

Il s'agit du sel maître cryptographique à utiliser pour la session SRTP. La longueur du sel est déterminée par la suite cryptographique à laquelle s'applique la clé. Si la longueur ne correspond pas à celle spécifiée pour la suite cryptographique, le paramètre "crypto" en question sera considéré comme étant non valide. Chaque sel maître doit représenter un nombre aléatoire sur le plan cryptographique et doit être unique pour chaque flux de média proposé.

4.3.3 lifetime

Ce champ représente la durée de vie optionnelle de la clé maîtresse, mesurée par le nombre maximal de paquets SRTP ou SRTCP utilisant cette clé maîtresse (le nombre de paquets SRTP et le nombre de paquets SRTCP doivent tous les deux être inférieurs à la durée de vie). La valeur de la durée de vie peut être écrite sous la forme d'un entier positif non nul ou d'une puissance de 2. La valeur "lifetime" ne doit pas dépasser la durée de vie maximale des paquets pour la suite cryptographique. Si la durée de vie est trop longue ou bien non valide, le paramètre "crypto" entier sera considéré comme étant non valide. Si le champ lifetime n'est pas présent, la valeur de la durée de vie par défaut doit être utilisée. Cela est pratique lorsque la durée de vie de la clé cryptographique SRTP est la valeur par défaut.

4.3.4 masterKeyId

Ce champ optionnel indique la politique concernant la manière dont les clés doivent être identifiées pour la session SRTP. MKI est l'identificateur de la clé maîtresse SRTP. Si l'identificateur MKI est donné, sa longueur doit également être fournie. La longueur de MKI correspond à la taille du champ MKI, spécifié en octets, dans le paquet SRTP. Si elle n'est pas donnée ou que sa valeur dépasse 128 (octets), le paramètre "crypto" entier sera considéré comme étant non valide.

Comme il a été mentionné ci-dessus, le paramètre de clé peut contenir une ou plusieurs clés maîtresses. Lorsqu'il contient plus d'une clé maîtresse, toutes les clés maîtresses doivent inclure une valeur MKI. En cas d'utilisation de l'identificateur MKI, la longueur de ce dernier doit être la même pour toutes les clés contenues dans un paramètre "crypto" donné.

4.4 Initialisation de contexte cryptographique SRTP

Outre les divers paramètres SRTP définis ci-dessus, trois informations sont essentielles au fonctionnement des chiffres SRTP par défaut:

- SSRC: source de synchronisation
- ROC: compteur de cycles complets pour une source SSRC donnée
- SEQ: numéro de séquence pour une source SSRC donnée

Dans une session d'unidiffusion, telle que définie dans la présente Recommandation, les valeurs des paramètres ci-dessus doivent satisfaire trois contraintes. Selon la première contrainte, concernant la source SSRC, un flux de clé SRTP doit être unique pour chaque participant. Comme il est expliqué dans le protocole SRTP, le flux de clé ne doit pas être réutilisé sur deux textes en clair différents ou plus.

La réutilisation du flux de clé rend le texte chiffré vulnérable à l'analyse cryptographique. Par exemple, des champs en clair connus dans un flux peuvent révéler des parties du flux de clé réutilisé, ce qui pourrait dévoiler encore plus de texte en clair dans d'autres flux. Etant donné que tous les mécanismes de chiffrement SRTP actuels utilisent des flux de clé, le partage de clé est un problème général (RFC 3711). Le protocole SRTP limite ce problème en incluant la source SSRC de l'émetteur dans le flux de clé. Cependant, il ne résout pas complètement ce problème dans la mesure où le protocole de transport en temps réel peut être à l'origine de collisions de sources SSRC, qui sont certes très rares (RFC 3550) mais possibles. Au cours d'une collision, deux sources SSRC ou plus qui partagent une clé maîtresse possèderont des flux de clé identiques pour des

parties qui se chevauchent de l'espace du numéro de séquence RTP. La description de sécurité SRTP évite la réutilisation de flux de clé en imposant des clés maîtresses uniques pour l'émetteur et le récepteur de cette description. La première contrainte est ainsi satisfaite.

Il convient également de noter que les collisions de sources SSRC posent un second problème: la source SSRC sert à identifier le contexte cryptographique et, par là-même, le chiffre, la clé, le compteur ROC, etc., afin de traiter les paquets entrants. En cas de collisions de sources SSRC, l'identification du contexte cryptographique devient ambiguë, ce qui empêche le traitement correct des paquets. Par ailleurs, si un paquet BYE RTCP doit être envoyé pour une source SSRC entrant en collision, il peut également être nécessaire de protéger ce paquet.

La seconde contrainte réside dans le fait que le compteur ROC doit être mis à zéro au moment où chaque source SSRC commence à envoyer des paquets. Ainsi, la notion de "retardataire" n'existe pas dans les descriptions de sécurité SRTP, car les flux sont unidiffusés entre deux entités. Le compteur ROC et le numéro SEQ forment un "indice de paquet" dans les transformations SRTP par défaut, le compteur ROC étant systématiquement mis à zéro au début de la session, conformément à la présente Recommandation.

La troisième contrainte réside dans le fait que la valeur initiale du numéro SEQ doit être choisie dans l'intervalle 0..2^15 – 1; cela permet d'éviter toute ambiguïté lorsque des paquets se perdent au début de la session. Si au début d'une session, une source SSRC choisit aléatoirement une valeur élevée pour le numéro de séquence, elle met le récepteur dans une situation ambiguë: en cas de perte des paquets initiaux en transit jusqu'à un nouveau cycle du numéro de séquence (c'est-à-dire si le numéro de séquence dépasse 2^16 – 1), le récepteur peut ne pas se rendre compte que son compteur ROC doit être incrémenté. En imposant une valeur SEQ initiale dans l'intervalle 0..2^15 – 1, la détermination de l'indice de paquet SRTP permettra de trouver la valeur correcte du compteur ROC, à moins que la totalité des 2^15 premiers paquets se perdent (ce qui paraît très peu probable même si cela n'est pas impossible). Voir la section 3.3.1 de la spécification du protocole SRTP en ce qui concerne la détermination de l'indice de paquet (RFC 3771).

4.4.1 Rattachement différé de sources SSRC à un contexte cryptographique

L'indice de paquet dépend par conséquent de la source SSRC, du numéro SEQ d'un paquet entrant et du compteur ROC, lequel représente une variable de contexte cryptographique SRTP. Ainsi, le protocole SRTP dépend grandement de l'univocité de la source SSRC en ce qui concerne la sécurité. Compte tenu des contraintes susmentionnées, il est possible d'établir des contextes cryptographiques SRTP unidiffusés sans avoir à négocier de valeurs SSRC dans la description de sécurité SRTP. La spécification SRTP recommande au contraire une méthode appelée "rattachement différé" (*late binding*). Lorsqu'un paquet arrive, la source SSRC qui y est contenue peut être rattachée au contexte cryptographique au début d'une session (c'est-à-dire à l'arrivée du paquet SRTP) plutôt qu'au moment de la signalisation de session (c'est-à-dire à la réception d'un message H.245). A l'arrivée du paquet contenant la source SSRC, toutes les données nécessaires pour le contexte cryptographique SRTP sont traitées par le récepteur (à noter que la valeur du compteur ROC est par définition zéro; si des valeurs non nulles devaient être prises en charge, d'autres données de signalisation seraient nécessaires). En d'autres termes, le contexte cryptographique associé à une session RTP sécurisée utilisant le rattachement différé est initialement identifié par le message H.245 sous la forme:

<*, address, port>

où * est une source SSRC générique, "address" est l'adresse de réception locale provenant de **mediaChannel** et "port" est le port de réception local provenant de **portNumber**. A l'arrivée du premier paquet contenant ssrcX dans son champ SSRC, le contexte cryptographique

<ssrcX, address, port>

est instancié compte tenu des contraintes suivantes:

- les paquets de média sont authentifiés: L'authentification doit réussir; sinon, le contexte cryptographique n'est pas instancié;
- les paquets de média ne sont pas authentifiés: Le contexte cryptographique est automatiquement instancié.

Il convient de noter que l'utilisation du rattachement différé en l'absence d'authentification des paquets de média SRTP n'est pas recommandée en raison du risque élevé d'attaque à la sécurité (cela est évidemment valable pour la non-authentification SRTP en général).

A noter en outre qu'en cas d'utilisation du rattachement différé sans authentification, un état local sera créé à la réception d'un paquet provenant de toute source SSRC inconnue. La non-authentification SRTP n'est donc pas recommandée, car elle favorise les attaques de type déni de service, ce qui n'est pas le cas du rattachement différé avec authentification.

4.4.2 Partage de contextes cryptographiques entre sessions ou sources SSRC

Compte tenu des contraintes et procédures décrites ci-dessus, il n'est pas nécessaire de signaler explicitement la source SSRC, le compteur ROC et le numéro SEQ pour une session RTP d'unidiffusion. Ainsi, aucun paramètre "crypto" SRTP n'est prévu pour la signalisation de ces éléments. Par conséquent, en cas de rattachement différé, plusieurs sources SSRC relevant de la même entité partageront les paramètres crypto SRTP. La multiplicité de sources SSRC relevant d'une même entité est due soit à la présence de sources multiples (microphones, caméras, etc.), soit au fait que les charges utiles RTP nécessitent un multiplexage des sources SSRC à l'intérieur de cette même session.

Le protocole H.245 autorise la définition de plusieurs sessions RTP dans une même description de média; ces sessions RTP partageront elles aussi les paramètres crypto SRTP. Une application qui utilise le paramètre crypto SRTP de cette façon partage une clé maîtresse entre les différentes sessions RTP ou sources SSRC et doit modifier la clé maîtresse lorsque le nombre cumulé de paquets pour toutes les sources SSRC approche 2^31 paquets. Les sources SSRC qui partagent une clé maîtresse seront chacune unique.

La durée de vie de toutes les clés qui sont obtenues à partir d'une clé maîtresse est déterminée par la durée de vie de celle-ci. Ainsi, si la durée de vie de la clé maîtresse est 2^31 paquets et qu'une clé dérivée a envoyé 2^31 – y paquets, seuls y paquets pourront être envoyés par toute clé dérivée de cette clé maîtresse. Cela est dû au fait que la durée de vie est fonction de l'entropie ou du caractère aléatoire de la clé et au fait que le caractère aléatoire n'est pas augmenté lors de la détermination d'une clé à partir d'une clé maîtresse, le caractère aléatoire ou l'entropie étant des paramètres inhérents à la clé.

4.4.3 Suppression de contextes cryptographiques

Le mécanisme défini ci-dessus permet de créer des contextes cryptographiques; toutefois, dans la pratique, des participants de sessions peuvent aussi souhaiter supprimer des contextes cryptographiques avant la fin d'une session. Etant donné qu'un contexte cryptographique contient des informations qui ne peuvent pas être automatiquement récupérées (par exemple le compteur ROC), il importe que l'émetteur et le récepteur se mettent d'accord sur les conditions dans lesquelles un contexte cryptographique peut être supprimé et, ce qui est certainement plus important, sur les conditions dans lesquelles un contexte cryptographique ne peut pas être supprimé.

Même lorsque le rattachement différé est utilisé pour un flux unidiffusé, le compteur ROC est perdu et ne peut pas être automatiquement récupéré (sauf s'il est mis à zéro) une fois que le contexte cryptographique est supprimé.

La suppression de contextes cryptographiques doit s'effectuer à la réception d'un message **CloseLogicalChannel**. Par ailleurs, cette suppression sera soumise aux mêmes règles que celles qui

régissent la suppression de sources SSRC du tableau des membres (RFC 3711); à noter que cette opération peut résulter d'un paquet BYE SRTCP ou d'une simple expiration de temporisation due à l'inactivité. Les participants de sessions inactifs qui veulent empêcher l'expiration de la temporisation associée à leurs contextes cryptographiques doivent par conséquent envoyer des paquets SRTCP à des intervalles réguliers.

5 Procédures

Les procédures SRTP décrites ci-après ne doivent être utilisées que pour négocier la sécurité de flux de média unidiffusés entre deux entités dans des situations où le canal de signalisation H.245 est protégé par un protocole de sécurité de type encapsulation de données par exemple, IPsec (RFC 2401), TLS (RFC 2246). L'échange de paramètres crypto SRTP au moyen de messages H.245 assurera les fonctions suivantes:

1) échange et négociation de capacités de chiffrement et d'intégrité de médias SRTP;
2) négociation et établissement des algorithmes de chiffrement et d'intégrité ainsi que des clés et des paramètres de session initiaux à utiliser pour les flux SRTP dans chaque sens;
3) modification des algorithmes de chiffrement et d'intégrité ainsi que des clés et des paramètres de session à tout moment au cours de la session SRTP.

5.1 Echange de capacités de sécurité

Les suites cryptographiques SRTP ainsi que les algorithmes de chiffrement et d'intégrité qu'un point d'extrémité H.323 peut prendre en charge seront identifiés par **SrtpCryptoCapability**.

L'échange de capacités de sécurité sera assuré par l'échange de capacités entre terminaux au moyen d'une ou de plusieurs entrées **h235SecurityCapability** du tableau **capabilityTable** du message **TerminalCapabilitySet** H.245. Le champ **mediaCapability** dans l'entrée **h235SecurityCapability** du tableau **capabilityTable** est utilisé pour associer la capacité de sécurité à une entrée particulière de capacité de média dans le tableau **capabilityTable**.

Le champ **encryptionAuthenticationAndIntegrity** dans l'entrée **h235SecurityCapability** contient le champ **genericH235SecurityCapability** qui spécifiera les suites cryptographiques SRTP identifiées par les identificateurs d'objet H.235.8. Si le champ **standard** de **capabilityIdentifier** du champ **genericH235SecurityCapability** contient l'identificateur d'objet H.235.8 (voir Tableau 1), la **SrtpCryptoCapability** contiendra un ou plusieurs paramètres **SrtpCryptoInfo** représentant les suites cryptographiques que prend en charge le point d'extrémité H.323. Le champ **cryptoSuite** dans le champ **SrtpCryptoInfo** contient un identificateur d'objet tel que défini dans le Tableau 2 qui identifie une suite cryptographique particulière. A l'intérieur du champ **SrtpCryptoInfo**, le champ **sessionParams** indique les paramètres de session et le champ **allowMKI** précise si l'identificateur MKI est pris en charge par le point d'extrémité H.323.

5.2 Négociation initiale

5.2.1 Offre cryptographique initiale

Chaque offre cryptographique est transportée dans un message **OpenLogicalChannel** distinct; elle doit contenir une structure **SrtpCryptoInfo** dans **SrtpCryptoCapability** ainsi qu'une ou plusieurs structures **SrtpKeyParameters** dans **SrtpKeys**.

Dans le cas des procédures H.245 normales (qui ne sont pas des procédures de connexion rapide), le point d'extrémité H.323 doit inclure l'offre cryptographique telle que décrite dans les structures **SrtpCryptoInfo** et **SrtpKeyParameters** dans un message **OpenLogicalChannel** H.245 pour le sens aller (entre le point d'extrémité H.323 offrant et le point d'extrémité H.323 répondant). Le point d'extrémité H.323 devrait offrir, parmi les capacités de sécurité qu'il prend en charge, celle qui a été

indiquée au cours de l'échange de capacités entre terminaux comme étant la capacité de sécurité préférée du maître.

Dans le cas de procédures de connexion rapide, le point d'extrémité H.323 offrant doit envoyer chaque offre cryptographique décrite dans les structures **SrtpCryptoInfo** et **SrtpKeyParameters** dans des messages **OpenLogicalChannel** H.245 distincts pour le sens aller (entre le point d'extrémité H.323 offrant et le point d'extrémité H.323 répondant).

Les messages **OpenLogicalChannel** offerts doivent être listés par ordre de préférence, la suite cryptographique dont le degré de préférence est le plus élevé étant indiquée en premier. En règle générale, les suites cryptographiques pour lesquelles la préférence est élevée devraient être plus fortes sur le plan cryptographique que les suites cryptographiques pour lesquelles la préférence est moins élevée.

Lorsqu'elle envoie une offre cryptographique, l'entité offrante doit être prête à prendre en charge la sécurité de média conformément à l'un quelconque des paramètres crypto offerts. Deux problèmes se posent alors. Premièrement, l'entité offrante ne connaît pas la clé que l'entité répondante utilisera pour les médias qu'elle lui envoie. Etant donné que les médias peuvent arriver avant la réponse cryptographique, un retard ou une troncature (*clipping*) peut être constaté. Si elle n'accepte pas ce phénomène, l'entité offrante devrait appliquer un mécanisme tel que des procédures d'établissement d'appel différé H.460.11.

Dans le cas de plusieurs offres, un autre problème peut se poser: l'entité offrante n'est pas en mesure de déterminer l'offre que l'entité répondante a acceptée tant que la réponse cryptographique n'a pas été reçue; les médias peuvent cependant arriver avant cette dernière. Si elle n'accepte pas cette situation, l'entité offrante a deux solutions: soit elle n'enverra pas plus d'une offre, soit elle appliquera un mécanisme tel que des procédures d'établissement d'appel différé H.460.11.

La structure **SrtpCryptoInfo** peut inclure des paramètres de session.

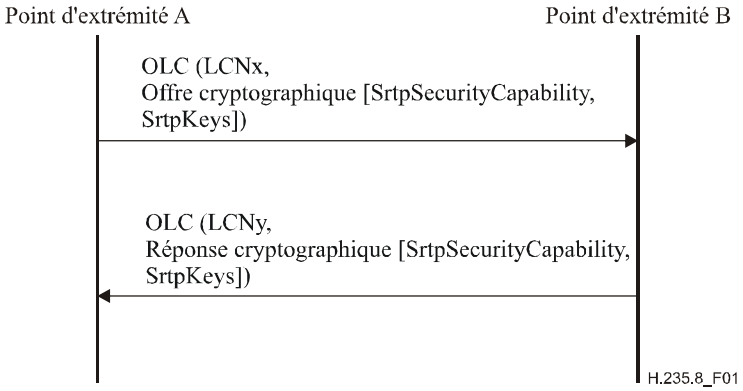

Figure 1/H.235.8 – Echange offre-réponse dans le cas d'une connexion rapide

5.2.1.1 Réponse cryptographique initiale

5.2.1.1.1 Généralités

Ces procédures s'appliquent aussi bien aux procédures de connexion rapide qu'aux procédures H.245 normales. Une réponse cryptographique doit contenir une structure **SrtpCryptoInfo** dans **SrtpCryptoCapability** ainsi qu'une ou plusieurs structures **SrtpKeyParameters** dans **SrtpKeys**.

Le point d'extrémité H.323 répondant doit appliquer la suite cryptographique choisie à partir de l'offre cryptographique envoyée au canal SRTP unidirectionnel correspondant dans le sens retour et doit produire la ou les clés à utiliser pour ce canal SRTP dans le sens retour.

En outre, le point d'extrémité H.323 répondant doit inclure une ou plusieurs clés dans **SrtpKeys** à utiliser pour le flux SRTP entre le point d'extrémité H.323 répondant et le point d'extrémité H.323 offrant. Le point d'extrémité H.323 répondant peut également inclure tout paramètre de session issu de l'offre cryptographique qu'il souhaite négocier.

Seuls les paramètres valides sont acceptés; les paramètres valides ne violent aucune des règles générales définies pour les descriptions de sécurité ni aucune règle particulière définie pour la méthode de transport et de clé en question.

En cas de connexion rapide, lors du choix d'une des offres cryptographiques valides, l'entité répondante devrait retenir l'offre cryptographique la plus élevée en termes de préférence qu'elle peut prendre en charge, c'est-à-dire le premier paramètre pris en charge valide de la liste, compte tenu des capacités de l'entité répondante et des politiques de sécurité appliquées. Si aucune des offres n'est valide, ou aucune de celles qui sont valides n'est prise en charge, le flux de média offert doit être rejeté.

Lorsqu'une offre cryptographique est acceptée, la réponse cryptographique doit contenir la ou les clés que l'entité répondante utilisera pour le média envoyé à l'entité offrante. A noter qu'une clé doit être fournie quels que soient les paramètres de direction contenus dans l'offre ou dans la réponse.

Par ailleurs, tout paramètre de session négocié doit être inclus dans la réponse cryptographique. Les paramètres de session déclaratifs fournis par l'entité offrante ne sont pas inclus dans la réponse cryptographique, mais l'entité répondante peut fournir son propre ensemble de paramètres de session déclaratifs.

Une fois qu'elle a accepté l'un des paramètres crypto offerts, l'entité qui répond peut commencer à envoyer les médias à l'entité à l'origine de l'offre conformément à l'offre cryptographique choisie. A noter toutefois que l'entité à l'origine de l'offre peut ne pas être en mesure de traiter correctement ces paquets de média tant que la réponse cryptographique n'a pas été reçue.

5.2.1.1.2 Procédures de connexion rapide

Dans le cas de procédures de connexion rapide, le point d'extrémité H.323 répondant qui reçoit les offres cryptographiques dans un ou plusieurs messages **OpenLogicalChannel** H.245 doit répondre en acceptant l'une des offres par l'envoi d'un message **OpenLogicalChannel** H.245 contenant la réponse cryptographique telle que représentée dans la Figure 1, ou en rejetant toutes les offres cryptographiques par l'envoi d'un message **ReleaseComplete** avec **ReleaseCompleteReason** mis à **securityDenied**, ou par l'envoi d'un élément **FastConnectRefused** dans un message H.225.0. S'il ne prend pas en charge la présente Recommandation ou aucune des propositions contenues dans l'offre cryptographique en question, le point d'extrémité H.323 répondant doit rejeter cette dernière en envoyant un message **ReleaseComplete** avec **ReleaseCompleteReason** mis à **securityDenied**, ou en envoyant un élément **FastConnectRefused** dans un message H.225.0.

5.2.1.1.3 Procédures H.245 normales

Dans le cas de procédures H.245 normales (qui ne sont pas des procédures de connexion rapide), on appliquera la méthode qui suit. S'il n'a pas déjà envoyé un message **OpenLogicalChannel** contenant une offre cryptographique avant qu'il en ait reçu un, le point d'extrémité H.323 doit envoyer un message **OpenLogicalChannelAck** suivi d'un message **OpenLogicalChannel** contenant la réponse cryptographique telle que représentée dans la Figure 2.

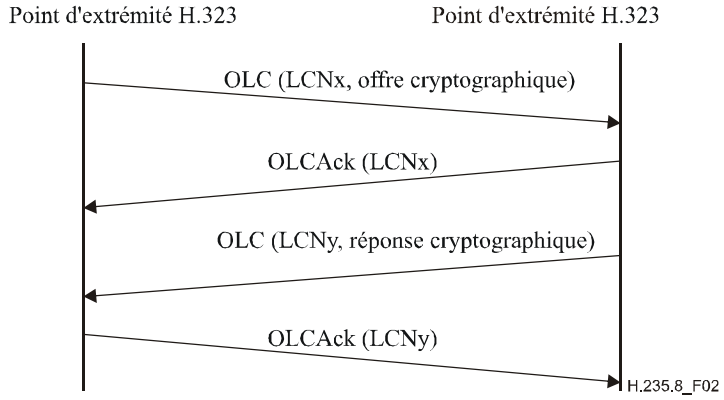

Figure 2/H.235.8 – Echange offre-réponse

Si le point d'extrémité H.323 a déjà envoyé un message **OpenLogicalChannel** contenant une offre cryptographique avant qu'il en ait reçu un, les points d'extrémité H.323 maître et esclave procéderont ainsi:

1) un point d'extrémité H.323 maître doit traiter l'offre cryptographique reçue et, si celle-ci est compatible avec l'offre cryptographique qu'il a déjà envoyée, il doit accepter l'offre cryptographique reçue comme réponse cryptographique en envoyant un message **OpenLogicalChannelAck**, comme le montre la Figure 3. Si l'offre cryptographique reçue n'est pas compatible avec l'offre cryptographique qu'il a déjà envoyée, il doit rejeter l'offre cryptographique reçue en envoyant un message **OpenLogicalChannelReject** avec la valeur **cause** de **securityDenied**, comme le montre la Figure 4. Le terme "compatible" signifie que les paramètres qui suivent dans l'offre cryptographique doivent correspondre aux paramètres contenus dans la réponse cryptographique: **cryptoSuite** et les paramètres de session négociés;

2) un point d'extrémité H.323 esclave doit traiter l'offre cryptographique reçue et, si cette dernière est compatible avec l'offre cryptographique qu'il a déjà envoyée, il doit accepter l'offre cryptographique reçue comme réponse cryptographique en envoyant un message **OpenLogicalChannelAck**, comme le montre la Figure 3. Si l'offre cryptographique reçue n'est pas compatible avec l'offre cryptographique qu'il a déjà envoyée et s'il souhaite l'accepter, il doit le faire en envoyant les messages qui suivent représentés dans la Figure 4.

   a) **OpenLogicalChannelAck** pour accepter l'offre cryptographique initiale envoyée par le maître.

   b) **CloseLogicalChannel** pour mettre fin à sa propre offre cryptographique initiale si **OpenLogicalChannelReject** n'a pas déjà été reçu du maître.

   c) **OpenLogicalChannel** avec une réponse cryptographique correspondant à l'offre cryptographique envoyée par le maître.

S'il ne prend pas en charge la proposition contenue dans l'offre ou s'il ne souhaite pas accepter l'offre cryptographique, le point d'extrémité H.323 esclave doit rejeter cette dernière en envoyant un message **OpenLogicalChannelReject** avec **cause** mis à **securityDenied.**

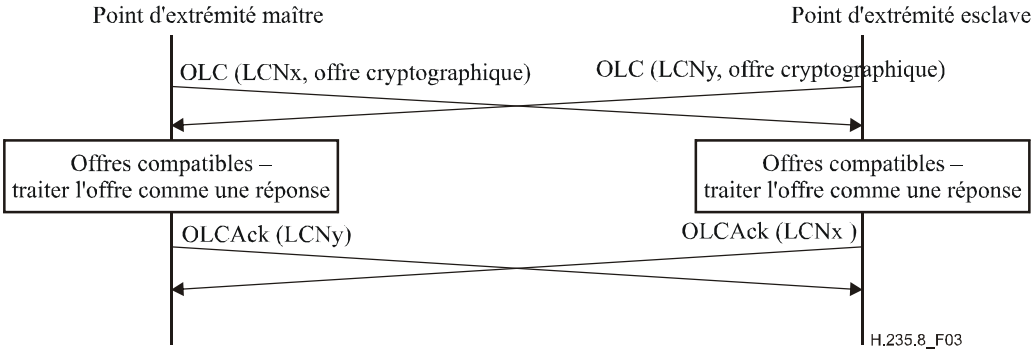

Figure 3/H.235.8 – Echange simultané offre-réponse compatibles

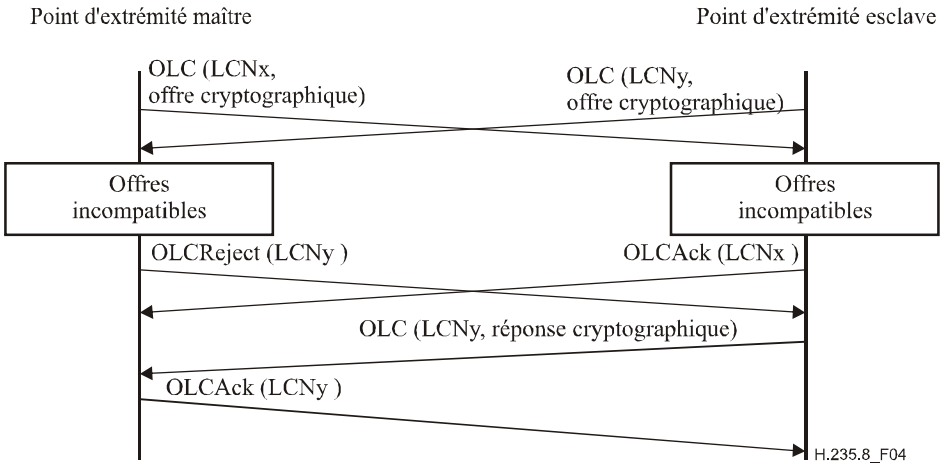

Figure 4/H.235.8 – Echange simultané offre-réponse incompatibles

5.2.1.2 Traitement de la réponse initiale par l'entité offrante

Lorsqu'elle reçoit la réponse cryptographique, l'entité offrante doit vérifier qu'une des offres cryptographiques initiales a été acceptée et est indiquée dans la réponse cryptographique. Par ailleurs, cette dernière doit inclure une ou plusieurs clés qui seront utilisées pour les médias que l'entité répondante envoie à l'entité offrante.

L'entité offrante doit vérifier que les clés contenues dans la réponse cryptographique ne correspondent à aucune des clés contenues dans l'offre cryptographique. Si cette dernière contient des paramètres de session négociés obligatoires, l'entité offrante doit vérifier que lesdits paramètres sont inclus dans la réponse cryptographique et correspondent aux paramètres contenus dans l'offre cryptographique. Si la réponse cryptographique contient des paramètres de session déclaratifs obligatoires, l'entité offrante doit être en mesure de les prendre en charge.

Si une quelconque des actions susmentionnées échoue, la négociation sera considérée comme ayant échoué.

5.3 Modification de session

Une fois qu'il a été établi, le flux de média SRTP peut être modifié à tout moment au moyen de nouveaux échanges offre-réponse afin d'effectuer un recalcul de clé ou de modifier la suite cryptographique. La nouvelle offre cryptographique et la nouvelle réponse cryptographique doivent être transportées dans les paramètres **SrtpCryptoCapability** et **SrtpKeys** d'un message **OpenLogicalChannel** H.245 de manière à ouvrir un nouveau canal logique qui remplacera celui en place au moyen des procédures **replacementFor**. Le point d'extrémité H.323 offrant doit inclure les

offres cryptographiques dans un ou plusieurs messages **OpenLogicalChannel** H.245 à l'intérieur d'un message H.225.0.

Le point d'extrémité H.323 répondant qui reçoit les offres cryptographiques doit répondre en acceptant l'une des offres par l'envoi d'un message **OpenLogicalChannel** H.245 dans un message H.225.0 ou en rejetant les offres au moyen d'un message **OpenLogicalChannelReject** avec **cause** mis à **securityDenied**. Si l'offre cryptographique est rejetée, les anciens paramètres crypto restent en place.

Lors de l'établissement d'une nouvelle clé maîtresse, une fenêtre de temps est prévue au cours de laquelle le point d'extrémité H.323 doit recevoir les médias chiffrés conformément à l'ancien et au nouvel échange offre-réponse. L'identificateur MKI issu du paquet SRTP entrant doit être utilisé pour associer ce paquet soit à l'ancienne clé maîtresse soit à la nouvelle. Pour cette raison, s'il est prévu que les clés soient modifiées au cours d'une session qui ne modifie ni les adresses et ni les ports d'origine/de destination, l'utilisation de l'identificateur MKI est obligatoire pour permettre au récepteur d'identifier les données associées aux clés lors de la modification de ces dernières.

5.4 Absence de négociation

Dans le cas où aucune négociation des paramètres de suite cryptographique, de clé cryptographique ou de session n'est prévue, l'émetteur détermine les paramètres de sécurité relatifs au flux en question. En l'absence de mécanisme de négociation, l'émetteur doit prévoir exactement une offre cryptographique et le récepteur doit soit l'accepter soit la rejeter en envoyant un message **ReleaseComplete** avec **ReleaseCompleteReason** mis à **securityDenied** ou un message **OpenLogicalChannelReject** avec **cause** mis à **securityDenied**. L'émetteur devrait choisir la description de sécurité qu'il juge la plus sûre pour ses fins.

5.5 Correction d'erreur directe

Une clé maîtresse différente doit être spécifiée pour protéger un flux FEC qui est envoyé à une paire adresse et/ou port IP différente de celle du flux de média SRTP auquel il s'applique, comme il est décrit dans la norme RFC 2733 section 11.1. Ce flux FEC doit être établi au moyen d'un message **OpenLogicalChannel** H.245 distinct avec **dataType** mis à **fec**. La clé maîtresse associée au flux FEC doit être transportée dans le champ **genericKeyMaterial** du paramètre **secureSharedSecret** (**V3KeySyncMaterial**) figurant dans le conteneur **h235Key** dans le paramètre **encryptionSync** du message **OpenLogicalChannel** H.245. La clé maîtresse doit être différente de toutes les autres clés maîtresses offertes pour le flux de média associé.

6 Cryptographie à clé publique pour la protection de l'échange de clés dans le protocole SRTP

Il est possible d'appliquer des procédures de cryptographie à clé publique supplémentaires afin d'assurer la confidentialité et l'authentification de bout en bout des données de clé de session SRTP échangées entre deux points d'extrémité H.323 en chiffrant puis en signant ces données. La cryptographie à clé publique peut être utilisée dans le cas où le protocole de sécurité d'encapsulation (par exemple IPsec, TLS) ne s'applique que jusqu'à un dispositif intermédiaire et où, par conséquent, il n'assure pas la sécurité de bout en bout.

La clé de session SRTP qui chiffre les médias SRTP entre le point d'extrémité appelant et le point d'extrémité appelé est chiffrée au moyen de la clé publique du point d'extrémité appelé, et signée au moyen de la clé privée du point d'extrémité appelant. De la même façon, une autre clé de session SRTP qui chiffre les médias SRTP entre le point d'extrémité appelé et le point d'extrémité appelant doit être chiffrée au moyen de la clé publique du point d'extrémité appelant, et signée au moyen de la clé privée du point d'extrémité appelé. La procédure décrite dans le présent paragraphe peut être appliquée jusqu'à une passerelle, un portier ou un point d'extrémité.

La clé de session SRTP doit être transportée au moyen de corps de la syntaxe de message cryptographique (CMS, *cryptographic message syntax*) dans des messages H.245. La syntaxe de message cryptographique (RFC 3852) sert à signer et chiffrer numériquement un contenu de message arbitraire. La syntaxe CMS permet d'effectuer des encapsulations multiples et ainsi d'imbriquer des enveloppes d'encapsulation les unes dans les autres. En particulier, les données de clé de session SRTP doivent être transportées dans un corps **EnvelopedData** CMS signé au moyen d'un corps **SignedData** CMS.

6.1 Identification de points d'extrémité

On utilisera les éléments suivants pour identifier un point d'extrémité, une passerelle ou un portier dans un certificat de clé publique:

- URL H.323;
- URL normalisée non-H.323 (par exemple, *tel*);
- identification/certificat de dispositif (FFS).

Un certificat de clé publique doit être utilisé pour déclarer l'association de l'identité du point d'extrémité à sa clé publique. L'URL H.323 ou une URL normalisée non H.323 doit être stockée dans le champ **subjectAltName** du certificat.

Les points d'extrémité peuvent gérer une mémoire de clé locale contenant les certificats de clé publique d'autres points d'extrémité avec lesquels ils souhaitent établir des communications de bout en bout sécurisées. Un point d'extrémité qui envoie un contenu signé pour assurer une authentification de bout en bout doit prévoir un certificat de clé publique contenant la clé publique nécessaire pour vérifier la signature. Un point d'extrémité récepteur doit:

a) soit vérifier que le certificat de l'émetteur est signé par une autorité de certification reconnue (CA, *certification authority*);

b) soit faire confiance à une assertion de sécurité relative au certificat fournie par un tiers. L'assertion doit être signée au moyen de données de clé globalement vérifiables.

NOTE – Cela peut être avantageux dans des scénarios où une infrastructure PKI globale pour les utilisateurs n'est pas disponible et où des certificats autosignés ou des certificats de dispositif sont utilisés.

6.2 Procédures d'échange de clé SRTP

S'ils souhaitent garantir la confidentialité et l'authentification de bout en bout de leurs données de clé de session SRTP dans le cas où l'appel, lors de son établissement, traverse un ou plusieurs dispositifs de signalisation intermédiaires, les points d'extrémité appelant et appelé devraient utiliser la cryptographie à clé publique et l'échange de certificats de clé publique X.509 (RFC 3280).

Les procédures offre-réponse décrites dans les paragraphes précédents du présent document restent inchangées à l'exception des points indiqués ci-après.

6.2.1 Echange de capacités

Pour négocier l'utilisation de certificats de clé publique en vue de l'échange de clé SRTP, le point d'extrémité H.323 doit définir le champ **genericH235SecurityCapability** dans le champ **encryptionAuthenticationAndIntegrity** dans une entrée **h235SecurityCapability** du tableau **capabilityTable** d'un message **TerminalCapabilitySet** H.245 comme suit:

- **capabilityIdentifier** doit contenir l'identificateur d'objet CMS H.235.8 (voir Tableau 4) dans le champ **standard**;
- **maxbitRate, collapsing**, **nonCollapsing,** et **transport** doivent être inutilisés;
- **nonCollapsingRaw** doit contenir le paramètre **SrtpCryptoCapability**.

6.2.2 Echange de clés

Si la clé de session SRTP doit être chiffrée au moyen de clés publiques, la clé de session SRTP chiffrée est alors transportée à l'intérieur de corps de syntaxe CMS dans des messages H.245. Le corps **EnvelopedData** CMS ainsi que le corps **SignedData** CMS doivent être transportés non pas dans le champ **SrtpKeys** mais dans le champ **genericKeyMaterial** du paramètre **secureSharedSecret** (**V3KeySyncMaterial**) figurant dans le conteneur **h235Key** dans le paramètre **encryptionSync** de messages **OpenLogicalChannel** H.245. Le corps **EnvelopedData** CMS doit être placé dans le champ **genericKeyMaterial** suivi immédiatement du corps **SignedData** CMS.

La structure **SrtpKeys** doit être chiffrée au moyen de la clé de chiffrement de contenu (CEK, *content encryption key*) CMS et transportée dans la structure **EncryptedContentInfo** d'un corps **EnvelopedData** CMS.

La présence d'un corps CMS contenant les données de clé de session SRTP dans le conteneur **genericKeyMaterial** doit être identifiée au moyen de la valeur de l'identificateur d'objet CMS H.235.8 (voir Tableau 4) dans le champ **standard** de **capabilityIdentifier** à l'intérieur du champ **genericH235SecurityCapability** de **encryptionAuthenticationAndIntegrity** dans **h235Media** du **dataType** de message OLC.

Tableau 4/H.235.8 – Identificateur d'objet CMS H.235.8

Valeur de l'identificateur d'objet
{ itu-t (0) recommendation (0) h (8) 235 version (0) 4 94 }

6.3 Utilisation de corps CMS

Le point d'extrémité qui produit les données de clé de session SRTP **SrtpKeys**, c'est-à-dire le point d'extrémité émetteur, doit chiffrer ces données au moyen de la clé de chiffrement de contenu CMS, laquelle est elle-même chiffrée par la clé publique de l'autre point d'extrémité, à savoir le point d'extrémité récepteur, et doit placer les données de clé de session SRTP chiffrées dans un corps **EnvelopedData** CMS. Le point d'extrémité émetteur doit ensuite signer numériquement le corps **EnvelopedData** au moyen de sa clé privée et créer un corps **SignedData** CMS "signature séparée". Enfin, il doit inclure le certificat et sa clé publique dans le corps **SignedData** CMS et doit envoyer le corps **EnvelopedData** ainsi que le corps **SignedData** "signature séparée" au point d'extrémité récepteur. La création des corps **EnvelopedData** et **SignedData** par le point d'extrémité émetteur est décrite de façon plus détaillée dans les paragraphes qui suivent.

Les corps **EnvelopedData** et **SignedData** "signature séparée" sont représentés dans la Figure 5.

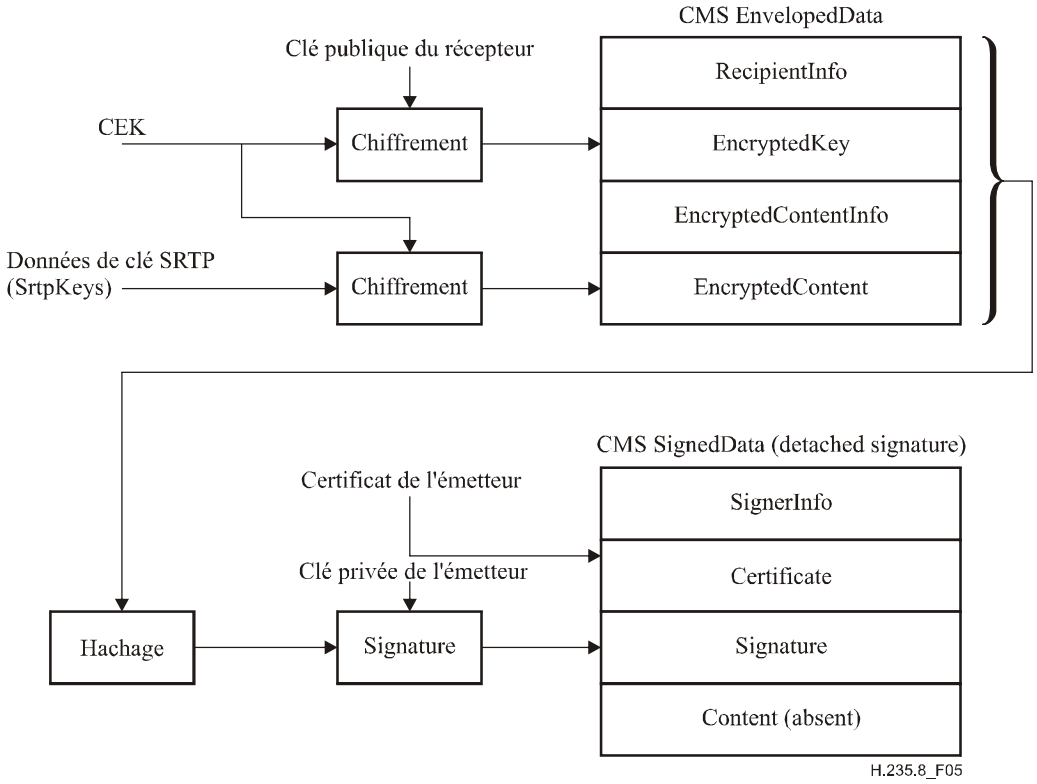

Figure 5/H.235.8 – Corps EnvelopedData et SignedData CMS

6.3.1 Procédures applicables au point d'extrémité émetteur

Le point d'extrémité émetteur doit appliquer les procédures qui suivent afin de produire, chiffrer et signer les données de clé de session SRTP.

6.3.1.1 Corps EnvelopedData

Le point d'extrémité émetteur doit construire le corps **EnvelopedData** comme suit:

1) produire les données de clé de session SRTP **SrtpKeys** pour la suite cryptographique;
2) produire une clé de chiffrement de contenu (CEK, *content encryption key*) aléatoire;
3) chiffrer la clé CEK au moyen de la clé publique du point d'extrémité récepteur. On part du principe que le point d'extrémité émetteur possède déjà la clé publique et le certificat du point d'extrémité récepteur. Placer l'identificateur de l'algorithme utilisé pour le chiffrement de la clé CEK dans le champ **keyEncryptionAlgorithm** de la structure **RecipientInfo.ktri**;
4) placer la clé CEK chiffrée dans le champ **encryptedKey** d'une structure **RecipientInfo** d'un corps **EnvelopedData**. Le champ **rid** de la structure **RecipientInfo.ktri** est employé pour identifier le certificat et la clé publique du point d'extrémité récepteur, laquelle était utilisée pour chiffrer la clé CEK;
5) chiffrer les données de clé SRTP **SrtpKeys** au moyen de la clé CEK et placer l'identificateur de l'algorithme utilisé pour le chiffrement dans le champ **contentEncryptionAlgorithm** de la structure **EncryptedContentInfo**;
6) placer les données de clé SRTP chiffrées dans le champ **encryptedContent** de la structure **EncryptedContentInfo**.

6.3.1.2 Corps SignedData

Le point d'extrémité émetteur doit construire le corps **SignedData** "signature séparée" comme suit:

1) calculer un résumé de message, ou une valeur de hachage, sur le corps **EnvelopedData**. L'identificateur de l'algorithme du résumé de message est placé dans le champ **digestAlgorithm** de la structure **SignerInfo**;

2) signer le résumé de message au moyen de la clé privée du point d'extrémité émetteur et placer la valeur de signature dans le champ **signature** de la structure **SignerInfo**. L'identificateur de l'algorithme de signature est placé dans le champ **signatureAlgorithm** de la structure **SignerInfo**;

3) placer le certificat contenant la clé publique du point d'extrémité émetteur dans la structure **certificates** de la structure **SignerData**. Le champ **sid** de la structure **SignerInfo** doit être défini de manière à identifier le certificat au moyen soit du nom distinctif de l'émetteur et du numéro de série du certificat, soit de la valeur d'extension subjectKeyIdentifier X.509;

4) le champ **eContentType** de la structure **encapContentInfo** dans le corps **SignedData** doit contenir l'identificateur d'objet id-envelopedData. Le champ **eContent** de la structure **encapContentInfo** dans le corps **SignedData** doit être absent étant donné qu'il s'agit d'une signature séparée et que le contenu signé réel est le corps **EnvelopedData**.

6.3.2 Procédures applicables au point d'extrémité récepteur

Le point d'extrémité récepteur doit appliquer les procédures qui suivent afin de vérifier et de déchiffrer les données de clé de session SRTP.

Si le point d'extrémité récepteur ne parvient pas à valider certaines données dans les procédures décrites ci-dessous, l'appel sera rejeté en envoyant un message **ReleaseComplete** avec **ReleaseCompleteReason** mis à **securityDenied**, ou en envoyant un élément **FastConnectRefused** dans un message H.225.0.

6.3.2.1 Corps SignedData

Le point d'extrémité récepteur doit vérifier le corps **SignedData** "signature séparée" reçu comme suit:

1) obtenir le certificat du point d'extrémité émetteur à partir de la structure **certificates** de la structure **SignerData**;

2) valider le certificat du point d'extrémité émetteur. Les détails concernant la validation du trajet du certificat n'entrent pas dans le cadre de la présente Recommandation. S'il n'est pas en mesure d'authentifier le point d'extrémité émetteur, le récepteur peut rejeter l'appel;

3) le point d'extrémité récepteur peut ensuite ajouter le certificat validé à sa mémoire de clé;

4) vérifier la valeur de signature dans le champ **signature** de la structure **SignerInfo** au moyen de la clé publique du point d'extrémité émetteur à partir du certificat validé. Utiliser l'algorithme de signature spécifié dans le champ **signatureAlgorithm** de la structure **SignerInfo**. Le résultat du déchiffrement est le résumé de message calculé sur le corps **EnvelopedData** par le point d'extrémité émetteur;

5) calculer le résumé de message sur le corps **EnvelopedData** reçu en utilisant l'identificateur de l'algorithme du résumé de message spécifié dans le champ **digestAlgorithm** de la structure **SignerInfo**;

6) comparer la valeur du résumé de message déchiffré avec la valeur du résumé de message calculé. Si les deux résumés de message correspondent, le corps **EnvelopedData** peut ensuite être traité. S'ils ne correspondent pas, le point d'extrémité récepteur doit rejeter l'appel.

6.3.2.2 Corps EnvelopedData

Le point d'extrémité récepteur doit extraire les données de clé de session SRTP du corps **EnvelopedData** comme suit:

1) utiliser le champ **rid** de la structure **RecipientInfo** pour identifier le certificat et la clé privée correspondante du point d'extrémité récepteur dans la mémoire de clé de ce dernier. S'il reçoit un corps **EnvelopedData** chiffré au moyen d'une clé publique qui lui est inconnue, le point d'extrémité récepteur doit rejeter l'appel;
2) extraire la clé CEK chiffrée du champ **encryptedKey** d'une structure **RecipientInfo.ktri** d'un corps **EnvelopedData**;
3) déchiffrer la clé CEK chiffrée au moyen de la clé privée du point d'extrémité récepteur ainsi que de l'algorithme spécifié dans le champ **keyEncryptionAlgorithm** de la structure **RecipientInfo.ktri**;
4) extraire les données de clé de session SRTP chiffrées du champ **encryptedContent** de la structure **EncryptedContentInfo**;
5) déchiffrer les données de clé de session SRTP chiffrées au moyen de la clé CEK et de l'algorithme spécifié dans le champ **contentEncryptionAlgorithm** de la structure **EncryptedContentInfo**.

7 Syntaxe relative aux descriptions de sécurité SRTP H.235

La syntaxe ASN.1 est définie ci-après.

```
H235-SRTP DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

IMPORTS
     GenericData
FROM H323-MESSAGES;


 SrtpCryptoCapability ::= SEQUENCE OF SrtpCryptoInfo -- utilisé dans
 genericH235SecurityCapability


 SrtpCryptoInfo ::= SEQUENCE
 {
    cryptoSuite                     OBJECT IDENTIFIER OPTIONAL ,
    sessionParams                   SrtpSessionParameters OPTIONAL,
    allowMKI                        BOOLEAN OPTIONAL,
    ...
 }

 SrtpKeys ::= SEQUENCE OF SrtpKeyParameters  -- utilisé dans V3KeySyncMaterial

 SrtpKeyParameters ::= SEQUENCE
 {
    masterKey                       OCTET STRING,
    masterSalt                      OCTET STRING,
    lifetime                        CHOICE
    {
        powerOfTwo                  INTEGER,
        specific                    INTEGER,
        …
    } OPTIONAL,
    mki                             SEQUENCE
    {
        length                      INTEGER(1..128),
        value                       OCTET STRING,

      ...
    } OPTIONAL,
```

```
    ...
}


SrtpSessionParameters ::= SEQUENCE
{
    kdr                             INTEGER(0..24) OPTIONAL, -- puissance de 2
    unencryptedSrtp                 BOOLEAN OPTIONAL,
    unencryptedSrtcp                BOOLEAN OPTIONAL,
    unauthenticatedSrtp             BOOLEAN OPTIONAL,
    fecOrder                        FecOrder OPTIONAL,
    windowSizeHint                  INTEGER(64..65535) OPTIONAL,
    newParameter                    SEQUENCE OF GenericData  OPTIONAL,
    ...
}

FecOrder ::= SEQUENCE
{
    fecBeforeSrtp                   NULL OPTIONAL,
    fecAfterSrtp                    NULL OPTIONAL,
    ...
}

END
```

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