Groupe de travail Réseau
Request for Comments : 4568
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

F. Andreasen, Cisco Systems
M. Baugher, Cisco Systems
D. Wing, Cisco Systems
juillet 2006

Définition d'attributs de sécurité dans le protocole de description de session (SDP) pour les flux de support

Statut du présent mémoire
Le présent document spécifie un protocole de normalisation Internet pour la communauté Internet, et appelle à discussion et suggestions en vue de son amélioration. Prière de se rapporter à l'édition en cours des "Internet Official Protocol Standards" (normes officielles du protocole Internet) (STD 1) pour connaître l'état de la normalisation et le statut du présent protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le présent document définit un attribut de chiffrement du protocole de description de session (SDP, *Session Description Protocol*) pour les flux de supports en envoi individuel. L'attribut décrit une clé de chiffrement et d'autres paramètres qui servent à configurer la sécurité pour un flux de supports en envoi individuel soit dans un seul message, soit dans un échange en aller-retour. L'attribut peut être utilisé avec divers transports de supports SDP, et le présent document définit comment l'utiliser pour des flux de supports en envoi individuel du protocole de transport sûr en temps réel (SRTP, *Secure Real-time Transport Protocol*). L'attribut SDP "crypto" exige que les services d'un protocole de sécurité des données sécurisent le message SDP.

Table des matières

1. Introduction

Le protocole de description de session (SDP) [RFC4566] décrit des sessions multimédia, qui peuvent être des flux de supports audio, vidéo, tableau d'affichage, télécopie, modem, et autres. Les services de sécurité comme l'authentification de l'origine des données, l'intégrité, et la confidentialité sont souvent nécessaire pour ces flux. Le protocole de transport sûr en temps réel (SRTP, *Secure Real-time Transport Protocol*) [RFC3711] fournit des services de sécurité pour la prise en charge de RTP et est signalé par l'utilisation d'un transport RTP sécurisé (par exemple, "RTP/SAVP" ou "RTP/SAVPF") dans une ligne de support SDP (m=). Cependant, il n'existe aucun moyen dans SDP lui-même pour configurer SRTP au delà de l'utilisation de valeurs par défaut. Le présent document spécifie un nouvel attribut SDP appelé "crypto", qui est utilisé pour signaler et négocier les paramètres de chiffrement pour les flux de supports en général, et pour SRTP en particulier. La définition de l'attribut "crypto" dans le présent document est limitée aux flux de supports en envoi individuel entre deux parties où chaque source a une unique clé de chiffrement ; la prise charge des flux de supports en diffusion groupée ou des flux multipoints en envoi individuel fera l'objet de travaux ultérieurs.

L'attribut "crypto" est défini de façon générique pour permettre son utilisation avec SRTP et tous autres transports sûrs qui peuvent établir des paramètres de chiffrement avec un seul message ou en un seul échange aller-retour utilisant le modèle d'offre/réponse [RFC3264]. Les extensions aux transports autres que SRTP sortent cependant du domaine d'application du présent document. Chaque type de transport de supports sûr doit avoir sa propre spécification pour le paramètre d'attribut "crypto". Ces définitions sont fréquemment uniques au type particulier de transport et doivent être spécifiées dans une RFC sur la voie de la normalisation et être enregistrés par l'IANA selon les procédures définies à la Section 10. Le présent document définit les paramètres de sécurité et le matériel de chiffrement seulement pour SRTP.

Il serait contre productif de ne pas sécuriser les clés de chiffrement et les autres paramètres au moins autant que le sont les données. Les protocoles de sécurité des données comme SRTP s'appuient sur un système séparé de gestion de clés pour établir de façon sûre les clés de chiffrement et/ou d'authentification. Les protocoles de gestion de clé fournissent des procédures d'établissement de clé authentifiées (AKE, *authenticated key establishment*) pour authentifier l'identité de chaque point d'extrémité et protéger contre les attaques par interposition, en réflexion/répétition, capture de connexion, et certaines attaques de déni de service [skeme]. Avec la clé, un protocole AKE tel que MIKEY [RFC3830], GDOI [RFC3547], KINK [RFC4430], IKE [RFC4306], multi parties sécurisées [RFC3851], [RFC2015], ou TLS [RFC2246] dissémine en toute sécurité les informations qui décrivent la clé et les données de session de sécurité. AKE est nécessaire parce que il est n'a rien à voir avec la fourniture d'une clé sur un support où un attaquant peut espionner la clé, altérer la définition de la clé pour la rendre inopérante, ou changer les paramètres de la session de sécurité pour obtenir un accès non autorisé aux informations relatives à la session.

SDP, cependant, n'a pas été conçu pour fournir des services AKE, et les descriptions de sécurité de supports définies dans le présent document n'ajoutent pas de service AKE à SDP. La présente spécification n'est pas un remplacement d'un protocole de gestion de clé ni de transport de messages de gestion de clé dans SDP [RFC4567]. Les descriptions de sécurité SDP définies ici conviennent pour les seuls cas restreints où IPsec, TLS, ou autre protocole d'encapsulation des données de sécurité (par exemple, SIP S/MIME) protègent le message SDP. Le présent document ajoute des descriptions de sécurité aux messages SDP chiffrés et/ou authentifiés par le nouvel attribut SDP "crypto", qui fournit les paramètres de chiffrement d'un flux de supports.

L'attribut "crypto" peut être adapté à tout transport de supports, mais sa définition précise est unique pour un transport particulier.

La Section 2 donne les conventions de notation et est suivie par une déclaration d'applicabilité pour l'attribut "crypto" à la Section 3. La Section 4 introduit l'attribut SDP général "crypto", et la Section 5 définit comment il est utilisé avec et sans le modèle d'offre/réponse. La Section 6 définit les détails de l'attribut "crypto" nécessaires pour SRTP, et la Section 7 définit l'utilisation spécifique de SRTP de l'attribut avec et sans le modèle d'offre/réponse. La Section 8 précise les considérations

de sécurité, et la Section 9 donne la grammaire en ABNF pour l'attribut général "crypto" ainsi que son utilisation spécifique dans SRTP. Les considérations relatives à l'IANA sont à la Section 10.

2. Conventions de notation

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

n^r est l'exponentiation, où n est multiplié r fois par lui-même ; n et r sont entiers. 0..k est une gamme d'entiers de tous les entiers de 0 à k, inclus.

Les termes "transport" et "transport de supports" sont utilisés pour signifier "protocole de transport" comme défini dans la [RFC4566].

Des notes explicatives sont fournies en plusieurs endroits de ce document ; ces notes sont marquées par un retrait.

3. Applicabilité

La [RFC4567] fournit des capacités similaires de distribution des clés de chiffrement et est destinée à être utilisée quand la signalisation doit être confidentielle et/ou protégée en intégrité séparément du matériel de chiffrement.

À l'opposé, la présente spécification porte le matériel de chiffrement au sein du message SDP, et elle est destinée à être utilisée quand le matériel de chiffrement est protégé le long de la signalisation. Les mises en œuvre DOIVENT employer des mécanismes de sécurité qui assurent la confidentialité et l'intégrité du matériel de chiffrement. Quand cette spécification est utilisée dans le contexte de SIP [RFC3261], l'application DEVRAIT employer l'URI SIPS ou S/MIME pour fournir la protection du message SDP et du matériel de chiffrement qu'il contient. L'utilisation de la couche transport ou de la sécurité de la couche IP au lieu de l'URI SIPS ou S/MIME N'EST PAS RECOMMANDÉE car la protection du message SDP et du matériel de chiffrement qu'il contient ne peut pas être assurée par toutes les entités intermédiaires comme les mandataires SIP.

4. Attribut et paramètres SDP "crypto"

Un nouvel attribut SDP de niveau support appelé "crypto" décrit la suite de chiffrement, les paramètres de clé et de session pour la ligne précédente de support d'envoi individuel. L'attribut "crypto" DOIT seulement apparaître au niveau support SDP (et non au niveau session). L'attribut "crypto" suit le format :

a=crypto:<tag> <crypto-suite> <key-params> [<session-params>]
(voir au paragraphe 9.1 la grammaire ABNF formelle)

Les champs "tag", "crypto-suite", "key-params", et "session-params" sont décrits dans les paragraphes qui suivent. Les valeurs de chacun de ces champs sont insensibles à la casse, sauf notation contraire. Cependant, les mises en œuvre sont invitées à utiliser la casse réelle montrée dans le présent document et ses extensions. Noter que selon les règles normales de SDP, le nom d'attribut "crypto" lui-même est sensible à la casse. On donne ci-dessous un exemple de l'attribut "crypto" pour le transport "RTP/SAVP", c'est-à-dire, l'extension RTP sécurisé au profil audio/vidéo [RFC3711]. Dans ce qui suit, les nouvelles lignes sont incluse pour les seules raisons de formatage :

```
a=crypto:1 AES_CM_128_HMAC_SHA1_80
    inline:PS1uQCVeeCFCanVmcjkpPywjNWhcYD0mXXtxaVBR|2^20|1:32
```

La crypto-suite est AES_CM_128_HMAC_SHA1_80, key-params est défini par le texte commençant par "inline:", et session-params est omis.

4.1 Étiquette

L'étiquette *(tag)* est un nombre décimal utilisé comme identifiant pour un attribut "crypto" particulier (voir les détails au paragraphe 9.1) ; les zéros en tête NE DOIVENT PAS être utilisés. L'étiquette DOIT être unique parmi tous les attributs "crypto" pour une ligne de supports donnée. Elle est utilisée avec le modèle d'offre/réponse pour déterminer lequel des divers attributs "crypto" offerts a été choisi par celui qui répond (voir au paragraphe 5.1).

Dans le modèle offre/réponse, l'étiquette est un paramètre négocié.

4.2 Suite de chiffrement

Le champ crypto-suite est un identifiant qui décrit les algorithmes de chiffrement et d'authentification (par exemple, AES_CM_128_HMAC_SHA1_80) pour le transport en question (voir les détails au paragraphe 9.1). Les valeurs possibles pour le paramètre crypto-suite sont définis dans le contexte du transport, c'est-à-dire que chaque transport définit un espace de noms séparé pour l'ensemble des crypto-suites. Par exemple, la crypto-suite "AES_CM_128_HMAC_SHA1_80" définie dans le contexte de transport "RTP/SAVP" s'applique seulement à SRTP ; la chaîne peut être réutilisée pour un autre transport (par exemple, "RTP/SAVPF" [RFC4585]) mais une définition distincte va être nécessaire.

Dans le modèle d'offre/réponse, la crypto-suite est un paramètre négocié.

4.3 Paramètres de clé

Le champ key-params donne un ou plusieurs ensembles de matériel de chiffrement pour la crypto-suite en question. Le champ consiste en un indicateur de méthode suivi par deux points, et les informations de chiffrement réelles comme montré ci dessous :

key-params = <key-method> ":" <key-info>
(voir au paragraphe 9.1 la grammaire ABNF formelle)

Le matériel de chiffrement peut être fourni par différents moyens à partir de cela pour key-params ; cependant ceci sort de notre domaine d'application. Une seule méthode est définie dans le présent document, à savoir "inline", qui indique que le matériel de chiffrement réel est fourni dans le champ key-info lui-même. Il y a un seul espace de noms pour la méthode de clé *(key-method)*, c'est-à-dire, key-method est indépendant du transport. De nouvelles méthodes de clé (par exemple, l'utilisation d'un URL) pourront être définies à l'avenir dans une RFC sur la voie de la normalisation. Bien que la méthode de clé elle-même puisse être générique, la définition des informations de clé qui l'accompagnent est spécifique non seulement de la méthode de clé, mais aussi du transport en question. Key-info code le matériel de chiffrement pour une suite de chiffrement, qui définit ce matériel de chiffrement. De nouvelles méthodes de clé DOIVENT être enregistrées par l'IANA selon les procédures du paragraphe 10.2.1.

Key-info est défini comme une chaîne d'octets générale (voir les détails au paragraphe 9.1) ; la syntaxe et la sémantique spécifiques du transport et de la méthode de clé DOIVENT être fournies dans une RFC sur la voie de la normalisation pour chaque combinaison de transport et de méthode de clé qui l'utilise ; les définitions pour SRTP sont fournies à la Section 6. Noter que de telles définitions sont fournies dans le contexte d'un transport particulier (par exemple, "RTP/SAVP") et d'une méthode de clé spécifique (par exemple, "inline"). L'IANA va enregistrer la liste des méthodes de clé prises en charge pour chaque transport.

Quand plusieurs clés sont incluses dans les paramètres de clé, il DOIT être possible de déterminer quelles clés sont utilisées dans un certain paquet de support par une simple inspection du paquet de support reçu; une approche d'épreuve et erreur entre les clés possibles NE DOIT PAS être effectuée.

Pour SRTP, cela pourrait être réalisé par l'utilisation d'identifiant de clé maîtresse (MKI, *Master Key Identifier*) [RFC3711]. L'utilisation des valeurs <"From, "To"> n'est pas prise en charge dans les descriptions de sécurité SRTP pour les raisons expliquées au paragraphe 6.1.

Dans le modèle d'offre/réponse, le paramètre clé est déclaratif.

4.4 Paramètres de session

Les paramètres de session sont spécifiques d'un certain transport et leur utilisation est FACULTATIVE dans le cadre des descriptions de sécurité, où elles sont juste définies comme des chaînes de caractères générales. Si les paramètres de session sont à utiliser pour un certain transport, leur syntaxe et sémantique spécifique du transport DOIVENT être fournies dans une RFC sur la voie de la normalisation ; les définitions pour SRTP sont fournies à la Section 6.

Dans le modèle d'offre/réponse, les paramètres de session peuvent être soit négociés, soit déclaratifs ; la définition de paramètres spécifique de session DOIT indiquer si ils sont négociés ou déclaratifs. Les paramètres négociés s'appliquent aux données envoyées dans les deux directions, tandis que les paramètres déclaratifs ne s'appliquent qu'aux supports

envoyés par l'entité qui a généré le SDP. Donc, un paramètre déclaratif dans une offre s'applique aux supports envoyés par l'offreur, tandis qu'un paramètre déclaratif dans une réponse s'applique aux supports envoyés par celui qui répond.

4.5 Exemple

Cet exemple montre l'utilisation de l'attribut "crypto" pour le type de transport "RTP/SAVP" (comme défini à la Section 5). La ligne "a=crypto" est en fait une longue ligne ; elle est montrée sur deux lignes à cause du formatage de la page.

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=Séminaire SDP
i=Séminaire sur le protocole de description de session
u=http://www.exemple.com/seminars/sdp.pdf
e=j.doe@exemple.com (Jane Doe)
c=IN IP4 161.44.17.12/127
t=2873397496 2873404696
m=video 51372 RTP/SAVP 31
a=crypto:1 AES_CM_128_HMAC_SHA1_80
     inline:d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj|2^20|1:32
m=audio 49170 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_32
     inline:NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32
m=application 32416 udp wb
a=orient:portrait
```

Ce message SDP décrit trois flux de supports, dont deux utilisent le transport "RTP/SAVP". Chacun a un attribut "crypto" pour le transport "RTP/SAVP". Ces descriptions spécifiques de SRTP sont définies à la Section 6.

5. Utilisation générale de l'attribut "crypto"

Dans cette section, on décrit l'utilisation générale de l'attribut "crypto" en dehors de toutes règles spécifiques du transport ou de la méthode.

5.1 Utilisation avec le modèle d'offre/réponse

Les règles générales d'offre/réponse pour l'attribut "crypto" sont en plus des règles spécifiées dans la [RFC3264], qui DOIVENT être suivies, sauf mention contraire. La RFC 3264 définit le fonctionnement pour les flux en envoi individuel et en diffusion groupée ; les paragraphes qui suivent décrivent le fonctionnement seulement pour des flux en envoi individuel entre deux parties, car la prise en charge des flux en diffusion groupée (et des flux multipoints en envoi individuel) est encore à l'étude.

5.1.1 Génération de l'offre initiale – flux en envoi individuel

Quand on génère une offre initiale pour un flux en envoi individuel, un ou plusieurs attributs "crypto" DOIVENT être présents pour chaque flux de supports pour lesquels la sécurité est désirée. Chaque attribut"crypto" pour un certain flux de supports DOIT contenir une étiquette unique.

L'ordre de plusieurs lignes "a=crypto" est significatif : la ligne du chiffrement préféré figure en premier. Chaque attribut "crypto" décrit la suite de chiffrement, la ou les clés, et éventuellement les paramètres de session offerts pour le flux de supports. En général, une suite de chiffrement "très préférée" DEVRAIT être cryptographiquement plus forte qu'une suite de chiffrement "moins préférée".

La suite de chiffrement s'applique toujours au support dans les directions prises en charge par le flux de supports (par exemple, envoi et réception). La ou les clés, s'appliquent cependant aux paquets de données (par exemple, paquets SRTP et SRTCP) qui vont être envoyés par la même partie qui a généré le SDP. C'est-à-dire que chaque point d'extrémité détermine ses propres clés de transmission et envoie ces clés, dans SDP, à l'autre point d'extrémité.

Ceci est fait pour la cohérence. Aussi, dans le cas de SRTP, par exemple, RTCP sécurisé va encore s’écouler dans les deux direction d'envoi et de réception pour un flux unidirectionnel.

Le paramètre inline porte le matériel de chiffrement utilisé par un point d'extrémité pour chiffrer le flux de supports transmis par ce point d'extrémité. Le même matériel de chiffrement est utilisé par le receveur pour déchiffrer ces flux.

L'offre peut inclure des paramètres de session. Il n'y a pas de règles générales d'offre pour les paramètres de session ; des règles spécifiques peuvent être fournies à la place au titre des définitions spécifiques du transport de tous les paramètres de session.

Lorsque il produit une offre, l'offreur DOIT être prêt à prendre en charge la sécurité du support en accord avec tous les attributs "crypto" inclus dans l'offre. Il y a cependant deux problèmes associés. Tout d'abord, l'offreur ne sait pas quelle clé celui qui répond va utiliser pour le support envoyé à l'offreur. Ensuite, l'offreur peut n'être pas capable de déduire quels attributs "crypto" offerts ont été acceptés. Comme le support peut arriver avant la réponse, des délais ou une coupure peuvent survenir. Si c'est inacceptable pour l'offreur, l'offreur DEVRAIT utiliser un mécanisme qui sort du domaine d'application du présent document pour empêcher ce problème.

Par exemple, dans SIP [RFC3261], une précondition "sécurité" définie dans la [RFC5027] pourrait résoudre ce problème.

5.1.2 Génération de la réponse initiale – flux en envoi individuel

Quand celui qui répond reçoit l'offre initiale avec un ou plusieurs attributs "crypto" pour un certain flux de supports en envoi individuel, celui qui répond DOIT soit accepter exactement un des attributs "crypto" offerts, soit le flux offert DOIT être rejeté.

Si celui qui répond souhaite indiquer la prise en charge d'autres attributs "crypto", il peut en faire la liste en utilisant les extensions de déclaration simple de capacité SDP [RFC3407].

Seuls les attributs "crypto" qui sont valides peuvent être acceptés ; les attributs valides ne violent aucune des règles générales définies pour les descriptions de sécurité, ni aucune des règles spécifiques définies pour le transport et la méthode de clé en question. Quand il choisit un des attributs "crypto" valides, celui qui répond DEVRAIT choisir l'attribut "crypto" préféré qu'il peut prendre en charge, c'est-à-dire, le premier attribut "crypto" valide pris en charge dans la liste, selon les capacités et politiques de sécurité de celui qui répond.

Si il y a un ou plusieurs attributs "crypto" dans l'offre, mais si aucun d'eux n'est valide ou si aucun des valides n'est pris en charge, le flux de supports offert DOIT être rejeté.

Quand un attribut "crypto" offert est accepté, l'attribut "crypto" dans la réponse DOIT contenir ce qui suit :

* l'étiquette et la suite de chiffrement provenant de l'attribut "crypto" accepté dans l'offre (la même suite de chiffrement DOIT être utilisée dans les directions d'envoi et de réception) ;
* la ou les clés de celui qui répond vont être utilisées pour le support envoyé à l'offreur. Noter qu'une clé DOIT être fournie, sans considération des attributs de direction dans l'offre ou la réponse.

De plus, tous les paramètres de session qui sont négociés DOIVENT être inclus dans la réponse. Les paramètres déclaratifs de session fournis par l'offreur ne sont pas inclus dans la réponse ; cependant, celui qui répond peut fournir son propre ensemble de paramètres déclaratifs de session.

Une fois que celui qui répond a accepté un des attributs "crypto" offerts, celui qui répond PEUT commencer à envoyer des supports à l'offreur en accord avec l'attribut "crypto" choisi. Noter cependant que l'offreur peut n'être pas capable de traiter correctement de tels paquets de supports jusqu'à la réception de la réponse.

5.1.3 Traitement de la réponse initiale – flux en envoi individuel

Quand l'offreur reçoit la réponse, il DOIT vérifier qu'une des suites de chiffrement initialement offertes et son étiquette d'accompagnement ont été acceptées et qu'il en est fait écho dans la réponse. Aussi, la réponse DOIT inclure une ou plusieurs clés, qui vont être utilisées pour les supports envoyés de celui qui répond à l'offreur.

Si l'offre contenait des paramètres de session négociés obligatoires (voir au paragraphe 6.3.7) l'offreur DOIT vérifier que lesdits paramètres sont inclus dans la réponse et qu'il les prend en charge. Si la réponse contient des paramètres de session déclaratifs obligatoires, l'offreur DOIT être capable de les prendre en charge.

Si une des conditions ci-dessus échoue, la négociation DOIT échouer.

5.1.4 Modification de la session

Une fois qu'un flux de supports a été établi, il PEUT être modifié à tout moment, comme décrit à la Section 8 de la [RFC3264]. Une telle modification PEUT être déclenchée par le service de sécurité, par exemple, afin d'effectuer un changement de clé ou de suite de chiffrement. Si la sécurité du flux de supports utilisant les descriptions de sécurité générales définies ici est désirée, l'attribut "crypto" DOIT être inclus dans ces nouveaux échanges d'offre/réponse. Les procédures sont similaires à celles définies aux paragraphes 5.1.1, 5.1.2, et 5.1.3 du présent document, sous réserve des considérations fournies à la Section 8 de la [RFC3264].

5.2 Utilisation en dehors de l'offre/réponse

L'attribut "crypto" peut aussi être utilisé en dehors du contexte de l'offre/réponse lorsque il n'y a pas de négociation de la suite de chiffrement, de la clé de chiffrement, ou des paramètres de session. Dans ce cas, l'envoyeur détermine les paramètres de sécurité pour le flux. Comme il n'y a pas de mécanisme de négociation, l'envoyeur DOIT inclure exactement un attribut "crypto", et le receveur DOIT soit l'accepter, soit NE DEVRAIT PAS recevoir le flux associé. L'envoyeur DEVRAIT choisir la description de sécurité qu'il estime la plus sûre pour ses projets.

5.3 Considérations générales de rétro compatibilité

Dans le modèle d'offre/réponse, il est possible que celui qui répond prenne en charge un certain transport sûr (par exemple, "RTP/SAVP") et accepte le flux de supports offert, mais qu'il ne prenne pas en charge l'attribut "crypto" défini dans le présent document et donc l'ignore. L'offreur peut reconnaître cette situation en voyant un flux de supports accepté dans la réponse qui n'inclut pas de ligne "crypto". Dans ce cas, la négociation de sécurité définie ici DOIT échouer.

Des problèmes similaires existent quand les descriptions de sécurité sont utilisées en dehors du modèle d'offre/réponse. Mais la source d'une description de sécurité non négociée n'a pas l'indication que le receveur a ignoré l'attribut "crypto".

6. Descriptions de sécurité SRTP

Dans cette section, on donne les définitions pour les descriptions de sécurité pour les flux de supports SRTP. Dans la section suivante, on définit comment utiliser les descriptions de sécurité SRTP avec et sans le modèle d'offre/réponse.

Les descriptions de sécurité SRTP DOIVENT n'être utilisées qu'avec le transport SRTP (par exemple, "RTP/SAVP" ou "RTP/SAVPF"). On spécifie ci-dessous les descriptions de sécurité pour le profil "RTP/SAVP", défini dans la [RFC3711]. Cependant, il est prévu que d'autres profils RTP sûrs (par exemple, "RTP/SAVPF") puissent utiliser les mêmes descriptions, qui sont en accord avec la spécification du protocole SRTP [RFC3711].

Il n'est pas garanti qu'un point d'extrémité soit capable de configurer son service SRTP avec un paramètre particulier d'attribut "crypto", mais SRTP garantit une interopérabilité minimale parmi les points d'extrémité SRTP grâce aux paramètres SRTP par défaut [RFC3711]. Les points d'extrémité SRTP qui ont des capacités plus importantes prennent en charge diverses valeurs de paramètre au delà des paramètres SRTP par défaut, et ces valeurs peuvent être configurées par les descriptions de sécurité SRTP définies ici. Un point d'extrémité qui ne prend pas en charge l'attribut "crypto" va l'ignorer en accord avec le SDP. Un tel point d'extrémité ne va pas traiter correctement ce flux de supports. En utilisant le modèle d'offre/réponse, l'offreur et celui qui répond peuvent négocier les paramètres "crypto" pour qu'ils soient utilisés avant le commencement de la session multimédia (voir au paragraphe 7.1).

Plus de vingt paramètres de chiffrement figurent dans la spécification SRTP. Beaucoup de ces paramètres ont des valeurs fixées pour des transformations cryptographiques particulières. Au moment de l'établissement de la session, cependant, il n'est généralement pas nécessaire de fournir un réglage unique pour beaucoup des paramètres SRTP, comme la longueur du sel et les fonctions pseudo aléatoires (PRF, *pseudo-random function*). Donc, il est possible de simplifier la liste des paramètres en définissant des "suites de chiffrement" qui fixent un ensemble de valeurs de paramètre SRTP pour la session de sécurité. Cette approche est suivie par les descriptions de sécurité SRTP, qui utilisent les paramètres généraux de description de sécurité comme suit :

*suite de chiffrement : identifie les transformations de chiffrement et d'authentification.
* paramètre de clé : matériel de chiffrement et paramètres SRTP.
* paramètres de session : les paramètres suivants sont définis :
  - KDR *(Key Derivation Rate)* : le taux de déduction de clé SRTP est le taux d'application d'une fonction pseudo aléatoire à une clé maîtresse.
  - UNENCRYPTED_SRTP : les messages SRTP ne sont pas chiffrés.

- UNENCRYPTED_SRTCP : les messages SRTCP ne sont pas chiffrés.
- UNAUTHENTICATED_SRTP : les messages SRTP ne sont pas authentifiés.
- FEC_ORDER : ordre de correction d'erreur directe (FEC, *forward error correction*) relative aux services SRTP.
- FEC_KEY : clé maîtresse pour la FEC quand le flux de FEC est envoyé à une adresse et/ou accès distinct.
- WSH *(Window Size Hint)* : conseil de taille de fenêtre.
- Extensions : des paramètres d'extension peuvent être définis.

Se référer à la spécification SRTP pour une liste complète des paramètres et leur description au paragraphe 8.2 de la [RFC3711]. Sans considération du paramètre UNENCRYPTED_SRTCP, les offreurs et ceux qui répondent aux descriptions SDP de sécurité NE DOIVENT PAS utiliser le bit E SRTCP pour outrepasser UNENCRYPTED_SRTCP ou la valeur par défaut, qui est pour chiffrer tous les messages SRTCP (voir au paragraphe 6.3.2). Le paramètre de clé, la suite de chiffrement, et les paramètres de session montrés ci-dessus sont décrits en détails dans les paragraphes qui suivent.

6.1 Paramètre de clé SRTP

Les descriptions de sécurité SRTP définissent l'utilisation de la méthode de clé "inline" comme décrit dans ce qui suit. L'utilisation de toute autre méthode de chiffrement (par exemple, URL) pour les descriptions de sécurité SRTP est pour étude ultérieure.

Le type de clé "inline" contient le matériel de chiffrement (clé maîtresse et sel) et toute politique relative à cette clé maîtresse, incluant la durée pendant laquelle elle peut être utilisée (durée de vie) et si il utilise un identifiant de clé maîtresse (MKI) pour associer un paquet SRTP entrant à une clé maîtresse particulière. Les mises en œuvre conformes obéissent aux politiques associées à une clé maîtresse et NE DOIVENT PAS accepter les paquets entrants qui violent la politique (par exemple, après l'expiration de la durée de vie de la clé maîtresse).

Le paramètre de clé contient une ou plusieurs clés maîtresses de chiffrement, dont chacune DOIT être une valeur unique cryptographiquement aléatoire [RFC1750] par rapport aux autres clés maîtresses dans le message SDP entier (c'est-à-dire, incluant les clés maîtresses pour les autres flux). Chaque clé suit le format :

"inline:" <clé||sel> ["|" durée de vie] ["|" MKI ":" longueur]

clé||sel : clé maîtresse et sel enchaînés, codés en base64 (voir la Section 3 de la [RFC3548])
durée de vie : durée de vie de la clé maîtresse (nombre maximum de paquets SRTP ou SRTCP utilisant cette clé maîtresse)
MKI:longueur : MKI et longueur du champ MKI dans les paquets SRTP

La définition suivante donne un exemple pour AES_CM_128_HMAC_SHA1_80 :

inline:d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj|2^20|1:4

Le premier champ ("d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj") du paramètre est la clé maîtresse de chiffrement ajoutée au sel maître ; les deux sont d'abord enchaînés puis codés en base64. La longueur du sel et de la clé enchaînés est déterminée par la crypto-suite pour laquelle la clé s'applique. Si la longueur (après décodage du base64) ne correspond pas à celle spécifiée pour la crypto-suite, l'attribut "crypto" en question DOIT être considéré comme invalide. Chaque clé maîtresse et sel DOIT être un nombre cryptographiquement aléatoire et DOIT être unique pour le message SDP entier. Quand la clé et le sel sont décodés du base64, les caractères de bourrage (c'est-à-dire, une ou deux "=" à la fin des données codées en base64) sont éliminés (voir les détails dans la [RFC3548]). Le codage base64 suppose que l'entrée de codage base64 est un nombre entier d'octets. Si une certaine crypto-suite exige l'utilisation d'une clé et d'un sel enchaînés avec une longueur qui n'est pas un nombre entier d'octets, ladite crypto-suite DOIT définir un schéma de bourrage résultant en une entrée de base64 qui est un nombre entier d'octets. Par exemple, si la longueur définie était 250 bits, six bits de bourrage vont alors être nécessaires, ce qui pourrait être défini comme étant les six derniers bits d'une entrée de 256 bits.

Le second champ est la durée de vie FACULTATIVE de la clé maîtresse mesurée comme un nombre maximum de paquets SRTP ou SRTCP en utilisant cette clé maîtresse (c'est-à-dire, le nombre de paquets SRTP et le nombre de paquets SRTCP doivent chacun être inférieurs à la durée de vie). La valeur de durée de vie PEUT être écrite comme un entier décimal positif (non zéro) ou comme une puissance de deux (voir les détails dans la grammaire au paragraphe 9.2) ; les zéros en tête NE DOIVENT PAS être utilisés. La valeur "durée de vie" NE DOIT PAS excéder la durée de vie maximum de paquet pour la crypto-suite. Si la durée de vie est trop grande ou par ailleurs invalide, l'attribut "crypto" entier DOIT alors être considéré comme invalide. La valeur par défaut PEUT être implicitement signalée en omettant la durée de vie (noter que le champ durée de vie n'inclut jamais un caractère deux-points, tandis que le troisième champ en comporte toujours un). Ceci est pratique quand la durée de vie de la clé de chiffrement SRTP est la valeur par défaut. Comme raccourci pour éviter d'avoir de longues valeurs décimales, la syntaxe de la durée de vie permet d'utiliser le littéral "2^", qui indique "deux à la puissance de". L'exemple ci-dessus montre un cas où la durée de vie est spécifiée comme 2^20. L'exemple suivant, qui est pour la

suite de chiffrement AES_CM_128_HMAC_SHA1_80, a une valeur par défaut pour le champ durée de vie, ce qui signifie que les valeurs par défaut de SRTP et SRTCP vont être utilisées (voir la [RFC3711]) :

inline:YUJDZGVmZ2hpSktMbW9QUXJzVHVWd3l6MTIzNDU2|1066:4

L'exemple montre une clé de 30 octets et le sel enchaîné qui est codé en base64 : l'enchaînement de 30 octets de la clé et du sel est étendu à 40 caractères (octets) par le codage trois en quatre de base64.

Le troisième champ, qui est aussi FACULTATIF, est l'identifiant de clé maîtresse (MKI) et sa longueur en octets.

"MKI" est l'identifiant de clé maîtresse associé à la clé maîtresse SRTP. Le MKI est défini ici comme un entier décimal positif qui est codé comme un entier gros boutien dans les paquets SRTP réels ; les zéros en tête NE DOIVENT PAS être utilisés dans la représentation d'entier. Si le MKI est donné, alors la longueur du MKI DOIT aussi être donnée et séparée du MKI par un caractère deux-points (":"). La longueur du MKI est la taille du champ MKI dans le paquet SRTP, spécifiée en octets comme un entier décimal ; les zéros en tête NE DOIVENT PAS être utilisés. Si la longueur du MKI n'est pas donnée ou si sa valeur excède 128 (octets) l'attribut "crypto" entier DOIT alors être considéré comme invalide. La sous chaîne "1:4" dans le premier exemple alloue à la clé un identifiant de clé maîtresse de 1 qui est long de 4 octets, et le second exemple alloue un identifiant de clé maîtresse de 4 octets de 1066 à la clé. Une ou plusieurs clés maîtresses avec leur MKI associé peuvent être initialement définies, et ensuite mises à jour, ou supprimées et de nouvelles sont définies.

SRTP offre une seconde caractéristique pour spécifier la durée de vie d'une clé maîtresse avec deux valeurs, appelées "From" et "To," qui sont définies dans l'espace de numéros de séquence de SRTP [RFC3711]. La présente spécification de descriptions de sécurité SRTP ne prend cependant pas en charge la caractéristique <"From", "To"> car la durée de vie d'une clé maîtresse AES est de 2^48 paquets SRTP, ce qui signifie qu'il n'y a pas de raisons cryptographiques de remplacer une clé maîtresse pour des applications pratiques en point à point. Pour cette raison, il n'est pas besoin de prendre en charge deux moyens pour signaler la mise à jour de clé. Le MKI est choisi plutôt que <"From", "To"> par la présente spécification pour les très peu d'applications qui en ont besoin parce que la caractéristique MKI est plus simple (bien que le MKI ajoute des octets à chaque paquet, tandis que <"From", "To"> ne le fait pas).

Comme mentionné plus haut, le paramètre clé peut contenir une ou plusieurs clés maîtresses. Quand le paramètre clé contient plus d'une clé maîtresse, toutes les clés maîtresses dans ce paramètre clé DOIVENT inclure une valeur de MKI.

Quand on utilise le MKI, la longueur du MKI DOIT être la même pour toutes les clés dans un attribut "crypto" donné.

6.2 Suites de chiffrement

Les suites de chiffrement SRTP définissent les transformations de chiffrement et d'authentification qui vont être utilisées pour les flux de supports SRTP. La spécification SRTP a défini trois suites de chiffrement, qui sont décrites plus en détails dans les paragraphes qui suivent dans le contexte des descriptions de sécurité SRTP. Le tableau ci-dessous donne une vue d'ensemble des suites de chiffrement et de leurs paramètres :

	AES_CM_128_HMAC_SHA1_80	AES_CM_128_HMAC_SHA1_32	F8_128_ HMAC_SHA1_80
Long. clé maîtresse	128 bits	128 bits	128 bits
Longueur du sel maître	112 bits	112 bits	112 bits
Durée de vie SRTP :	2^48 paquets	2^48 paquets	2^48 paquets
Durée de vie SRTCP :	2^31 paquets	2^31 paquets	2^31 paquets
Chiffrement :	Mode compteur AES	Mode compteur AES	Mode F8 AES
Clé de chiffrement	128 bits	128 bits	128 bits
MAC :	HMAC-SHA1	HMAC-SHA1	HMAC-SHA1
Étiquette d'auth. SRTP	80 bits	32 bits	80 bits
Étiquette d'auth. SRTCP	80 bits	80 bits	80 bits
Long. clé d'auth. SRTP	160 bits	160 bits	160 bits
Long. clé d'auth. SRTCP	160 bits	160 bits	160 bits

6.2.1 AES_CM_128_HMAC_SHA1_80

AES_CM_128_HMAC_SHA1_80 est le chiffrement SRTP en mode compteur AES par défaut et HMAC-SHA1 est l'authentification de message avec l'étiquette d'authentification de 80 bits. La longueur de la clé maîtresse est de 128 bits et a une durée de vie par défaut d'un maximum de 2^48 paquets SRTP ou de 2^31 paquets SRTCP, quel que soit celui qui vient en premier (voir la page 39 de la [RFC3711].

SRTP permet 2^48 paquets SRTP ou 2^31 paquets SRTCP, quel que soit celui qui vient en premier. Cependant, il est RECOMMANDÉ qu'une gestion de clés automatisée permette un changement de clés facile et efficace à des intervalles bien plus petits que 2^31 paquets étant donnés les débits de supports d'aujourd'hui ou même les débits de supports de TVHD.

La longueur des clés de chiffrement SRTP et SRTCP est de 128 bits. La longueur de la clé d'authentification SRTP et SRTCP est de 160 bits (voir les considérations sur la sécurité à la Section 8). La valeur du sel maître est de 112 bits et la valeur du sel de session est de 112 bits. La fonction pseudo aléatoire (PRF) est la fonction pseudo aléatoire SRTP par défaut qui utilise AES en mode compteur avec une longueur de clé de 128 bits.

La longueur de la clé décodée de base64 et la valeur du sel pour cette suite de chiffrement DOIVENT être de 30 caractères (c'est-à-dire, 240 bits) ; autrement, l'attribut "crypto" est considéré comme invalide.

6.2.2 AES_CM_128_HMAC_SHA1_32

Cette suite de chiffrement est identique à AES_CM_128_HMAC_SHA1_80 sauf que l'étiquette d'authentification est de 32 bits.

La longueur de la clé décodée de base64 et la valeur du sel pour cette suite de chiffrement DOIVENT être de 30 octets c'est-à-dire, 240 bits ; autrement, l'attribut "crypto" est considéré comme invalide.

6.2.3 F8_128_HMAC_SHA1_80

Cette suite de chiffrement est identique à AES_CM_128_HMAC_SHA1_80 sauf que le chiffrement est F8 [RFC3711].

La longueur de la clé décodée de base64 et la valeur du sel pour cette suite de chiffrement DOIVENT être de 30 octets, c'est-à-dire, 240 bits ; autrement, l'attribut "crypto" est considéré comme invalide.

6.2.4 Ajout de nouvelles définitions de suites de chiffrement

Si de nouvelles transformations sont ajoutées à SRTP, les nouvelles définitions pour ces transformations DEVRAIENT être données pour les descriptions de sécurité SRTP et publiées dans une RFC sur la voie de la normalisation. Les paragraphes 6.2.1 à 6.2.3 illustrent comment définir les valeurs de suites de chiffrement pour des transformations cryptographiques particulières. Toute nouvelle suite de chiffrement DOIT être enregistrée par l'IANA suivant les procédures de la Section 10.

6.3 Paramètres de session

Les descriptions de sécurité SRTP définissent un ensemble de paramètres "session", qui peuvent être FACULTATIVEMENT utilisés pour outrepasser les valeurs de session SRTP par défaut pour les flux SRTP et SRTCP. Ces paramètres configurent une session RTP pour des services SRTP. Les paramètres de session fournissent des informations spécifiques de la session pour établir le contexte cryptographique SRTP.

6.3.1 KDR=n

KDR *(Key Derivation Rate)* spécifie le taux de déduction de clés, comme décrit au paragraphe 4.3.1 de la [RFC3711].

La valeur n DOIT être un entier décimal dans l'ensemble {1,2,...,24} qui note une puissance de 2 de 2^1 à 2^24, inclus ; les zéros en tête NE DOIVENT PAS être utilisés. Le taux de déduction de clé SRTP contrôle la fréquence de déduction d'une nouvelle clé de session à partir d'une clé maîtresse SRTP [RFC3711] donnée dans la déclaration. Quand le taux de déduction de clé n'est pas spécifié (c'est-à-dire, si le paramètre KDR est omis) une seule déduction de clé initiale est effectuée [RFC3711].

Dans le modèle offre/réponse, KDR est un paramètre déclaratif.

6.3.2 UNENCRYPTED_SRTCP et UNENCRYPTED_SRTP

Les charges utiles de paquets SRTP et SRTCP sont chiffrées par défaut. Les paramètres de session UNENCRYPTED_SRTCP et UNENCRYPTED_SRTP modifient le comportement par défaut des suites de chiffrement avec lesquelles ils sont utilisés :

* UNENCRYPTED_SRTCP signale que les charges utiles de paquet SRTCP ne sont pas chiffrées.
* UNENCRYPTED_SRTP signale que les charges utiles de paquet SRTP ne sont pas chiffrées.

Dans le modèle offre/réponse, ces paramètres sont négociés. Si UNENCRYPTED_SRTCP est signalé pour la session, le bit E SRTCP DOIT alors être à 0 dans tous les messages SRTCP. Si la valeur par défaut est utilisée, tous les messages SRTCP sont chiffrés, et le bit E DOIT être établi (à 1) sur tous les messages SRTCP.

6.3.3 UNAUTHENTICATED_SRTP

Les charges utiles de paquets SRTP et SRTCP sont authentifiées par défaut. Le paramètre de session UNAUTHENTICATED_SRTP signale que les messages SRTP ne sont pas authentifiés. L'utilisation de UNAUTHENTICATED_SRTP N'EST PAS RECOMMANDÉE (voir les Considérations sur la sécurité).

La spécification SRTP exige l'utilisation de l'authentification de message pour SRTCP, mais pas pour SRTP [RFC3711].

Dans le modèle offre/réponse, ce paramètre est négocié.

6.3.4 FEC_ORDER=order

FEC_ORDER signale l'utilisation de la correction d'erreur directe pour les paquets RTP [RFC2733]. Les valeurs de correction d'erreur directe pour "order" sont FEC_SRTP ou SRTP_FEC. FEC_SRTP signale que la FEC est appliquée avant le traitement SRTP par l'envoyeur des supports SRTP et après le traitement SRTP par le receveur des supports SRTP ; FEC_SRTP est la valeur par défaut. SRTP_FEC est le traitement inverse.

Dans le modèle offre/réponse, FEC_ORDER est un paramètre déclaratif.

6.3.5 FEC_KEY=key-params

FEC_KEY signale l'utilisation d'une ou de clés maîtresses séparées pour un flux de correction d'erreur directe (FEC). La ou les clés maîtresses sont spécifiées avec exactement le même format que le paramètre de clé SRTP défini au paragraphe 6.1, et les règles de sémantique sont les mêmes - en particulier, la ou les clés maîtresses DOIVENT être différentes de toutes les autres clés maîtresses dans le SDP. Un FEC_KEY DOIT être spécifié quand le flux de FEC est envoyé à une adresse IP et/ou accès différent de celui du flux de supports auquel il s'applique (c'est-à-dire, la ligne "m=") par exemple, comme décrit au paragraphe 11,1 de la [RFC2733]. Quand un flux de FEC est envoyé aux mêmes adresse IP et accès que le flux de supports auquel il s'applique, un FEC_KEY NE DOIT PAS être spécifié. Si un FEC_KEY est spécifié dans ce dernier cas, l'attribut "crypto" en question DOIT être considéré comme invalide.

Dans le modèle d'offre/réponse, FEC_KEY est un paramètre déclaratif.

6.3.6 Conseil de taille de fenêtre

SRTP définit le paramètre SRTP-WINDOW-SIZE au paragraphe 3.3.2 de la [RFC3711] pour protéger contre les attaques en répétition. La valeur minimum est 64 [RFC3711] ; cependant, cette valeur peut être considérée comme trop faible pour certaines applications (par exemple, la vidéo).

Le paramètre de session Conseil de taille de fenêtre (WSH, *Window Size Hint*) fournit un conseil sur la taille que devrait avoir cette fenêtre pour fonctionner de façon satisfaisante (par exemple, sur la base de la connaissance par l'envoyeur du nombre de paquets par seconde). Cependant, il peut y avoir assez d'informations données dans des attributs SRTP comme "a=maxprate" [RFC3890] et les modificateurs de bande passante pour permettre au receveur pour déduire le paramètre de façon satisfaisante. Par conséquent, cette valeur est seulement considérée comme un conseil au receveur du SDP qui PEUT choisir d'ignorer la valeur fournie. La valeur est un entier décimal ; les zéros en tête NE DOIVENT PAS être utilisés.

Dans le modèle offre/réponse, WSH est un paramètre déclaratif.

6.3.7 Définition de nouveaux paramètres de session SRTP

De nouveaux paramètres de session SRTP pour les descriptions de sécurité SRTP peuvent être définis dans une RFC sur la voie de la normalisation et enregistrés par l'IANA en accord avec les procédures d'enregistrement définies Section 10.

Les nouveaux paramètres de session SRTP sont obligatoires par défaut. Un paramètre de session SRTP nouvellement défini qui est préfixé d'un caractère tiret ("-") est cependant considéré comme facultatif et PEUT être ignoré. Si un attribut "crypto" SDP est reçu avec un paramètre de session inconnu qui n'est pas préfixé d'un caractère "-", cet attribut "crypto" DOIT être considéré comme invalide.

6.4 Initialisation de contexte de chiffrement SRTP

En plus des divers paramètres SRTP définis ci-dessus, il y a trois éléments d'information qui sont critiques pour le fonctionnement des chiffrements SRTP par défaut :
* SSRC *(Synchronization SouRCe)* : source de synchronisation
* ROC *(Roll-Over Counter)* : compteur de retour à zéro pour une certaine SSRC
* SEQ *(SEQuence number)* : numéro de séquence pour une certaine SSRC

Dan une session en envoi individuel, comme défini ici, il y a trois contraintes sur ces valeurs.

La première contrainte est sur la SSRC, qui rend un flux de clés SRTP unique parmi les autres participants. Comme expliqué dans SRTP, le flux de clés NE DOIT PAS être réutilisé sur deux pièces de texte source différentes ou plus. La réutilisation de flux de clés rend le texte chiffré vulnérable à la cryptanalyse. Une vulnérabilité est que les champs de texte source connus dans un flux peuvent exposer des portions du flux de clés réutilisées, et cela pourrait encore exposer plus de texte source dans d'autres flux. Comme toutes les transformations de chiffrement SRTP courantes utilisent des flux de clés, le partage de clés est un problème général [RFC3711]. SRTP atténue ce problème en incluant la SSRC de l'envoyeur dans le flux de clés. Mais SRTP ne résout pas ce problème intégralement parce que le protocole de transport en temps réel a des collisions de SSRC, qui bien que très rares [RFC3550] sont quand même possibles. Durant une collision, deux SSRC ou plus qui partagent la clé maîtresse vont avoir des flux de clés identiques pour les portions qui se chevauchent de l'espace de numéros de séquence RTP. Les descriptions de sécurité SRTP évitent la réutilisation de flux de clés en rendant uniques les clés maîtresses EXIGÉES pour l'envoyeur et le receveur de la description de sécurité. Donc, la première contrainte est satisfaite.

On note aussi qu'il y a un second problème avec les collisions de SSRC : la SSRC est utilisée pour identifier le contexte de chiffrement et par là le chiffrement, la clé, le ROC, etc. pour traiter les paquets entrants. Dans le cas de collisions de SSRC, l'identification du contexte de chiffrement devient ambiguë et le traitement correct de paquet peut ne pas se faire. De plus, si un paquet BYE RTCP est à envoyer pour une SSRC en collision, ce paquet peut aussi devoir être sécurisé. Dans un scénario de point à multipoint (en envoi individuel) ceci peut être problématique pour la même raison, c'est-à-dire, on ne sait pas quels contextes de chiffrement possibles utiliser. Noter que ces problèmes ne sont pas uniquement pour les descriptions SDP de sécurité ; toute utilisation de SRTP doit les considérer.

La seconde contrainte est que le ROC DOIT être zéro au moment où chaque SSRC commence l'envoi de paquets. Donc, il n'y a pas de concept d'un "adhérent tardif" dans les descriptions de sécurité SRTP, qui sont contraintes d'être en envoi individuel et par paire. Le ROC et le SEQ forment un "indice de paquet" dans les transformations SRTP par défaut et le ROC est par conséquent réglé à zéro au commencement de la session, en accord avec le présent document.

La troisième contrainte est que la valeur initiale de SEQ DEVRAIT être choisie dans la gamme de 0 à 2^15-1 ; cela évite une ambiguïté quand des paquets sont perdus au début de la session. Si elle est au début d'une session, une SSRC peut choisir au hasard une valeur élevée de numéro de séquence et mettre le receveur dans une situation ambiguë : si les paquets initiaux sont perdus dans le transit jusqu'au point où le numéro de séquence revient à zéro (c'est-à-dire, dépasse 2^16-1) le receveur pourrait ne pas reconnaître que son ROC doit être incrémenté. En restreignant le SEQ initial à la gamme de 0 à 2^15-1, la détermination de l'indice de paquet SRTP va trouver la valeur de ROC correcte, sauf si tous les 2^15 premiers paquets sont perdus (ce qui semble, sinon impossible, plutôt improbable). Voir au paragraphe 3.3.1 de la spécification SRTP concernant la détermination de l'indice de paquet [RFC3711].

6.4.1 Lien tardif d'une ou plusieurs SSRC à un contexte de chiffrement

L'indice de paquet dépend donc de la SSRC, de la SEQ d'un paquet entrant, et du ROC, qui est une variable du contexte de chiffrement SRTP. Donc, SRTP a une forte dépendance de sa sécurité à l'unicité de la SSRC.

Étant données les contraintes ci-dessus, les contextes de chiffrement SRTP en envoi individuel peuvent être établis sans qu'il soit besoin de négocier des valeurs de SSRC dans les descriptions de sécurité SRTP. La présente spécification RECOMMANDE plutôt une approche appelée "lien tardif". Quand un paquet arrive, la SSRC qui y est contenue peut être liée au contexte de chiffrement au moment du commencement de la session (c'est-à-dire, à l'arrivée du paquet SRTP) plutôt qu'au moment de la signalisation de la session (c'est-à-dire, la réception d'un SDP). Avec l'arrivée du paquet contenant la SSRC, tous les éléments de données nécessaires pour le contexte de chiffrement SRTP sont détenus par le receveur. (Noter que la valeur de ROC est zéro par définition ; si des valeurs différentes de zéro devaient être prises en charge, de la signalisation supplémentaire serait nécessaire.) En d'autres termes, le contexte de chiffrement pour une session RTP sûre utilisant le lien tardif est initialement identifié par le SDP comme <*, adresse, accès> où '*' est une SSRC générique, "adresse" est l'adresse locale de réception provenant de la ligne "c=", et "accès" est l'accès local de réception provenant de

la ligne "m=". Quand le premier paquet arrive avec ssrcX dans son champ SSRC, le contexte de chiffrement <ssrcX, adresse, accès> est instancié sous réserve des contraintes suivantes :

* Les paquets de supports sont authentifiés : l'authentification DOIT réussir ; sinon, le contexte de chiffrement n'est pas instancié.
* Les paquets de supports ne sont pas authentifiés : le contexte de chiffrement est automatiquement instancié.

Noter que l'utilisation du lien tardif quand il n'y a pas d'authentification des paquets de supports SRTP est l'objet de nombreuses attaques contre la sécurité, et que par conséquent elle n'est PAS RECOMMANDÉE (bien sûr, ceci peut être dit pour SRTP non authentifié en général).

Noter que l'utilisation du lien tardif sans authentification va résulter en la création d'un état local par suite de la réception d'un paquet d'une SSRC inconnue. UNAUTHENTICATED_SRTP, donc, N'EST PAS RECOMMANDÉ parce qu'elle invite à de faciles attaques de déni de service. Au contraire, le lien tardif avec authentification ne souffre pas de cette faiblesse.

6.4.2 Partage des contextes cryptographiques entre les sessions ou SSRC

Avec les contraintes et les procédures décrites ci-dessus, il n'est pas nécessaire de signaler explicitement la SSRC, le ROC, et le SEQ pour une session RTP en envoi individuel. De sorte qu'il n'y a pas de paramètres a=crypto pour signaler la SSRC, le ROC, ou le SEQ. Donc, plusieurs SSRC provenant de la même entité vont partager des paramètres a=crypto lorsque le lien tardif est utilisé. Plusieurs SSRC provenant de la même entité apparaissent à cause de plusieurs sources (microphones, caméras, etc.) ou charges utiles RTP exigeant un multiplexage de SSRC au sein de cette même session. SDP permet aussi que plusieurs sessions RTP soient définies dans la même description de support ("m=") ; ces sessions RTP vont aussi partager les paramètres a=crypto. Une application qui utilise a=crypto de cette façon partage en série une clé maîtresse entre les sessions RTP ou les SSRC et DOIT remplacer la clé maîtresse quand le nombre agrégé de paquets parmi toutes les SSRC approche de 2^31 paquets. Les SSRC qui partagent une clé maîtresse DOIVENT être uniques les unes par rapport aux autres.

6.5 Suppression des contextes de chiffrement

Le mécanisme défini ci dessus traite de la question de la création des contextes de chiffrement. Cependant, en pratique, les participants à la session peuvent vouloir supprimer des contextes de chiffrement avant la terminaison de la session. Comme un contexte de chiffrement contient des informations qui ne peuvent pas être automatiquement récupérées (par exemple, le ROC) il est important qu'envoyeur et receveur s'accordent sur quand un contexte de chiffrement peut être supprimé, et peut-être plus important, quand il ne le peut pas.

Même quand le lien tardif est utilisé pour un flux en envoi individuel, le ROC est perdu et ne peut pas être récupéré automatiquement (sauf si il est zéro) une fois le contexte de chiffrement supprimé.

On résout ce problème comme suit. Quand les descriptions de sécurité SRTP sont utilisées, la suppression du contexte de chiffrement DOIT suivre les mêmes règles que la suppression de SSRC du tableau des membres [RFC3550] ; noter que cela peut arriver par suite d'un paquet BYE SRTCP ou d'une simple fin de temporisation pour cause d'inactivité. Les participants inactifs à la session qui souhaitent s'assurer que leurs contextes de chiffrement ne sont pas arrivés en fin de temporisation DOIVENT donc envoyer des paquets SRTCP à des intervalles réguliers.

7. Utilisation spécifique de SRTP de l'attribut "crypto"

La Section 5 décrit l'utilisation générale de l'attribut "crypto", et cette section la complète en décrivant l'utilisation spécifique de SRTP.

7.1 Utilisation avec le modèle d'offre/réponse

Dans ce paragraphe, on décrit comment les descriptions de sécurité SRTP sont utilisées avec le modèle d'offre/réponse pour négocier les capacités cryptographiques et communiquer les clés maîtresses SRTP. Les règles définies ci-dessous complètent les règles générales d'offre/réponse définies au paragraphe 5.1, qui DOIVENT être suivies, sauf spécification contraire. Noter que les règles ci-dessous définissent seulement le fonctionnement en envoi individuel ; la prise en charge des flux en envoi individuel pour la diffusion groupée et multipoint est pour étude ultérieure.

7.1.1 Génération de l'offre initiale – flux en envoi individuel

Quand l'offre initiale est générée, l'offreur DOIT suivre les étapes du paragraphe 5.1.1, ainsi que les étapes suivantes.

Pour chaque ligne de support en envoi individuel (m=) utilisant le transport RTP sûr où l'offreur veut spécifier des paramètres de chiffrement, l'offreur DOIT fournir au moins une description de sécurité SRTP valide (ligne "a=crypto") comme défini à la Section 6. Si le flux de supports inclut la correction d'erreur directe avec une adresse et/ou un accès différents de celui du flux de supports lui-même, un paramètre FEC_KEY DOIT être inclus, comme décrit au paragraphe 6.3.5.

Le paramètre inline porte la clé maîtresse SRTP utilisée par un point d'extrémité pour chiffrer les flux SRTP et SRTCP transmis par ce point d'extrémité. La même clé est utilisée par le receveur pour déchiffrer ces flux. Cependant, le receveur NE DOIT PAS utiliser la même clé pour les paquets SRTP ou SRTCP qu'il envoie à la session parce que le chiffrement et mode SRTP par défaut ne sont pas sûrs quand la clé maîtresse est réutilisée sur des flux SRTP distincts.

L'offreur PEUT inclure un ou plusieurs autres paramètres de session SRTP, comme défini au paragraphe 6.3. Noter cependant que si des paramètres de session SRTP sont inclus qui ne sont pas connus de celui qui répond, mais qui sont néanmoins obligatoires (voir au paragraphe 6.3.6) la négociation va échouer si celui qui répond ne les prend pas en charge.

7.1.2 Génération de la réponse initiale - flux en envoi individuel

Quand la réponse initiale est générée, celui qui répond DOIT suivre les étapes du paragraphe 5.1.2, ainsi que les étapes suivantes.

Pour chaque ligne de support en envoi individuel qui utilise le transport RTP sûr et contient une ou plusieurs lignes "a=crypto" dans l'offre, celui qui répond DOIT soit en accepter une (et seulement une) pour ce flux de supports, soit il DOIT rejeter le flux de supports. Seules les lignes "a=crypto" qui sont considérées être des descriptions de sécurité SRTP valides, comme défini à la Section 6, peuvent être acceptées. De plus, tous les paramètres (suite de chiffrement, paramètre de clé, et paramètres de session obligatoires) DOIVENT être acceptables pour celui qui répond afin que le flux de supports offert soit accepté. Noter que si le flux de supports inclut la correction d'erreur directe avec une adresse IP et/ou un accès différents de ceux du flux de supports lui-même, un paramètre FEC_KEY DOIT être inclus, comme décrit au paragraphe 6.3.5.

Quand celui qui répond accepte un flux de supports SRTP en envoi individuel avec une ligne "crypto", celui qui répond DOIT inclure une ou plusieurs clés maîtresses appropriées pour l'algorithme de chiffrement choisi ; la ou les clés maîtresses incluses dans la réponse DOIVENT être différentes de celles de l'offre.

Quand la ou les clés maîtresses ne sont pas partagées entre l'offreur et celui qui répond, des collisions de SSRC entre l'offreur et celui qui répond ne vont pas conduire à une réutilisation de flux de clés, et donc les collisions de SSRC ne doivent pas nécessairement être empêchées.

Si la correction d'erreur directe est incluse à une adresse et/ou accès IP séparée, la réponse DOIT inclure un paramètre FEC_KEY, comme décrit au paragraphe 6.3.5.

Des paramètres de session déclaratifs peuvent être ajoutés à la réponse comme d'habitude ; cependant, celui qui répond NE DEVRAIT PAS ajouter de paramètre de session obligatoire (voir au paragraphe 6.3.6) qui pourrait être inconnu de l'offreur.

Si celui qui répond ne peut pas trouver de ligne "crypto" valide qu'il prend en charge, ou si sa politique configurée interdit tout paramètre de clé de chiffrement (par exemple, longueur de clé) ou paramètre de session de chiffrement (par exemple, KDR, FEC_ORDER) il DOIT rejeter le flux de supports, sauf si il est capable de négocier avec succès l'utilisation de SRTP par d'autres moyens qui sortent du domaine d'application du présent document (par exemple, l'utilisation de MIKEY [RFC3830]).

7.1.3 Traitement de la réponse initiale - flux en envoi individuel

Quand l'offreur reçoit la réponse, il DOIT effectuer les étapes du paragraphe 5.1.3, ainsi que les étapes suivantes pour chaque flux de supports SRTP qu'il offre avec une ou plusieurs lignes "crypto" dedans.

Si le flux de supports a été accepté et si il contient une ligne "crypto", il DOIT être vérifié que la ligne "crypto" est valide selon les contraintes spécifiées à la Section 6 (incluant toutes contraintes de FEC).

Si l'offreur ne prend pas en charge ou ne veut pas honorer un ou plusieurs des paramètres SRTP dans la réponse, l'offreur DOIT considérer la ligne "crypto" comme invalide.

Si la ligne "crypto" n'est pas valide, ou si la politique configurée de l'offreur interdit tous les paramètres de clé de chiffrement (par exemple, longueur de clé) ou paramètres de session de chiffrement, la négociation de sécurité SRTP DOIT être réputée avoir échoué.

7.1.4 Modification de la session

Quand un flux de supports utilisant les descriptions de sécurité SRTP a été établi et qu'un nouvel échange offre/réponse est effectué, l'offreur et celui qui répond DOIVENT suivre les étapes du paragraphe 5.1.4, ainsi que les étapes suivantes.

Quand on modifie la session, tous les aspects négociés des flux de supports SRTP peuvent être modifiés Par exemple, une nouvelle suite de chiffrement peut être utilisée ou une nouvelle clé maîtresse peut être établie. Comme décrit dans la [RFC3264], quand un nouvel échange offre/réponse est fait, il va y avoir une fenêtre de temps où l'offreur et celui qui répond doivent être prêts à recevoir des supports en accord avec à la fois le vieil et le nouvel échange offre/réponse

Cette exigence s'applique ici aussi ; cependant, on devait noter que :

* Quand l'authentification n'est pas utilisée, il se peut qu'il ne soit possible ni pour l'offreur ni pour celui qui répond de déterminer si un certain paquet est chiffré selon l'ancien ou le nouvel échange d'offre/réponse. La [RFC3264] définit un couple de techniques pour traiter ce problème, par exemple, de changer les types de charge utile utilisés et/ou les adresses de transport. Noter, cependant, qu'un changement des adresses de transport peut avoir un impact sur la qualité de service ainsi que sur la traversée des pare-feu et NAT. Les descriptions de sécurité SRTP utilisent le MKI pour traiter cela (ce qui ajoute quelques octets à chaque paquet SRTP) comme décrit au paragraphe 6.1. Pour plus de détails sur le MKI, se reporter à la [RFC3711].

* Si celui qui répond change sa clé maîtresse, l'offreur ne va pas être capable de traiter les paquets sécurisés via cette clé maîtresse jusqu'à la réception de la réponse. Ceci pourrait être réglé en utilisant une "précondition" de sécurité [RFC5027].

Si l'offreur inclut une adresse IP et/ou un accès qui diffèrent de ceux utilisés précédemment pour un flux de supports (ou flux de FEC) l'offreur DOIT inclure une nouvelle clé maîtresse avec l'offre (et ce faisant, il va créer un nouveau contexte de chiffrement où le ROC est réglé à zéro). De même, si celui qui répond inclut une adresse IP et/ou accès qui diffère de ceux utilisés précédemment pour un flux de supports (ou flux de FEC) celui qui répond DOIT inclure une nouvelle clé maîtresse avec la réponse (et donc créer un nouveau contexte de chiffrement avec le ROC réglé à zéro). La raison en est que quand celui qui répond reçoit une offre ou quand l'offreur reçoit une réponse avec une adresse IP et/ou accès mis à jour, il n'est pas possible de déterminer si l'autre côté a accès aux anciens paramètres de contexte de chiffrement (et en particulier au ROC). Par exemple, si un côté est une passerelle de supports décomposée, ou si un agent d'utilisateur de boucle locale SIP est impliqué, il est possible que le point d'extrémité de supports ait changé et n'ait plus accès à l'ancien contexte de chiffrement. En exigeant toujours une nouvelle clé maîtresse dans ce cas, celui qui répond/offreur va savoir que le ROC est zéro pour cette offre/réponse, et toute contrainte de durée de vie de clé va être aussi trivialement satisfaite. Une autre considération s'applique ici aux relais de supports : si le relais change le point d'extrémité de supports sur un côté de façon transparente pour l'autre côté, le relais ne peut pas fonctionner comme un simple réflecteur de paquet mais va devoir s'engager activement dans le traitement de paquet SRTP et sa transformation (c'est-à-dire, déchiffrement et rechiffrement, etc.).

Finalement, on note que si la nouvelle offre est rejetée, les anciens paramètres de chiffrement restent en place.

7.1.5 Exemple d'offre/réponse

Dans cet exemple, l'offreur prend en charge deux suites de chiffrement (f8 et AES). La ligne a=crypto est en fait une longue ligne, bien qu'elle soit montrée sur deux dans le présent document du fait du format de page. L'exemple f8 montre deux paramètres inline ; comme on l'explique au paragraphe 6.1, il peut y avoir un ou plusieurs paramètres de clé (c'est-à-dire, inline) dans un attribut "crypto". De cette façon, plusieurs clés sont offertes pour prendre en charge la rotation de clé utilisant un identifiant de clé maîtresse (MKI, *Master Key Identifier*).

L'offreur envoie :

```
v=0
o=sam 2890844526 2890842807 IN IP4 10.47.16.5
s=Discussion SRTP
i=Ddiscussion de RTP sécurisé
```

```
u=http://www.exemple.com/seminars/srtp.pdf
e=marge@exemple.com (Marge Simpson)
c=IN IP4 168.2.17.12
t=2873397496 2873404696
m=audio 49170 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_80
     inline:WVNfX19zZW1jdGwgKCkgewkyMjA7fQp9CnVubGVz|2^20|1:4
FEC_ORDER=FEC_SRTP
a=crypto:2 F8_128_HMAC_SHA1_80
     inline:MTIzNDU2Nzg5QUJDREUwMTIzNDU2Nzg5QUJjZGVm|2^20|1:4;
     inline:QUJjZGVmMTIzNDU2Nzg5QUJDREUwMTIzNDU2Nzg5|2^20|2:4
FEC_ORDER=FEC_SRTP
```

Celui qui répond envoie :

```
v=0
o=jill 25690844 8070842634 IN IP4 10.47.16.5
s=Discussion SRTP
i=Dscussion de RTP sécurisé
u=http://www.exemple.com/seminars/srtp.pdf
e=homer@exemple.com (Homer Simpson)
c=IN IP4 168.2.17.11
t=2873397526 2873405696
m=audio 32640 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_80
     inline:PS1uQCVeeCFCanVmcjkpPywjNWhcYD0mXXtxaVBR|2^20|1:4
```

Dans ce cas, la session va utiliser la suite de chiffrement AES_CM_128_HMAC_SHA1_80 pour le trafic RTP et RTCP. Si F8_128_HMAC_SHA1_80 était choisi par celui qui répond, il y aurait deux clés inline associées au contexte de chiffrement SRTP. Une clé a une valeur de MKI de 1 et la seconde a un MKI de 2.

7.2 Utilisation spécifique de SRTP en dehors de l'offre/réponse

L'utilisation des descriptions de sécurité SRTP en dehors du modèle d'offre/réponse n'est pas défini.

L'utilisation des descriptions de sécurité SRTP en dehors du modèle offre/réponse pourrait avoir été défini pour des flux de supports en envoi seul ; cependant, ce ne serait pas un moyen pour indiquer la clé à utiliser pour SRTCP par le receveur du dit flux de supports.

7.3 Prise en charge du fourchement de SIP

Comme mentionné plus tôt, les descriptions de sécurité définies ici ne prennent pas en charge les flux de supports en diffusion groupée ou les flux en envoi individuel multipoint. Cependant, dans le protocole SIP, il est possible de recevoir plusieurs réponses à une seule offre due à l'utilisation du fourchement (voir la [RFC3621]). Recevoir plusieurs réponses conduit à quelques problèmes pour les descriptions de sécurité SRTP :

* Différentes personnes qui répondent peuvent choisir des chiffrements, clés, différents, etc.; cependant, il n'y a pas moyen pour l'offreur d'associer un paquet de support entrant particulier à une réponse particulière.

* Deux personnes ou plus qui répondent peuvent prendre la même SSRC, et donc les problèmes de collision de SSRC mentionnés plus haut peuvent survenir.

Comme déclaré précédemment, les cas de point à multipoint sortent du domaine d'application des descriptions SDP de sécurité. Cependant, il y a quand même des moyens de prendre en charge le fourchement de SIP, par exemple, en changeant le scénario de multipoint résultant du fourchement de SIP en plusieurs cas d'envoi individuel entre deux parties. Cela peut être fait comme suit : pour chaque réponse reçue au delà de la réponse initiale, produire une nouvelle offre à cette personne qui répond particulière en utilisant une nouvelle adresse de transport de réception (adresse IP et accès) ; noter que ceci exige la prise en charge de la méthode SIP UPDATE [RFC3311]. Aussi, pour assurer que deux sessions de supports ne sont pas établies par inadvertance avant que le UPDATE soit traité par une d'entre elles, on utilise les préconditions de sécurité [RFC5027].

Finalement, on note que tous les agents d'utilisateurs SIP qui ont reçu l'offre vont connaître la ou les clés proposées par l'offre initiale. Si l'offreur veut s'assurer de la sécurité à l'égard de tous les autres agents d'utilisateurs qui peuvent avoir reçu l'offre, un nouvel échange offre/réponse avec une nouvelle clé doit aussi être effectué avec celui qui répond. Noter que l'offreur ne peut pas déterminer si un seul ou plusieurs agents d'utilisateur SIP ont reçu l'offre, car des mandataires intermédiaires de fourchement peuvent ne transmettre qu'une seule réponse à l'offreur.

La description ci-dessus est destinée à suggérer une façon possible de prendre en charge le fourchement SIP. De nombreux détails manquent, et elle ne devait pas être considérée comme une spécification normative. D'autres approches peuvent aussi être possibles.

7.4 Considérations spécifiques de SRTP pour la rétro compatibilité

Il est possible que celui qui répond prenne en charge le transport SRTP et accepte le flux de supports offert, mais qu'il ne prenne pas en charge l'attribut "crypto" défini ici. L'offreur peut reconnaître cette situation en voyant un flux de supports SRTP accepté dans la réponse qui n'inclut pas de ligne "crypto". Dans ce cas, la négociation de sécurité définie ici DOIT être réputée avoir échoué.

Aussi, si un flux de supports avec un certain transport SRTP (par exemple, "RTP/SAVP") est envoyé à un appareil qui ne prend pas en charge SRTP, ce flux de supports va être rejeté.

7.5 Fonctionnement avec KEYMGT= et k= lines

Une offre PEUT inclure les deux lignes "a=crypto" et "a=keymgt" [RFC4567]. Selon les règles de SDP, celui qui répond va ignorer les lignes d'attribut qu'il ne comprend pas. Si celui qui répond prend en charge à la fois "a=crypto" et "a=keymgt", la réponse DOIT inclure "a=crypto" ou "a=keymgt", mais pas les deux, car l'inclusion des deux est indéfinie.

Une offre PEUT inclure les deux lignes "a=crypto" et "k=" [RFC4566]. Selon les règles de SDP, celui qui répond va ignorer les lignes d'attribut qu'il ne comprend pas. Si celui qui répond prend en charge à la fois "a=crypto" et "k=", la réponse DOIT inclure soit "a=crypto" soit "k=" mais pas les deux, car l'inclusion des deux est indéfinie.

8. Considérations sur la sécurité

Comme tous les messages SDP, ceux qui contiennent des descriptions de sécurité sont convoyés dans un protocole d'application encapsulant (par exemple, SIP, MGCP). Il est de la responsabilité du protocole encapsulant d'assurer la protection des descriptions SDP de sécurité. Donc, il est EXIGÉ que l'application invoque ses propres mécanismes de sécurité (par exemple, des multi parties sécurisées comme S/MIME [RFC3851]) ou autrement, d'utiliser un service de sécurité de couche inférieure (par exemple, TLS ou IPsec). Il est EXIGÉ que ce service de sécurité fournisse une forte authentification de message et le chiffrement de la charge utile du paquet, ainsi qu'une protection efficace contre la répétition.

La "protection contre la répétition" est nécessaire contre un attaquant qui a accès au canal de communications pour intercepter les messages et livrer des copies à la destination. Une attaque en répétition réussie va causer l'exécution par le receveur d'un traitement dupliqué sur un message ; l'attaque est pire quand le receveur dupé envoie une réponse dupliquée à l'initiateur. Les protections contre la répétition ne se trouvent pas dans S/MIME ni dans les autres normes de multi parties sûres, PGP/MIME. S/MIME et PGP/MIME, ont donc besoin d'être augmentés par un mécanisme de protection de l'intégrité approprié au protocole d'application encapsulant (par exemple, SIP, MGCP). Trois façons courantes de fournir la protection contre la répétition sont de placer un numéro de séquence dans le message, d'utiliser un horodatage, et pour le receveur de garder un résumé du message à comparer aux messages entrants. Il est normalement besoin d'avoir une "fenêtre" de répétition et une politique pour garder les informations d'état provenant des précédents messages dans un "tableau de répétition" ou une liste.

La discussion qui suit utilise une "authentification de message" et une "confidentialité de message" d'une manière cohérente avec SRTP [RFC3711]. "Confidentialité de message " signifie que seul le détenteur de la clé secrète de déchiffrement peut accéder au contenu du texte source du message. La clé de déchiffrement est la même clé que la clé de chiffrement, utilisant SRTP en mode compteur et les transformations de chiffrement f8, qui sont vulnérables à l'altération de message et ont besoin de l'authentification de message SRTP pour détecter une telle altération. "Authentification de message" et "validation d'intégrité de message" signifient généralement la même chose dans les normes de sécurité de l'IETF : un message SRTP est authentifié à la suite d'une vérification réussie d'intégrité de HMAC [RFC3711], ce qui prouve que le message a pour origine le détenteur d'une clé maîtresse SRTP et n'a pas été altéré en route. Un tel message "authentique" peut cependant être capturé par un attaquant et "répété" quand l'attaquant réinsère le paquet dans le canal. Un paquet répété

peut avoir divers effets néfastes sur la session, et SRTP utilise le numéro de séquence étendu pour détecter les paquets SRTP répétés [RFC3711].

La spécification SRTP identifie quels services et caractéristiques sont des valeurs par défaut qui sont de mise en œuvre normative (comme AES_CM_128_80) par rapport à d'utilisation normative (comme AES_CM_128_32).

8.1 Authentification des paquets

Les descriptions de sécurité définies ici signalent des services de sécurité pour les paquets RTP. Les messages RTP sont vulnérables à diverses attaques, comme la répétition et la falsification. Pour limiter ces attaques, les mécanismes SRTP de protection de l'intégrité du message DEVRAIENT être utilisés (la protection SRTP contre la répétition est toujours activée).

8.2 Réutilisation de flux de clés

Les descriptions de sécurité SRTP signalent les paramètres de configuration pour les sessions SRTP. Les sessions SRTP mal configurées sont vulnérables aux attaques contre leurs services de chiffrement quand elles fonctionnent avec les suites de chiffrement définies aux paragraphes 6.2.1, 6.2.2, et 6.2.3. Un service de chiffrement SRTP est "mal configuré" quand deux flux de supports ou plus sont chiffrés en utilisant le même flux de clés de blocs AES. Quand les envoyeurs et receveurs partagent des clés de session déduites, SRTP exige que les SSRC des participants à la session servent à rendre leurs flux de clés correspondants uniques, ce qui est violé dans le cas de collision de SSRC : la collision de SSRC SRTP affaiblit de façon drastique le chiffrement de charge utile SRTP ou SRTCP durant le temps où des flux de clés identiques sont utilisés [RFC3711]. Un attaquant, pourrait par exemple collecter des messages SRTP et SRTCP et attendre une collision. Cette attaque contre le chiffrement AES-CM et AES-f8 est évité entièrement quand chaque flux de supports a sa propre clé maîtresse unique dans les deux directions d'envoi et de réception. La présente spécification restreint l'utilisation de la description de sécurité SDP aux flux point à point en envoi individuel afin que les clés ne soient pas partagées entre les hôtes SRTP, et les clés maîtresses utilisées dans les directions d'envoi et de réception pour un certain flux de supports sont uniques.

8.3 Authentification et chiffrement de signalisation

Il n'y a pas cependant de raisons de subir la complexité et les coûts de calcul de SRTP, quand son établissement de clé est exposé à des parties non autorisées. Dans la plupart des cas, l'attribut SRTP "crypto" et ses paramètres sont vulnérables aux attaques de déni de service quand ils sont portés dans un message SDP non authentifié. Dans certains cas, l'intégrité ou la confidentialité du flux RTP peut être compromise. Par exemple, si un attaquant règle UNENCRYPTED pour le flux SRTP dans une offre, il pourrait en résulter que celui qui répond ne déchiffre pas les messages SRTP chiffrés. Dans le pire des cas, celui qui répond peut lui-même envoyer SRTP non chiffré et laisser ses données exposées à l'espionnage.

Donc, il est EXIGÉ que les multi parties MIME sécurisées, IPsec, TLS, ou autres services de sécurité des données soient utilisés pour assurer l'authentification de message pour le protocole encapsulant qui porte des messages SDP qui ont un attribut "crypto" (a=crypto). De plus, il est EXIGÉ que le chiffrement de la charge utile encapsulante soit utilisé chaque fois qu'un paramètre de clé maîtresse (inline) apparaît dans le message. Manquer à chiffrer le message SDP qui contient une clé maîtresse inline SRTP rend inutile le service d'authentification ou de chiffrement SRTP dans pratiquement toutes les circonstances. Manquer à authentifier un message SDP qui porte des paramètres SRTP rend le service d'authentification ou de chiffrement SRTP inutile dans la plupart des applications pratiques.

Quand le chemin de communication du message SDP se fait par des systèmes intermédiaires qui inspectent des parties du message SDP, les protocoles de sécurité tels que IPsec [RFC4301] ou TLS NE DEVRAIT PAS être utilisés pour chiffrer et/ou authentifier la description de sécurité. Dans le cas de système intermédiaire qui traite un message contenant des descriptions SDP de sécurité, les attributs "a=crypto" DEVRAIENT être protégés de bout en bout afin que le système intermédiaire ne puisse ni modifier la description de sécurité ni accéder au matériel de chiffrement. Les protocoles de sécurité réseau ou transport qui se terminent à chaque système intermédiaire, NE DEVRAIENT donc PAS être utilisés pour protéger les descriptions SDP de sécurité. Un protocole de sécurité DEVRAIT permettre que les descriptions de sécurité soient chiffrées et authentifiées de bout en bout indépendamment des portions du message SDP que tout système intermédiaire modifie ou inspecte : les multi parties MIME sécurisées sont RECOMMANDÉES pour la protection des messages SDP qui sont traités par des systèmes intermédiaires.

9. Grammaire

Dans cette section, on fournit d'abord la grammaire ABNF pour l'attribut "crypto" générique, et ensuite la grammaire ABNF pour l'utilisation spécifique de SRTP.

9.1 Grammaire générique de l'attribut "crypto"

La grammaire ABNF pour l'attribut "crypto" est la suivante :

"a=crypto:" tag 1*WSP crypto-suite 1*WSP key-params *(1*WSP session-param)

tag = 1*9DIGIT
crypto-suite = 1*(ALPHA / DIGIT / "_")
key-params = key-param *(";" key-param)
key-param = key-method ":" key-info
key-method = "inline" / key-method-ext
key-method-ext = 1*(ALPHA / DIGIT / "_")
key-info = 1*(%x21-3A / %x3C-7E) ; caractères visibles (imprimables) sauf deux-points
session-param = 1*(VCHAR) ; caractères visibles (imprimables)

où WSP, ALPHA, DIGIT, et VCHAR sont définis dans la [RFC4234].

9.2 Grammaire de l'attribut "crypto" SRTP

Ce paragraphe donne la grammaire ABNF [RFC4234] pour l'utilisation spécifique de SRTP de l'attribut SDP "crypto" :

crypto-suite = srtp-crypto-suite
key-method = srtp-key-method
key-info = srtp-key-info
session-param = srtp-session-param

srtp-crypto-suite = "AES_CM_128_HMAC_SHA1_32" / "F8_128_HMAC_SHA1_32" /
"AES_CM_128_HMAC_SHA1_80" / srtp-crypto-suite-ext

srtp-key-method = "inline"
srtp-key-info = key-salt ["|" lifetime] ["|" mki]

key-salt = 1*(base64) ; valeurs de clé binaire et de sel enchaînées, puis codées en base64 (section 3 de la RFC3548)

lifetime = ["2^"] 1*(DIGIT) ; voir au paragraphe 6.1 pour "2^"
mki = mki-valeur ":" mki-length
mki-valeur = 1*DIGIT
mki-length = 1*3DIGIT ; dans la gamme de 1 à 128.

srtp-session-param = kdr / "UNENCRYPTED_SRTP" / "UNENCRYPTED_SRTCP" / "UNAUTHENTICATED_SRTP" /
fec-order / fec-key / wsh / srtp-session-extension

kdr= "KDR=" 1*2(DIGIT) ; dans la gamme de 0 à 24, puissance de deux.

fec-order = "FEC_ORDER=" fec-type
fec-type = "FEC_SRTP" / "SRTP_FEC"
fec-key = "FEC_KEY=" key-params

wsh = "WSH=" 2*DIGIT ; la valeur minimum est 64
base64 = ALPHA / DIGIT / "+" / "/" / "="

srtp-crypto-suite-ext = 1*(ALPHA / DIGIT / "_")
srtp-session-extension = ["-"] 1*(VCHAR) ; caractères visibles [RFC4234] ; le premier ne doit pas être tiret ("-")

10. Considérations relatives à l'IANA

10.1 Enregistrement de l'attribut "crypto"

L'IANA a enregistré un nouvel attribut SDP comme suit :

Nom de l'attribut : crypto
Forme longue du nom : description de sécurité d'attribut de chiffrement pour flux de supports
Type d'attribut : niveau support
Soumis au jeu de caractères : non
Objet : descriptions de sécurité
Valeurs appropriées : voir la Section 4

10.2 Nouveaux registres IANA et procédures d'enregistrement

Les paragraphes suivants définissent un nouveau registre de l'IANA avec les sous registres associés à utiliser pour les descriptions SDP de sécurité. L'IANA a créé un registre de descriptions de sécurité SDP comme montré ci-dessous et décrit plus en détails dans les paragraphes qui suivent :

Descriptions de sécurité SDP
- Méthodes de clés (décrit en 10.2.1)
- Transports de flux de supports (décrit en 10.2.2)
  - Transport1 (par exemple, SRTP)
    - Méthodes de clé prises en charge (par exemple, inline)
    - suites de chiffrement
    - paramètres de session
  - Transport2
    : :

10.2.1 Registre et enregistrement de méthode de clé

L'IANA a créé un nouveau sous registre pour les méthodes de clé de description de sécurité SDP. Un enregistrement par l'IANA de méthode de clé DOIT être documenté dans une RFC en accord avec l'action de normalisation [RFC2434] et il DOIT fournir le nom de la méthode de clé en accord avec la grammaire pour key-method-ext définie au paragraphe 9.1.

10.2.2 Registre et enregistrement de transport de flux de supports

L'IANA a créé a nouveau sous registre pour les transports de flux de supports de description de sécurité SDP. Un enregistrement IANA de transport de flux de supports DOIT être documenté dans une RFC selon l'action de normalisation de la [RFC2434] et les procédures définies aux Sections 4 et 5 du présent document. L'enregistrement DOIT fournir le nom du transport et une liste des méthodes de clé prises en charge.

De plus, chaque nouveau registre de transport de flux de supports doit contenir un registre des suites de chiffrement et un registre des paramètres de session, ainsi que les instructions de l'IANA sur la façon de remplir ces registres.

10.3 Enregistrements initiaux

10.3.1 Méthode de clé

La méthode de clé de descriptions de sécurité suivante est enregistrée : inline

10.3.2 Transport de flux de supports SRTP

L'IANA a créé un sous registre SDP de transport de flux de supports de description de sécurité pour "SRTP". La méthode de clé prise en charge est "inline". La référence pour la description de sécurité SDP pour SRTP est le présent document.

10.3.2.1 Registre et enregistrement de suite de chiffrement SRTP

L'IANA a créé a nouveau sous registre pour les suites de chiffrement SRTP sous le transport SRTP des descriptions de sécurité SDP. Un enregistrement par l'IANA de suite de chiffrement SRTP DOIT indiquer le nom de la suite de chiffrement en accord avec la grammaire pour srtp-crypto-suite-ext définie au paragraphe 9.2.

La sémantique de la suite de chiffrement SRTP DOIT être décrite dans une RFC en accord avec l'action de normalisation de la [RFC2434], incluant la sémantique de la méthode de clé "inline" et toute la sémantique particulière des paramètres.

Les suites de chiffrement SRTP suivantes sont enregistrées :
AES_CM_128_HMAC_SHA1_80
AES_CM_128_HMAC_SHA1_32
F8_128_HMAC_SHA1_80

La référence pour ces suites de chiffrement est fournie dans le présent document.

10.3.2.2 Enregistrement de paramètre de session SRTP

L'IANA a créé un nouveau sous registre pour les paramètres de session SRTP sous le transport SRTP des descriptions de sécurité SDP. Un enregistrement par l'IANA de paramètre de session SRTP DOIT indiquer le nom du paramètre de session (srtp-session-extension comme défini au paragraphe 9.2) ; le nom NE DOIT PAS commencer par le caractère tiret ("-").

La sémantique du paramètre DOIT être décrite dans une RFC en accord avec l'action de normalisation de la [RFC2434]. Si des valeurs peuvent être allouées au paramètre, alors le format et les valeurs qui peuvent être allouées DOIVENT être décrits dans la RFC en accord également avec l'action de normalisation. Il DOIT aussi être spécifié si le paramètre est déclaratif ou négocié dans le modèle offre/réponse.

Les paramètres de session SRTP suivants sont enregistrés :
KDR
UNENCRYPTED_SRTP
UNENCRYPTED_SRTCP
UNAUTHENTICATED_SRTP
FEC_ORDER
FEC_KEY
WSH

La référence pour ces paramètres est le présent document.

11. Remerciements

Le présent document a été produit par le groupe de travail MMUSIC de l'IETF et a bénéficié de commentaires de ses participants. Le présent document a aussi bénéficié de discussions avec Elisabetta Cararra, Earl Carter, Per Cederqvist, Bill Foster, Matt Hammer, Cullen Jennings, Paul Kyzivat, David McGrew, Mats Naslund, Dave Oran, Jonathan Rosenberg, Dave Singer, Mike Thomas, Brian Weis, et Magnus Westerlund.

12. Références normatives

[RFC1750] D. Eastlake 3rd et autres, "Recommandations d'aléa pour la sécurité", décembre 1994. *(Info., remplacée par* RFC4086)

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2434] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, octobre 1998. *(Rendue obsolète par la* RFC5226)

[RFC2828] R. Shirey, "Glossaire de la sécurité sur l'Internet", FYI 36, mai 2000. (*Obsolète, voir* RFC4949)

[RFC3264] J. Rosenberg et H. Schulzrinne, "Modèle d'offre/réponse avec le protocole de description de session (SDP)", juin 2002. *(P.S. ; MàJ par* RFC8843)

[RFC3548] S. Josefsson, "Codages de données Base16, Base32, et Base64", juillet 2003. *(Obsolète, voir* 4648) *(Info)*

[RFC3550] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "RTP : un protocole de transport pour les applications en temps réel", STD 64, juillet 2003. *(MàJ par* RFC7164, RFC7160, RFC8083, RFC8108, RFC8860)

[RFC3711] M. Baugher et autres, "Protocole de transport sécurisé en temps réel (SRTP)", mars 2004. *(P.S.)*

[RFC4234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", octobre 2005. *(Remplace* RFC2234, *remplacée par* RFC5234)

[RFC4566] M. Handley, V. Jacobson et C. Perkins, "SDP : Protocole de description de session", juillet 2006. *(P.S. ; remplacée par RFC*8866*)*

12. Références pour information

[Bellovin] Bellovin, S., "Problème Areas pour the IP Security Protocols," dans Proceedings of the Sixth Usenix Unix Security Symposium, pp. 1-16, San Jose, CA, juillet 1996.

[RFC2015] M. Elkins, "Sécurité de MIME avec Pretty Good Privacy (PGP)", octobre 1996. (*MàJ par* RFC3156) *(P.S.)*

[RFC2104] H. Krawczyk, M. Bellare et R. Canetti, "HMAC : Hachage de clés pour l'authentification de message", février 1997.

[RFC2246] T. Dierks et C. Allen, "Protocole TLS version 1.0", janvier 1999. *(P.S. ; MàJ par* RFC7919)

[RFC2733] J. Rosenberg et H. Schulzrinne, "Format de charge utile RTP pour la correction d'erreur directe générique", décembre 1999. (*Obsolète, voir* RFC5109) *(P.S.)*

[RFC2974] M. Handley, C. Perkins, E. Whelan, "Protocole d'annonce de session (SAP)", octobre 2000. *(Expérimentale)*

[RFC3261] J. Rosenberg et autres, "SIP : Protocole d'initialisation de session", juin 2002. (*Mise à jour par* 3265, 3853, 4320, 4916, 5393, 6665, 8217, 8760)

[RFC3311] J. Rosenberg, "Méthode UPDATE du protocole d'initialisation de session (SIP) ", octobre 2002.

[RFC3312] G. Camarillo, éd., "Intégration de la gestion de ressource et du protocole d'initialisation de session (SIP)", octobre 2002. (*MàJ par* RFC4032, RFC5027) *(P.S.)*

[RFC3407] F. Andreasen, "Déclaration simple de capacité du protocole de description de session (SDP)", octobre 2002. *(P.S.)*

[RFC3547] M. Baugher, B. Weis, T. Hardjono et H. Harney, "Le domaine d'interprétation de groupe", juillet 2003. *(Obsolète, voir la* RFC6407)

[RFC3830] J. Arkko et autres, "MIKEY : Gestion de clé multimédia pour l'Internet", août 2004. (*MàJ par* RFC4738) *(P.S.)*

[RFC3851] B. Ramsdell, "Spécification du message d'extensions de messagerie Internet multi-objets/sécurisé (S/MIME) version 3.1", juillet 2004. *(Obsolète, voir* RFC5751)

[RFC3890] M. Westerlund, "Modificateur de bande passante indépendant du transport pour le protocole de description de session (SDP)", septembre 2004. *(P.S.)*

[RFC4301] S. Kent et K. Seo, "Architecture de sécurité pour le protocole Internet", décembre 2005. *(P.S.) (Remplace la* RFC2401)

[RFC4306] C. Kaufman, "Protocole d'échange de clés sur Internet (IKEv2)", décembre 2005. *(Obsolète, voir la* RFC5996)

[RFC4430] S. Sakane et autres, "Négociation de clés Kerberos sur Internet (KINK)", mars 2006. *(P.S.)*

[RFC4567] J. Arkko et autres, "Extensions de gestion de clés pour le protocole de description de session (SDP) et le protocole d'écoulement en temps réel (RTSP)", juillet 2006. *(P.S.)*

[RFC4585] J. Ott et autres, "Profil RTP étendu pour rétroaction fondée sur le protocole de contrôle de transport en temps réel (RTCP) (RTP/AVPF)", juillet 2006. *(P.S., MàJ par* RFC8108)

[RFC5027] F. Andreasen, D. Wing, "Préconditions de sécurité pour les flux de support du protocole de description de session (SDP)", octobre 2007. (*MàJ* RFC3312) *(P.S.)*

[skeme] Krawczyk, H., "SKEME: A Versatile Secure Key Exchange Mechanism pour the Internet", ISOC Secure Networks and Distributed Systems Symposium, San Diego, 1996.

Appendice A – Raisons de la directionnalité du matériel de chiffrement

Les descriptions SDP de sécurité définissent le matériel de chiffrement pour la direction envoyeuse, qui est incluse dans le SDP. Donc, la clé qui est portée dans un message SDP est une clé de déchiffrement pour le receveur de ce message SDP. Ceci est au contraire de la majorité des informations incluses dans SDP, qui décrivent des information pour la direction receveuse (ou de réception et d'envoi). Cette directionnalité inversée des informations génère quelques défis dans l'utilisation du mécanisme dans le modèle offre/réponse et en particulier avec SIP, où les supports précoces et le fourchement exigent une considération particulière (comme décrit au paragraphe 7.3). Il y a cependant de bonnes raisons pour que cela soit fait, qui peuvent être résumées comme suit :

D'abord, il y a la philosophie générale de sécurité de laisser l'entité qui envoie le trafic décider quelle clé utiliser pour le protéger. SRTP utilise le mode compteur, qui est sûr quand les compteurs ne se chevauchent pas entre les envoyeurs qui partagent une clé maîtresse ; la façon la plus sûre d'éviter le chevauchement de compteur est que chaque point d'extrémité génère sa propre clé maîtresse. Ensuite, si les descriptions SDP de sécurité avaient été conçues pour garder la directionnalité normale des informations SDP, il en aurait résulté des problèmes avec la prise en charge des supports précoces et du fourchement SIP : si une offre génère plusieurs réponses et si le matériel de chiffrement était pour la direction de réception, certaines des valeurs de paramètres (par exemple lifetime) devraient être partagées entre tous ceux qui répondent (envoyeurs de supports) ce qui conduirait à une complexité considérable, exigeant éventuellement des changements ou des extensions à SRTP. D'autres problèmes ont aussi été découverts, qu'on décrira plus loin.

Dans les scénarios qui suivent, on analyse ce qui va se passer si les descriptions SDP de sécurité avaient été conçues de telle sorte que le matériel de chiffrement soit celui de réception (plutôt que dans la conception actuelle, où le matériel de chiffrement est celui d'envoi) :

Scénario A : cas de non fourchement

Dans ce scénario,l'offre inclut le matériel de chiffrement receveur, celui qui répond le reçoit et commence à envoyer des paquets de données à l'offreur. Si il y avait un seul attribut crypto dans l'offre, il n'y aurait pas d'ambiguïté sur quelle suite de chiffrement a été utilisée et donc, le paquet entrant pourrait être traité. Cependant, dans le cas où l'offre incluait plusieurs attributs de chiffrement, l'offreur ne saurait pas lequel a été choisi, et donc, si l'offreur a reçu des paquets avant le retour de la réponse, l'offreur sera dans l'incapacité de traiter ces paquets (problème 1). (L'utilisation de MKI a été suggérée comme solution possible, cependant dela implique des frais généraux supplémentaires par paquet.)

Scenario B : cas de fourcheent en série

Dans ce scénario, Alice génère une offre à Bob, qui commence à envoyer des supports (précoces) à Alice (pas de réponse encore retournée). Dans ce scénario, on suppose qu'on ne rencontre pas aussi le scénario A (par exemple, l'offre inclut un seul attribut de chiffrement) et que Bob utilise une valeur de synchronisation de source (SSRC) de 1 pour ses paquets SRTP et SRTCP. Alice a donc un contexte de chiffrement pour SSRC 1, incluant le compteur de retour à zéro (ROC, *Roll Over Counter*) et le numéro de séquence (SEQ, *Sequence Number*) RTP associé. Bob transmet maintenant l'appel à Carol (Bob n'a toujours pas généré de réponse). À ce moment, Bob a la clé d'Alice, ce qui peut parfois être une faiblesse de sécurité. Comme l'échange se poursuit, Carol reçoit l'offre originale, incluant l'attribut de chiffrement offert et commence à envoyer des paquets de supports à Alice. Il se trouveque Carol choisit juste une valeur de SSRC de 1, comme l'a fait Bob. Quand Carol commence à générer des paquets, il y a un potentiel pour que ce que la RFC 3711 appelle un problème de "bourrage double" (problème 2) ainsi qu'un potentiel que le ROC soit hors de synchronisation entre Alice et Carol (problème 3). Noter que comme Bob et Carol utilisent (probablement) des adresses de source de transport différentes, la réutilisation de SSRC ne constitue pas une collision de SSRC (bien que cela puisse toujours être interprété comme tel par Alice). Selon la RFC 3711, comme la clé maîtresse serait partagée entre Bob et Carol dans ce cas, il est RECOMMANDÉ qu'Alice quitte la session à ce moment afin d'éviter le problème du bourrage double. On devait aussi noter que la RFC 3711 recommande de ne pas partager les clés maîtresses SRTP, dont le fourchement peut accidentellement s'introduire quand le matériel de chiffrement est pour la direction receveuse.

Si on examine à nouveau le scénario ci-dessus, mais cette fois avec du matériel de chiffrement dans l'offre (et la réponse) étant le matériel de chiffrement d'envoi (comme spécifié par les descriptions SDP de sécurité) le scénario ressemble plutôt à ceci : Bob choisit à nouveau le SSRC 1, et il va devoir le renvoyer en réponse à Alice, car Alice doit prendre connaissance de la clé d'envoi de Bob. Bob commence aussi à envoyer des supports vers Alice (une coupure peut se produire jusqu'à ce que Alice reçoive la réponse de Bob). Bob transmet à nouveau l'appel à Carol qui commence aussi à envoyer des supports précoces en utilisant SSRC 1. Cependant, Carol doit générer une nouvelle réponse (pour le dialogue entre Alice et Carol) afin que Alice traite les paquets de Carol. À réception de cette réponse, Alice peut initier un nouvel échange offre/réponse (pour déplacer la session à une autre adresse de transport comme décrit au paragraphe 7.3). Dans ce cas, il y a une clé maîtresse par session et un flux de clés unique sans considération de collision ou non des SSRC.

Scénario C : cas de fourchement parallèle

Dans ce scénario, Alice génère une offre (avec le matériel de chiffrement en réception) qui se trouve fourchée en parallèle à Bob et Carol. Bob et Carol commencent tous deux à envoyer des paquets (supports précoces) à Alice. Si Bob et Carol choisissent des SSRC différentes, tout va bien initialement. Cependant, un des paramètres de contexte de chiffrement est la durée de vie de clé maîtresse, et comme Bob et Carol partagent la même clé maîtresse (à leur insu) ils ne savent pas quand ils doivent changer de clés (problème 4). Si ils choisissent la même SSRC, on a à nouveau le problème du bourrage double (problème 2).

En résumé, si le matériel de chiffrement était pour la direction de réception, on aurait les problèmes suivants :

- Problème 1 : l'offreur ne sait pas parmi plusieurs offres de chiffrement laquelle a été choisie par celui qui répond.

- Problème 2 : la réutilisation de SSRC (ou les collisions de SSRC) entre plusieurs répondants (fourchement en série ou en parallèle) peut conduire ou problème du bourrage double.

- Problème 3 : une partie des paramètres de contexte de chiffrement (spécifiquement, le ROC) n'est pas communiqué mais déduit, et si on permet que plusieurs entités utilisent la même SSRC (en séquence) le ROC peut être faux.

- Problème 4 : tous les contextes de chiffrement qui partagent une clé maîtresse doivent tenir un ensemble partagé de compteurs (pour la durée de vie de la clé maîtresse) et si on permet que plusieurs entités sur des plateformes différentes partagent une clé maîtresse, on va avoir besoin d'un mécanisme pour synchroniser ces compteurs.

Le problème 1 pourrait être résolu en utilisant le MKI comme proposé par ailleurs ; cependant, il en résulterait l'utilisation de bande passante supplémentaire pour chaque paquet de support SRTP. Résoudre le problème 2 implique le besoin d'être capable de synchroniser les valeurs de SSRC avec celui qui répond (ou l'abandon de la session quand se produit une réutilisation de SSRC ou une collision de SSRC). Le problème 3 implique d'être capable de synchroniser les valeurs de ROC sur la base de la SSRC (ou l'abandon de la session quand la réutilisation de SSRC se produit). Le problème 4 pourrait être résolu en ayant l'offreur (Alice, c'est-à-dire, l'entité qui reçoit les supports) qui détermine combien de paquets ont réellement été générés par l'ensemble total des envoyeurs à Alice et, donc, être celui qui initie le changement de clés. En cas de perte de paquets, etc. ceci n'est pas à toutes épreuves, mais en pratique cela pourrait probablement être réglé par l'utilisation d'une marge de sécurité raisonnable.

En conclusion, on pourrait s'attendre, du point de vue de l'offre/réponse et de SIP, à ce que le matériel de chiffrement de l'offre (et de la réponse) soit le matériel de chiffrement de réception ; cependant, faire ainsi serait brader la sécurité en faveur de la facilité d'utilisation de SIP, par exemple, les problèmes de bourrage double et de durée ce vie de la clé maîtresse, et violerait la règle de la RFC 3711 sur le partage d'une clé maîtresse SRTP entre les sessions SRTP.

Adresse des auteurs

Flemming Andreasen
Cisco Systems, Inc.
499 Thornall Street, 8th Floor
Edison, New Jersey 08837
USA
mél : fandreas@cisco.com

Mark Baugher
5510 SW Orchid Street
Portland, Oregon 97219
USA
mél : mbaugher@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA
mél : dwing@cisco.com

Déclaration complète de droits de reproduction

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourrait être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à http://www.ietf.org/ipr.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf- ipr@ietf.org.

Remerciement

Le financement de la fonction d'édition des RFC est actuellement fourni par la Internet Society.