Groupe de travail Réseau
Request for Comments : 3711
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

M. Baugher & D. McGrew, Cisco Systems, Inc.
M. Naslund, E. Carrara & K. Norrman
Ericsson Research
mars 2004

Protocole sécurisé de transport en temps réel (SRTP)

Résumé
Le présent document décrit le protocole sécurisé de transport en temps réel (SRTP, *Secure Real-time Transport Protocol*) un profil du protocole de transport en temps réel (RTP, *Real-time Transport Protocol*) qui assure la confidentialité, l'authentification de message, et la protection contre la répétition du trafic RTP et du trafic de contrôle pour RTP, le protocole de contrôle de transport en temps réel (RTCP, *Real-time Transport Control Protocol*).

Table des matières

1. Introduction

Le présent document décrit le protocole de transport sécurisé en temps réel (*Secure Real-Time Transport Protocol*) un profil du protocole de transport en temps réel (RTP, *Real-time Transport Protocol*) qui peut fournir la confidentialité, l'authentification de message, et la protection contre la répétition au trafic RTP et au trafic de contrôle pour RTP, le protocole de contrôle du transport en temps réel (RTCP, *Real-time Transport Control Protocol*) [RFC3550].

SRTP fournit un cadre pour le chiffrement et l'authentification de message des flux RTP et RTCP (Section 3). SRTP définit un ensemble de transformations cryptographiques par défaut (Sections 4 et 5) et il permet d'introduire à l'avenir de nouvelles transformations (Section 6). Avec la gestion de clés appropriée (Sections 7 et 8), SRTP est sûr (Section 9) pour les applications RTP en envoi individuel et en diffusion groupée (Section 11).
SRTP peut réaliser un fort débit et une faible expansion de paquet. Il se révèle une protection convenable pour les environnements hétérogènes (mélange de réseaux filaires et sans fils). Pour obtenir ces caractéristiques, les transformations par défaut sont décrites sur la base d'un chiffrement de flux additionnel pour le chiffrement, d'une fonction fondée sur le hachage de clé pour l'authentification de message, et d'un indice "implicite" pour le séquençage/synchronisation fondé sur le numéro de séquence RTP pour SRTP et d'un numéro d'indice pour RTCP sécurisé (SRTCP).

1.1 Conventions de notation

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans la [RFC2119].

La terminologie se conforme à celle de la [RFC2828] avec les exceptions suivantes. Pour simplifier, on utilise le terme "aléatoire" tout au long du document pour noter les valeurs générées de façon aléatoire ou pseudo-aléatoire. De grandes quantités de bits aléatoires peuvent être difficiles à obtenir, et pour la sécurité de SRTP, le pseudo-aléatoire est suffisant [RFC1750].

Par convention, la représentation adoptée est l'ordre des octets du réseau, c'est-à-dire, le bit (octet) le plus à gauche est celui de plus fort poids. Par OUX on veut dire l'addition au bit près modulo 2 de chaînes binaires, et || note l'enchaînement. En d'autres termes, si C = A || B, les bits de poids fort de C sont les bits de A, et les bits de moindre poids de C sont égaux aux bits de B. Les nombres hexadécimaux sont préfixés par 0x.

Le mot "chiffrement" inclut aussi l'utilisation de l'algorithme NUL (qui en pratique laisse les données en clair).

Avec un léger abus de notation, on utilise les termes "authentification de message" et "étiquette d'authentification" comme dans la pratique courante, bien que dans certaines circonstances, par exemple, les communications de groupe, le service fourni soit en fait seulement la protection de l'intégrité et non l'authentification de l'origine des données.

2. Buts et caractéristiques

Le but de la sécurité pour SRTP est d'assurer :
* la confidentialité des charges utiles RTP et RTCP, et
* l'intégrité de la totalité des paquets RTP et RTCP, ainsi que la protection contre la répétition des paquets.

Ces services de sécurité sont facultatifs et indépendants l'un de l'autre, sauf que la protection d'intégrité de SRTCP est

obligatoire (l'altération malveillante ou par erreur des messages RTCP pourrait autrement interrompre le traitement du flux RTP).

Les autres objectifs fonctionnels du protocole sont :

* un cadre permettant la mise à niveau avec de nouvelles transformations cryptographiques,

* un faible coût en bande passante, c'est-à-dire, un cadre qui préserve l'efficacité de la compression de l'en-tête RTP,

et, assurés par les transformations prédéfinies :

* un faible coût de calcul,

* une petite empreinte (c'est-à-dire, une petite taille du code et de la mémoire de données pour les informations de clés et les listes de répétition),

* une expansion limitée de paquet pour prendre en charge l'objectif d'économie de bande passante,

* l'indépendance à l'égard des couches transport, réseau, et physique sous-jacentes utilisées par RTP, en particulier une forte tolérance à la perte et au réarrangement des paquets.

Ces propriétés assurent que SRTP est un schéma de protection convenable pour RTP/RTCP dans les scénarios filaires et sans fils.

2.1 Caractéristiques

À côté des objectifs directs mentionnés ci-dessus, SRTP fournit certaines caractéristiques supplémentaires. Elles ont été introduites pour alléger le fardeau de la gestion de clés et pour encore augmenter la sécurité. Cela inclut :

* Une seule "clé maîtresse" peut fournir le matériel de clé pour la protection de la confidentialité et de l'intégrité, pour le flux SRTP et pour le flux SRTCP correspondant. Cela est réalisé avec une fonction de déduction de clé (paragraphe 4.3) qui fournit des "clés de session" pour les primitives de sécurité respectives, déduites de façon sécurisée de la clé maîtresse.

* De plus, la déduction de clé peut être configurée pour rafraîchir périodiquement les clés de session, ce qui limite la quantité de texte chiffré produite par une clé fixée, disponible pour une cryptanalyse adverse.

* Les "clés salées" sont utilisées pour protéger contre les attaques de pré calcul et de compromis temps/mémoire [MF00] [BS00].

Des détails sur les raisons de ces caractéristiques figurent à la Section 7.

3. Cadre de SRTP

RTP est le protocole de transport en temps réel [RFC3550]. On définit SRTP comme un profil de RTP. Ce profil est une extension du profil audio/vidéo de RTP [RFC3551]. Sauf lorsque noté explicitement, tous les aspects de ce profil s'appliquent avec l'ajout des dispositifs de sécurité SRTP. Conceptuellement, on considère SRTP comme une mise en œuvre "prise dans la pile" qui réside entre l'application RTP et la couche transport. SRTP intercepte les paquets RTP puis transmet un paquet SRTP équivalent du côté d'envoi, et intercepte les paquets SRTP et passe un paquet RTP équivalent sur le dessus de la pile du côté receveur.

RTCP sécurisé (SRTCP) fournit les mêmes services de sécurité à RTCP que SRTP le fait pour RTP. L'authentification de message SRTCP est OBLIGATOIRE et par là protège les champs RTCP pour garder la trace des adhésions, fournir des retours aux envoyeurs RTP, ou entretenir les compteurs de numéro de séquence de paquets. SRTCP est décrit au paragraphe 3.4.

3.1 RTP sécurisé

Le format d'un paquet SRTP est illustré par la Figure 1.

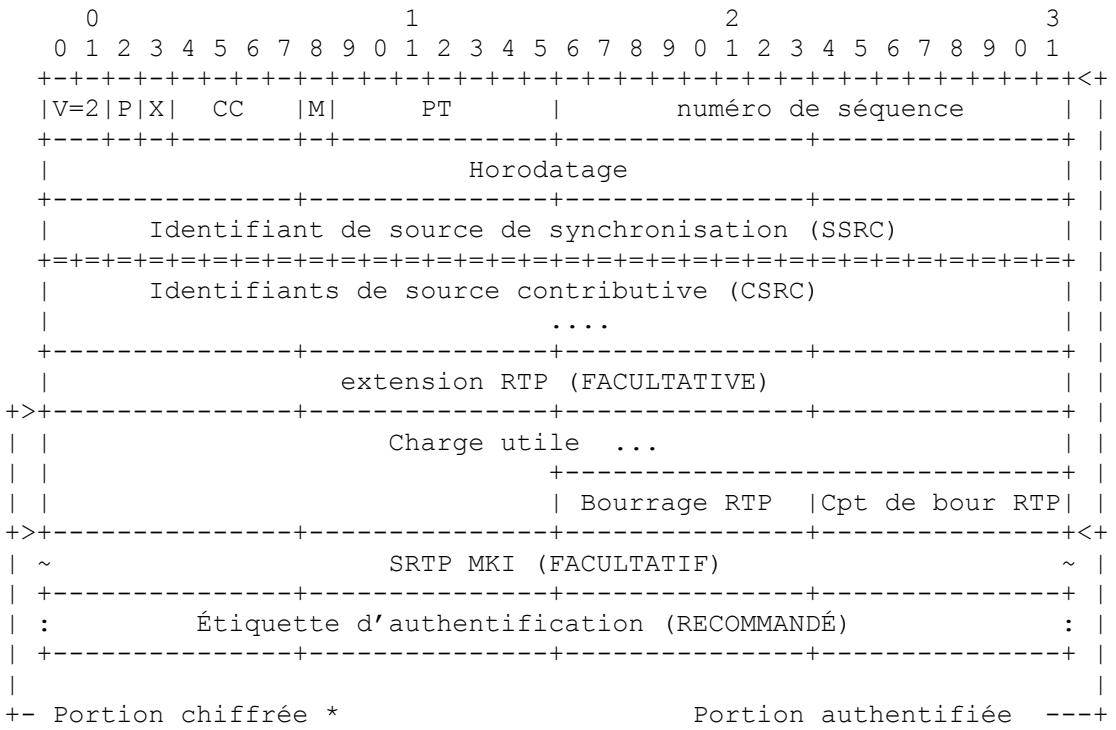


Figure 1. Format d'un paquet SRTP.

* Portion chiffrée a la même taille que le texte en clair pour les transformations prédéfinies de la Section 4.

La "Portion chiffrée" d'un paquet SRTP consiste en chiffrement de la charge utile RTP (y compris le bourrage RTP et le compte de bourrage RTP lorsqu'ils sont présents) du paquet RTP équivalent. La portion chiffrée PEUT être de la taille exacte du texte en clair ou PEUT être plus grande. La Figure 1 montre la charge utile RTP incluant tout bourrage possible pour RTP [RFC3550].

Aucune des transformations de chiffrement prédéfinies n'utilise de bourrage ; pour elles, les tailles de charge utile RTP et SRTP correspondent exactement. Les nouvelles transformations ajoutées à SRTP (suivant la Section 6) peuvent requérir un bourrage, et peuvent donc produire des charges utiles plus grandes. RTP fournit son propre format de bourrage (comme on le voit à la Figure 1) qui grâce à l'indicateur de bourrage dans l'en-tête RTP a l'avantage en termes de compacité par rapport aux bourrages qui utilisent des codes sans préfixe. Ce bourrage RTP DEVRA être la méthode par défaut pour les transformations qui exigent un bourrage. Les transformations PEUVENT spécifier d'autres méthodes de bourrage, et DOIVENT alors en spécifier la quantité, le format, et le traitement. Il est important de noter que les transformations de chiffrement qui utilisent le bourrage sont vulnérables à des attaques subtiles, en particulier lorsque l'authentification du message n'est pas utilisée [V02]. Chaque spécification d'une nouvelle transformation de chiffrement doit étudier attentivement et décrire les implications pour la sécurité du bourrage utilisé. Les codes d'authentification de message définissent leur propre bourrage, de sorte que celui par défaut ne s'applique pas aux transformations d'authentification.

Les champs MKI FACULTATIF et étiquette d'authentification RECOMMANDÉ sont les seuls définis par SRTP qui ne sont pas dans RTP. Seul l'alignement sur 8 bits est prévu.

MKI *(Master Key Identifier)* identifiant de clé maîtresse : longueur configurable, FACULTATIF. Le MKI est défini, signalé, et utilisé par la gestion de clé. Le MKI identifie la clé maîtresse à partir de laquelle la ou les clés de session sont déduites et qui authentifient et/ou chiffrent le paquet en question. Noter que le MKI NE DEVRA PAS identifier le contexte cryptographique SRTP, qui est identifié conformément au paragraphe 3.2.3. Le MKI PEUT être utilisé par la gestion de clé pour les besoins du changement de clé, en identifiant une certaine clé maîtresse au sein du contexte cryptographique (paragraphe 3.2.1).

Étiquette d'authentification : longueur configurable, RECOMMANDÉ. L'étiquette d'authentification est utilisée pour porter les données d'authentification de message. La portion authentifiée d'un paquet SRTP consiste en l'en-tête RTP suivi par la portion chiffrés du paquet SRTP. Donc, si le chiffrement et l'authentification sont toutes deux appliquées, le chiffrement DEVRA être appliqué avant l'authentification sur le côté envoyeur et l'inverse sur le côté receveur. L'étiquette d'authentification assure l'authentification de l'en-tête et de la charge utile RTP, et elle assure indirectement la protection contre la répétition par l'authentification du numéro de séquence. Noter que le MKI n'est pas protégé quant à son intégrité car cela n'assure aucune protection supplémentaire.

3.2 Contextes cryptographiques de SRTP

Chaque flux SRTP exige de l'envoyeur et du receveur qu'ils entretiennent les informations d'état cryptographique. Ces informations sont appelées le "contexte cryptographique".

SRTP utilise deux types de clés : les clés de session et les clés maîtresses. Par "clé de session", on entend une clé qui est utilisée directement dans une transformation cryptographique (par exemple, le chiffrement ou l'authentification du message) et par "clé maîtresse", on entend une chaîne binaire aléatoire (donnée par le protocole de gestion de clé) à partir de laquelle les clés de session sont déduites d'une façon cryptographiquement sûre. La ou les clés maîtresses et les autres paramètres du contexte cryptographique sont fournis par des mécanismes de gestion de clé externes à SRTP, voir la Section 8.

3.2.1 Paramètres indépendants de la transformation

Des paramètres indépendants de la transformation sont présents dans le contexte cryptographique indépendamment du chiffrement ou des transformations d'authentification particuliers qui sont utilisés. Les paramètres indépendants de la transformation du contexte cryptographique pour SRTP consistent en :

* un compteur de débordement (ROC, *Roll-Over Counter*) de 32 bits non signé, qui enregistre combien de fois le numéro de séquence RTP de 16 bits a été remis à zéro après être passé par 65 535. À la différence du numéro de séquence (SEQ) que SRTP extrait de l'en-tête du paquet RTP, le ROC est entretenu par SRTP comme décrit au paragraphe 3.3.1. On définit l'indice du paquet SRTP correspondant à un certain ROC et numéro de séquence RTP comme étant la quantité de 48 bits i = 2^16 * ROC + SEQ.

* seulement pour le receveur, un numéro de séquence de 16 bits s_l, qui peut être vu comme le plus fort numéro de séquence RTP reçu (voir son traitement au paragraphe 3.3.1) qui DEVRAIT être authentifié car l'authentification de message est RECOMMANDÉE,

* un identifiant pour l'algorithme de chiffrement, c'est-à-dire, le chiffre et son mode de fonctionnement,

* un identifiant pour l'algorithme d'authentification du message,
* une liste de répétitions, entretenue par le seul receveur (lorsque l'authentification et la protection contre la répétition sont fournies) contenant les indices des paquets SRTP récemment reçus et authentifiés,

* un indicateur MKI (0/1) pour savoir si un MKI est présent dans les paquets SRTP et SRTCP,

* si l'indicateur MKI est réglé à un, la longueur (en octets) du champ MKI, et (pour l'envoyeur) la valeur réelle du MKI actuellement actif (la valeur de l'indicateur MKI et sa longueur DOIVENT rester fixes pour la durée de vie du contexte),

* les clés maîtresses, qui DOIVENT être aléatoires et rester secrètes,

* pour chaque clé maîtresse, il y a un compteur du nombre de paquets SRTP qui ont été traités (envoyés) avec cette clé maîtresse (essentiel pour la sécurité, voir le paragraphe 3.3.1 et la Section 9),

* des entiers non négatifs n_e, et n_a, qui déterminent la longueur des clés de session pour le chiffrement, et l'authentification de message.

De plus, pour chaque clé maîtresse, un flux SRTP PEUT utiliser les valeurs associées suivantes :

* un sel maître, à utiliser dans la déduction de clé des clés de session. Cette valeur, lorsque elle est utilisée, DOIT être aléatoire, mais PEUT être publique. L'utilisation d'un sel maître est vivement RECOMMANDÉE, voir au paragraphe 9.2. Un sel "NUL" est traité comme 00...0.

* un entier dans l'ensemble {1,2,4,...,2^24}, le "taux_de_déduction_de_clé", où une valeur inspécifiée est traitée comme zéro. La contrainte d'être une puissance de 2 simplifie la mise en œuvre de la déduction de clé de session ; voir au paragraphe 4.3.

* une valeur de MKI,

* les valeurs <From, To>, qui spécifient la durée de vie d'une clé maîtresse, exprimées en termes de deux valeurs d'indice

de 48 bits à l'intérieur des gammes de validité de la clé maîtresse (incluant les points d'extrémité de la gamme). Pour l'utilisation de <From, To>, voir le paragraphe 8.1.1. <From, To> est une solution de remplacement du MKI et suppose qu'une clé maîtresse est en correspondance bijective avec la clé de session SRTP sur laquelle est définie la gamme <From, To>.

SRTCP DEVRA par défaut partager le contexte cryptographique de SRTP, avec les exceptions suivantes :

* aucun compteur de débordement ni valeur de s_l n'a besoin d'être entretenu car l'indice RTCP est porté explicitement dans chaque paquet SRTCP,

* une liste de répétitions séparée est conservée (lorsque la protection contre la répétition est fournie),

* SRTCP entretient un compteur séparé pour la clé maîtresse (même si la clé maîtresse est la même que celle pour SRTP, voir ci-dessous) comme moyen pour tenir un compte du nombre de paquets SRTCP qui ont été traités avec cette clé.

Noter en particulier que la ou les clés maîtresses PEUVENT être partagées entre SRTP et le SRTCP correspondant, si les transformations prédéfinies (y compris les déductions de clé) sont utilisées, mais la ou les clés de session NE DOIVENT PAS être ainsi partagées.

De plus, il peut y avoir des cas (voir la Section 8 et le paragraphe 9.1) où plusieurs flux SRTP au sein d'une certaine session RTP, identifiés par leur source de synchronisation (les SSRC, qui font partie de l'en-tête RTP) partagent la plupart des paramètres de contexte cryptographique (y compris les éventuelles clés maîtresse et de session). Dans de tels cas, tout comme dans le partage normal de paramètres SRTP/SRTCP ci-dessus, des listes distinctes de répétition et des compteurs de paquet pour chaque flux (SSRC) DOIVENT encore être tenus. Des indices SRTP séparés DOIVENT alors aussi être tenus.

Un sommaire des paramètres, des transformations prédéfinies, et des valeurs par défaut des paramètres ci-dessus (et des autres paramètres SRTP) se trouve à la Section 5 et au paragraphe 8.2.

3.2.2 Paramètres dépendants de la transformation

Tous les paramètres de chiffrement, authentification/intégrité, et déduction de clé sont définis dans la section transformations (Section 4). Des exemples typiques de tels paramètres sont la taille du bloc de chiffrement, les clés de session, les données pour la formation de vecteur d'initialisation (IV), etc. Les futures spécifications de transformations SRTP DEVRONT inclure une section qui fasse la liste des paramètres supplémentaires de contexte cryptographique pour cette transformation, s'il en est.

3.2.3 Transposition des paquets SRTP dans des contextes cryptographiques

On rappelle qu'une session RTP est définie pour chaque participant [RFC3550] par une paire d'adresses de transport de destination (une adresse réseau plus une paire d'accès pour RTP et RTCP) et qu'une session multimédia est définie comme une collection de sessions RTP. Par exemple, une session multimédia particulière comporte une session audio RTP, une session vidéo RTP, et une session texte RTP.

Un contexte cryptographique DEVRA être identifié de façon univoque par le triplet d'identifiant de contexte :

Identifiant de contexte = <SSRC, adresse de réseau de destination, numéro d'accès de transport de destination>

où l'adresse de réseau de destination et l'accès de transport de destination sont ceux qui sont dans le paquet SRTP. On suppose que, lorsque on lui présente ces informations, la gestion de clé retourne un contexte avec les informations décrites au paragraphe 3.2.

Comme on l'a noté ci-dessus, SRTP et SRTCP partagent par défaut le gros des paramètres du contexte cryptographique. Donc, restituer les paramètres du contexte cryptographique pour un flux SRTCP peut impliquer en pratique un lien avec le contexte cryptographique SRTP correspondant. Il appartient à la mise en œuvre d'assurer un tel lien, car l'accès RTCP peut n'être pas directement déductible du seul accès RTP. Autrement, la gestion de clé peut choisir de fournir des contextes SRTP et SRTCP séparés, dupliquant les paramètres communs (tels que la ou les clés maîtresses). Cette dernière approche permet aussi à SRTP et SRTCP d'utiliser, par exemple, des transformations distinctes, si cela est souhaité. Des considérations similaires se font jour lorsque plusieurs flux SRTP, formant une partie d'une seule session RTP, partagent des clés et autres paramètres.

Si aucun contexte valide ne peut être trouvé pour un paquet correspondant à un certain identifiant de contexte, le paquet DOIT être éliminé.

3.3 Traitement du paquet SRTP

Ce qui suit s'applique à SRTP. SRTCP est décrit au paragraphe 3.4. En supposant que l'initialisation du ou des contextes cryptographiques a eu lieu via la gestion de clé, l'envoyeur DEVRA faire ce qui suit pour construire un paquet SRTP :

1. Déterminer quel contexte cryptographique utiliser, comme décrit au paragraphe 3.2.3.

2. Déterminer l'indice du paquet SRTP en utilisant le compteur de débordement, le plus fort numéro de séquence dans le contexte cryptographique, et le numéro de séquence dans le paquet RTP, comme décrit au paragraphe 3.3.1.

3. Déterminer la clé maîtresse et le sel maître. Ceci est fait en utilisant l'indice déterminé à l'étape précédente ou le MKI actuel dans le contexte cryptographique, conformément au paragraphe 8.1.

4. Déterminer les clés de session et le sel de session (si ils sont utilisés par les transformations) comme décrit au paragraphe 4.3, en utilisant la clé maîtresse, le sel maître, le taux_de_déduction_de_clé, et les longueurs de clé de session dans le contexte cryptographique avec l'indice, déterminés aux étapes 2 et 3.

5. Chiffrer la charge utile RTP pour produire la portion chiffrée du paquet (voir au paragraphe 4.1, les chiffrements définis). Cette étape utilise l'algorithme de chiffrement indiqué dans le contexte cryptographique, la clé de chiffrement de session et le sel de session (s'il est utilisé) trouvés à l'étape 4 avec l'indice trouvé à l'étape 2.

6. Si l'indicateur MKI est réglé à un, ajouter le MKI au paquet.

7. Pour l'authentification de message, calculer l'étiquette d'authentification pour la portion authentifiée du paquet, comme décrit au paragraphe 4.2. Cette étape utilise le compteur de débordement actuel, l'algorithme d'authentification indiqué dans le contexte cryptographique, et la clé d'authentification de session trouvée à l'étape 4. Ajouter l'étiquette d'authentification au paquet.

8. Si nécessaire, mettre à jour le ROC comme au paragraphe 3.3.1, en utilisant l'indice du paquet déterminé à l'étape 2.

Pour authentifier et déchiffrer un paquet SRTP, le receveur DEVRA faire ce qui suit :

1. Déterminer quel contexte cryptographique utiliser comme décrit au paragraphe 3.2.3.
2. Appliquer l'algorithme du paragraphe 3.3.1 pour obtenir l'indice du paquet SRTP. L'algorithme utilise le compteur de débordement et le plus fort numéro de séquence dans le contexte cryptographique avec le numéro de séquence dans le paquet SRTP, comme décrit au paragraphe 3.3.1.
3. Déterminer la clé maîtresse et le sel maître. Si l'indicateur MKI dans le contexte est réglé à un, utiliser le MKI dans le paquet SRTP, autrement, utiliser l'indice de l'étape précédente, conformément au paragraphe 8.1.
4. Déterminer les clés de session, et le sel de session (si il est utilisé par la transformation) comme décrit au paragraphe 4.3, en utilisant la clé maîtresse, le sel maître, le taux_de_déduction_de_clé et les longueurs de clé de session dans le contexte cryptographique avec l'indice, déterminés aux étapes 2 et 3.
5. Pour l'authentification de message et la protection contre la répétition, vérifier d'abord si le paquet a été répété (paragraphe 3.3.2) en utilisant la liste des répétitions et l'indice comme déterminé à l'étape 2. Si le paquet est jugé répété, il DOIT alors être éliminé, et l'événement DEVRAIT être enregistré.

   Ensuite, effectuer la vérification de l'étiquette d'authentification, en utilisant le compteur de débordement de l'étape 2, l'algorithme d'authentification indiqué dans le contexte cryptographique, et la clé d'authentification de session provenant de l'étape 4. Si le résultat est "ÉCHEC D'AUTHENTIFICATION" (voir au paragraphe 4.2) le paquet DOIT être éliminé de la suite du traitement et l'événement DEVRAIT être enregistré.
6. Déchiffrer la portion chiffrée du paquet (voir au paragraphe 4.1 les chiffrements définis) en utilisant l'algorithme de déchiffrement indiqué dans le contexte cryptographique, la clé de chiffrement de session et le sel (si il est utilisé) trouvés à l'étape 4 avec l'indice de l'étape 2.
7. Mettre à jour le compteur de débordement et le plus haut numéro de séquence, s_l, dans le contexte cryptographique comme au paragraphe 3.3.1, en utilisant l'indice de paquet estimé à l'étape 2. Si la protection contre la répétition est fournie, mettre aussi à jour la liste des répétitions comme décrit au paragraphe 3.3.2.
8. Lorsque ils sont présents, retirer du paquet les champs MKI et Étiquette d'authentification.

3.3.1 Détermination de l'indice de paquet et mise à jour du ROC et de s_l

Les mises en œuvre SRTP utilisent un indice de paquet "implicite" pour la mise en séquence, c'est-à-dire que tous les

indices ne sont pas explicitement portés dans le paquet SRTP. Pour les transformations prédéfinies, l'indice i est utilisé dans la protection contre la répétition (paragraphe 3.3.2) dans le chiffrement (paragraphe 4.1) dans l'authentification de message (paragraphe 4.2) et pour la déduction de clé (paragraphe 4.3).

Lorsque la session commence, le côté envoyeur DOIT régler le compteur de débordement (ROC, *Roll Over Counter*) à zéro. Chaque fois que le numéro de séquence RTP, SEQ, revient à zéro modulo 2^16, le côté envoyeur DOIT incrémenter ROC de un, modulo 2^32 (voir les aspects de sécurité ci-dessous). L'indice du paquet de l'envoyeur est alors défini par

i = 2^16 * ROC + SEQ.

Les mises en œuvre du côté receveur utilisent le numéro de séquence RTP pour déterminer l'indice correct d'un paquet, qui est la situation du paquet dans la séquence de tous les paquets SRTP. Une approche robuste pour la bonne utilisation d'un compteur de débordement exige que son traitement et son utilisation soient bien définis. En particulier, les paquets RTP déclassés avec des numéros de séquence proches de 2^16 ou de zéro doivent être traités de façon appropriée.

L'estimation d'indice se fonde sur les valeurs de ROC et s_l tenues en local par le receveur. À l'établissement de la session, le ROC DOIT être réglé à zéro. Les receveurs qui se joignent à une session en cours DOIVENT obtenir la valeur de ROC actuelle en utilisant une signalisation hors bande comme la signalisation de gestion de clé. De plus, le receveur DEVRA initialiser s_l au numéro de séquence (SEQ) RTP du premier paquet SRTP observé (sauf si la valeur initiale est fournie par une signalisation hors bande comme la gestion de clé).

Sur des paquets SRTP consécutifs, le receveur DEVRAIT estimer l'indice comme i = 2^16 * v + SEQ, où v est choisi dans l'ensemble { ROC-1, ROC, ROC+1 } (modulo 2^32) de sorte que i soit proche (au sens modulo 2^48) de la valeur 2^16 * ROC + s_l (voir le pseudocode à l'Appendice A).

Après que le paquet a été traité et authentifié (lorsque activé pour les paquets SRTP pour la session) le receveur DOIT utiliser v pour une mise à jour conditionnelle de ses variables s_l et ROC comme suit. Si v = (ROC-1) mod 2^32, il n'y a alors pas de mise à jour de s_l ou ROC. Si v = ROC, alors s_l est réglé à SEQ si et seulement si SEQ est supérieur au s_l actuel ; il n'y a pas de changement pour ROC. Si v = (ROC+1) mod 2^32, alors s_l est réglé à SEQ et ROC est réglé à v.

Après un changement de clé (changement pour une nouvelle clé maîtresse) le compteur de débordement conserve toujours sa séquence de valeurs, c'est-à-dire qu'il NE DOIT PAS être remis à zéro.

Comme le compteur de débordement est long de 32 bits et que le numéro de séquence est long de 16 bits, le nombre maximum de paquets appartenant à un flux SRTP qui peut être sécurisé avec la même clé est de 2^48 en utilisant les transformations prédéfinies. Après que ce nombre de paquets SRTP a été envoyé avec une certaine clé (maîtresse ou de session) l'envoyeur NE DOIT PAS envoyer plus de paquets avec cette clé. (Il existe une limite similaire pour SRTCP, qui en pratique peut être plus restrictive, voir au paragraphe 9.2.) Cette limitation apporte un avantage pour la sécurité en fixant une limite supérieure à la quantité de trafic qui peut passer avant que les clés de chiffrement ne soient changées. Le changement de clés (voir au paragraphe 8.1) DOIT être déclanché, avant cette quantité de trafic, et PEUT être déclanché plus tôt, par exemple, pour une sécurité et un contrôle d'accès au support accrus. Recourir à la déduction de clé au moyen d'un taux_de_déduction_de_clé (voir au paragraphe 4.3) différent de zéro donne aussi une plus forte sécurité mais ne change pas la valeur absolue maximum ci-dessus.

Du côté receveur, il faut faire attention à la mise à jour de s_l et ROC : si l'authentification de message n'est pas présente, ni l'initialisation de s_l, ni la mise à jour du compteur de débordement ne peuvent être rendues complètement robustes. L'approche de "l'indice implicite" du receveur fonctionne pour les transformations prédéfinies tant que le réarrangement et la perte de paquets ne sont pas trop grands et que des erreurs binaires ne se produisent pas de façon malencontreuse. En particulier, 2^15 paquets devraient être perdus, ou un paquet devrait être déclassé de 2^15 paquets avant que soit perdue la synchronisation. Des pertes ou déclassements aussi drastiques perturberaient vraisemblablement l'application RTP elle-même.

L'algorithme pour l'estimation d'indice et la mise à jour du compteur de débordement est l'affaire de la mise en œuvre, et devrait tenir compte de l'environnement (par exemple, le taux de perte de paquets) et des cas où la synchronisation est probablement perdue, par exemple, lorsque le numéro de séquence initial (choisi au hasard par RTP) n'est pas connu à l'avance (non envoyé dans le protocole de gestion de clé) mais peut être proche du retour à zéro modulo 2^16.

Un schéma plus élaboré et plus robuste que celui donné ci-dessus est le traitement du propre "compteur de débordement" de RTP, voir l'Appendice A.1 de la [RFC3550].

3.3.2 Protection contre la répétition

Une protection sûre contre la répétition n'est possible que lorsque la protection de l'intégrité est présente. Il est RECOMMANDÉ d'utiliser la protection contre la répétition, à la fois pour RTP et pour RTCP, car seule, la protection de l'intégrité ne peut pas assurer la sécurité contre les attaques en répétition.

Un paquet est "répété" lorsque il est mémorisé par un adversaire, et ensuite réinjecté sur le réseau. Lorsque l'authentification de message est fournie, SRTP protège contre de telles attaques grâce à une liste de répétitions. Chaque receveur SRTP entretient une liste de répétitions, qui contient par conception les indices de tous les paquets qui ont été reçus et authentifiés. En pratique, la liste peut utiliser une approche de "fenêtre glissante", de sorte qu'une quantité fixe de mémorisation suffit pour la protection contre la répétition. Les indices de paquet qui sont après l'indice du paquet dans le

contexte de plus de SRTP-WINDOW-SIZE *(taille de fenêtre SRTP)* peuvent être supposés avoir été reçus ; SRTP-WINDOW-SIZE est un paramètre du côté receveur qui dépend de la mise en œuvre, et DOIT être d'au moins 64, mais il PEUT être réglé à une valeur plus élevée.

Le receveur vérifie l'indice d'un paquet entrant dans la liste de répétitions et la fenêtre. Seuls les paquets dont l'indice est après la fenêtre, ou, dans la fenêtre mais pas encore reçus, DEVRONT être acceptés.

Après que le paquet a été authentifié (si nécessaire, la fenêtre est d'abord décalée vers l'avant) la liste de répétition DEVRA être mise à jour avec le nouvel indice.

La liste de répétitions peut être efficacement mise en œuvre en utilisant un codage d'équivalence binaire pour représenter quels paquets ont été reçus, comme décrit dans l'architecture de sécurité pour IP [RFC2401].

3.4 RTCP sécurisé

RTCP sécurisé suit la définition de RTP sécurisé. SRTCP ajoute trois nouveaux champs obligatoires (l'indice SRTCP, un "fanion de chiffrement", et l'étiquette d'authentification) et un champ facultatif (le MKI) à la définition du paquet RTCP. Les trois champs obligatoires DOIVENT être ajoutés à un paquet RTCP afin de former un paquet SRTCP équivalent. Les champs ajoutés suivent toutes les autres extensions spécifiques du profil.

Conformément au paragraphe 6.1 de la [RFC3550], il y a un format de paquet EXIGÉ pour les paquets composés. SRTCP DOIT recevoir les paquets conformément à cette exigence, au sens que la première partie DOIT être un rapport d'envoyeur ou un rapport de receveur. Cependant, le préfixe de chiffrement RTCP (quantité aléatoire de 32 bits) spécifié dans la présente section NE DOIT PAS être utilisé car, comme on l'établit ici, il n'est applicable qu'à la méthode de chiffrement spécifiée dans la [RFC3550] et n'est pas nécessaire pour les mécanismes de chiffrement utilisés dans SRTP.

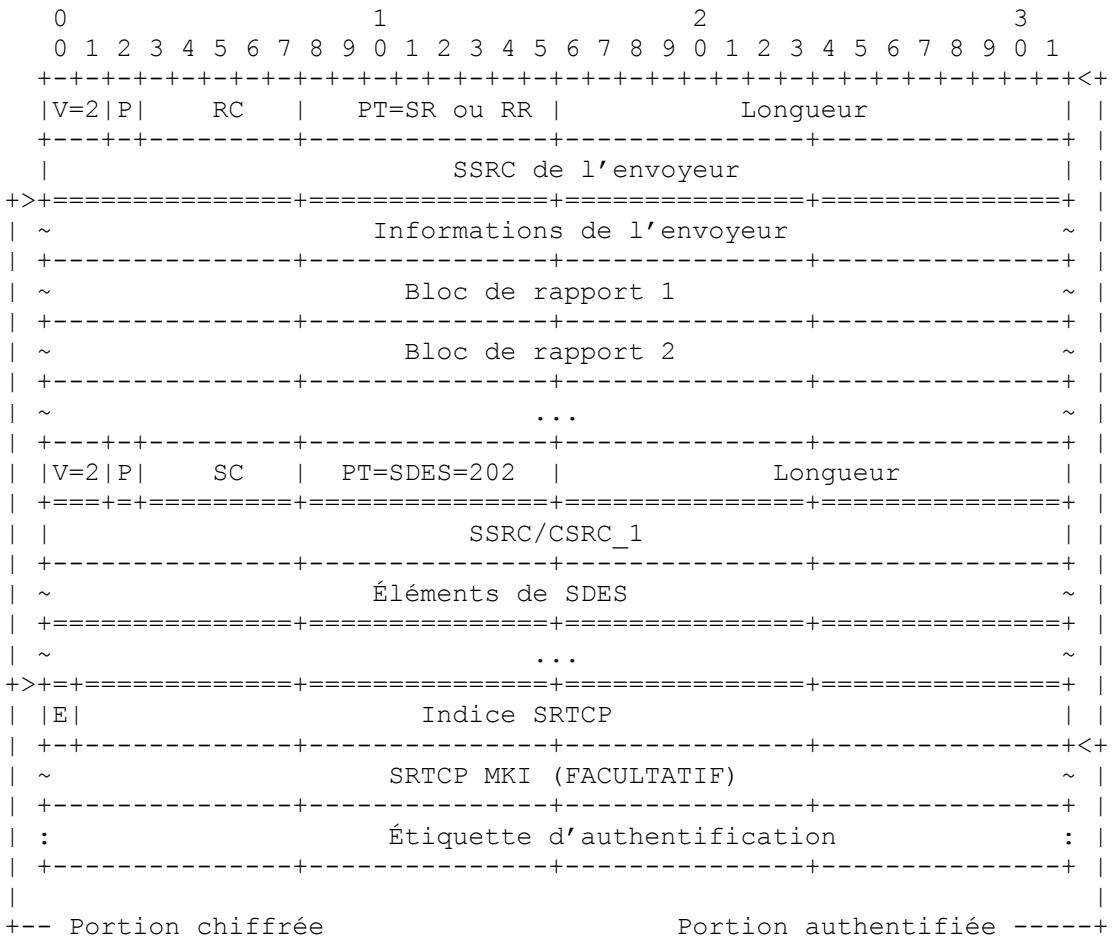


Figure 2. Exemple du format d'un paquet RTCP sécurisé, consistant en un paquet RTCP composé sous-jacent avec un rapport d'envoyeur et un paquet SDES

La portion chiffrée d'un paquet SRTCP comporte le chiffrement (paragraphe 4.1) de la charge utile RTCP du paquet RTCP composé équivalent, depuis le premier paquet RTCP, c'est-à-dire, depuis le neuvième (9) octet jusqu'à la fin du paquet composé. La portion authentifiée d'un paquet SRTCP comporte le paquet RTCP équivalent entier (éventuellement composé) le fanion E, et l'indice SRTCP (après que tout chiffrement a été appliqué à la charge utile).

Les champs ajoutés sont :
Fanion E : 1 bit, EXIGÉ
Le fanion E indique si le paquet SRTCP en cours est chiffré ou non chiffré. Le paragraphe 9.1 de la [RFC3550] permet le partage d'un paquet RTCP composé en deux paquets de couche inférieure, un à chiffrer et un à envoyer en clair. Le bit E réglé à "1" indique le paquet chiffré, et "0" indique le paquet non chiffré.

indice SRTCP : 31 bits, EXIGÉ
L'indice SRTCP est un compteur de 31 bits pour le paquet SRTCP. l'indice est explicitement inclus dans chaque paquet, à la différence de l'approche de l'indice "implicite" utilisé pour SRTP. L'indice SRTCP DOIT être réglé à zéro avant l'envoi du premier paquet SRTCP, et DOIT être incrémenté de un, modulo 2^31, après l'envoi de chaque paquet SRTCP. En particulier, après un changement de clé, l'indice SRTCP NE DOIT PAS être remis à nouveau à zéro.

Étiquette d'authentification : longueur configurable, EXIGÉ
L'étiquette d'authentification est utilisée pour porter les données d'authentification de message.

MKI : longueur configurable, FACULTATIF
Le MKI est l'indicateur de clé maîtresse, et fonctionne conformément à la définition de MKI à la Section 3.

SRTCP utilise les paramètres de contexte cryptographique et de traitement de paquet de SRTP par défaut, avec les changements suivants :

* Le receveur n'a pas besoin "d'estimer" l'indice, car il est explicitement signalé dans le paquet.

* Le chiffrement SRTCP prédéfini est spécifié au paragraphe 4.1, mais en utilisant la définition de la portion chiffrée SRTCP donnée dans ce paragraphe, et en utilisant l'indice SRTCP comme indice i. La transformation du chiffrement et des paramètres qui s'y rapportent DEVRA par défaut être la même que celles choisies pour la protection du ou des flux SRTP associés, tandis que l'algorithme NUL DEVRA être appliqué aux paquets RTCP qui ne sont pas à chiffrer. SRTCP peut avoir une transformation de chiffrement différente de celle utilisée par le SRTP correspondant. l'utilisation attendue de cette disposition est lorsque la première a le chiffrement NUL et que la dernière a un chiffrement non NUL.

La valeur du fanion E est allouée par l'envoyeur selon que le paquet a été chiffré ou non.

* Le déchiffrement SRTCP est effectué comme à la Section 4, mais seulement si le fanion E est égal à 1. Si il en est ainsi, la portion chiffrée est déchiffrée, en utilisant l'indice SRTCP comme indice i. Au cas où le fanion E est 0, la charge utile est simplement laissée inchangée.

* La protection SRTCP contre la répétition est comme défini au paragraphe 3.3.2, mais en utilisant l'indice SRTCP comme indice i et une liste de répétitions distincte qui est spécifique de SRTCP.

* L'étiquette d'authentification SRTCP prédéfinie est spécifiée au paragraphe 4.2, mais avec la portion authentifiée du paquet SRTCP donnée dans ce paragraphe (qui comporte l'indice). La transformation d'authentification et les paramètres qui s'y rapportent (par exemple, la taille de clé) DEVRA par défaut être la même que celle choisie pour la protection du ou des flux SRTP associés.

* Dans la dernière étape du traitement, seul l'envoyeur a besoin de mettre à jour la valeur de l'indice SRTCP en l'incrémentant modulo 2^31 et pour des raisons de sécurité, l'envoyeur DOIT aussi vérifier le nombre de paquets SRTCP traités, voir au paragraphe 9.2.

L'authentification de message pour RTCP est EXIGÉE, car c'est le protocole de contrôle (par exemple, il a un paquet BYE) pour RTP.

On doit prendre des précautions pour que l'expansion du paquet dans SRTCP (due à l'ajout des champs) ne soit cause que les messages SRTCP utilisent plus que leur part de la bande passante de RTCP. Pour éviter cela, les deux mesures suivantes DOIVENT être prises :

1. Lors de l'initialisation de la variable RTCP "avg_rtcp_size" définie au paragraphe 6.3 de la [RFC3550], il DOIT inclure la taille des champs qui seront ajoutés par SRTCP (indice, bit E, étiquette d'authentification, et lorsque il est présent, le MKI).

2. Lors de la mise à jour de "avg_rtcp_size" en utilisant la variable "taille_de_paquet" (paragraphe 6.3.3 de la [RFC3550]) la valeur de "taille_de_paquet" DOIT inclure la taille des champs supplémentaires ajoutés par SRTCP.

Ces mesures étant prises, les messages SRTCP ne vont pas utiliser plus que la bande passante allouée. L'effet de la taille des champs ajoutés sur le trafic SRTCP sera que les messages seront envoyés avec des intervalles de paquet plus longs.

L'augmentation des intervalles sera directement proportionnelle à la taille des champs ajoutés. Pour les transformations prédéfinies, la taille des champs ajoutés sera d'au moins 14 octets, et avec une limite supérieure qui dépend de la taille du MKI et de l'étiquette d'authentification.

4. Transformations cryptographiques prédéfinies

Bien que de nombreux algorithmes de chiffrement et d'authentification de message puissent être utilisés dans SRTP, on définit ci-dessous des algorithmes par défaut afin d'éviter la complexité d'une spécification de codages pour la signalisation d'identifiants d'algorithme et de paramètres. Les algorithmes définis ont été choisis parce qu'ils satisfont aux objectifs énumérés à la Section 2. Des recommandations sur la façon d'étendre SRTP avec les nouvelles transformations sont données à la Section 6.

4.1 Chiffrement

Les paramètres suivants sont communs aux transformations de chiffrement prédéfinies et non NULLES, spécifiées dans cette section.

* BLOCK_CIPHER-MODE indique le chiffrement de bloc utilisé et son mode de fonctionnement
* n_b est la taille en bits du bloc pour le chiffrement de bloc
* k_e est la clé de chiffrement de session
* n_e est la longueur en bits de k_e
* k_s est la clé salée de session
* n_s est la longueur en bits de k_s
* SRTP_PREFIX_LENGTH est la longueur en octets du préfixe de flux de clé, un entier non négatif, spécifié par le code d'authentification de message utilisé.

Les clés de session et sels distincts pour SRTP/SRTCP sont par défaut déduits comme spécifié au paragraphe 4.3.

Les transformations de chiffrement définies dans SRTP transposent l'indice de paquet SRTP et la clé secrète en un segment de flux de clé pseudo-aléatoire. Chaque segment de flux de clé chiffre un seul paquet RTP. Le processus de chiffrement d'un paquet consiste à générer le segment de flux de clé correspondant au paquet, et ensuite de OUXer au bit près ce segment de flux de clé sur la charge utile du paquet RTP pour produire la portion chiffrée du paquet SRTP. Au cas où la taille de la charge utile n'est pas un entier multiple de n_b bits, les bits en excédent (de moindre poids) du flux de clé sont simplement éliminés. Le déchiffrement est fait de la même façon, mais en échangeant les rôles du texte en clair et du texte chiffré.

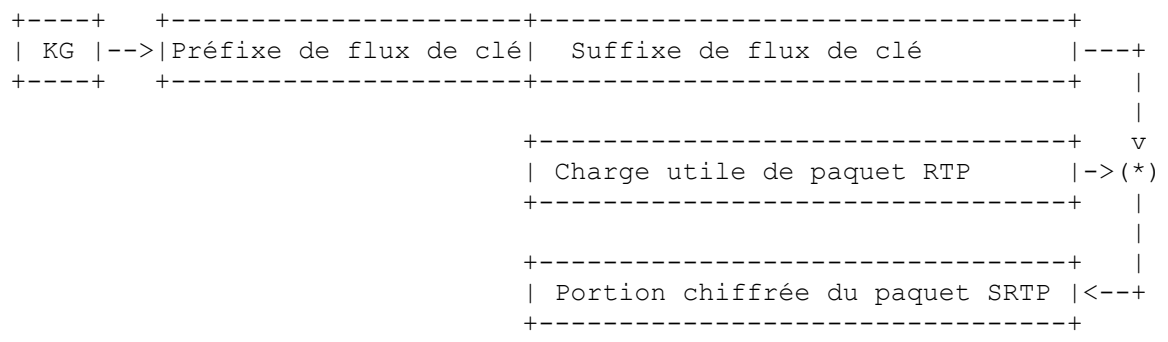


Figure 3 : Traitement par défaut du chiffrement SRTP.

Ici KG *(keystream generator)* note le générateur de flux de clé, et (*) note l'opération OUX au bit près.

La définition de la façon de générer le flux de clés, l'indice étant donné, dépend du chiffrement et de son mode de fonctionnement. Ci-dessous sont définis deux générateurs de flux de clés. Le chiffrement NUL est aussi défini, pour être utilisé lorsque le chiffrement de RTP n'est pas exigé.

La définition SRTP du flux de clés est illustré à la Figure 3. Les octets initiaux de chaque segment de flux de clés PEUVENT être réservés pour être utilisés dans un code d'authentification de message, auquel cas, le flux de clés utilisé pour le chiffrement commence immédiatement après le dernier octet réservé. Les octets réservés initiaux sont appelés le "préfixe de flux de clé" (à ne pas confondre avec le "préfixe de chiffrement" du paragraphe 6.1 de la [RFC3550]) et les octets restants sont appelés le "suffixe de flux de clé". Le préfixe de flux de clé NE DOIT PAS être utilisé pour le

chiffrement. Le processus est illustré à la Figure 3.

Le nombre d'octets dans le préfixe de flux de clé est noté SRTP_PREFIX_LENGTH. Le préfixe de flux de clé est indiqué par une valeur positive, non zéro, de SRTP_PREFIX_LENGTH. Cela signifie que, même si la confidentialité n'est pas à assurer, le résultat du générateur de flux de clé peut toujours avoir besoin d'être calculé pour l'authentification de paquet, auquel cas, le générateur de flux de clé par défaut (mode) DEVRA être utilisé.

Le chiffrement par défaut est la norme de chiffrement évolué (AES, *Advanced Encryption Standard*) [AES], et on définit deux modes de fonctionnement de AES, (1) AES en mode compteur d'entier segmenté, et (2) AES en mode f8. Dans le reste de cette section, soit E(k,x) l'AES appliqué à la clé k et le bloc d'entrée x.

4.1.1 AES en mode compteur

Conceptuellement, le mode compteur [AES-CTR] consiste à chiffrer des entiers successifs. La définition réelle est un peu plus compliquée, afin de rendre aléatoire le point de départ de la séquence d'entiers. Chaque paquet est chiffré avec un segment distinct de flux de clés, qui DEVRA être calculé comme suit.

Un segment de flux de clé DEVRA être l'enchaînement des blocs de résultat de 128 bits du chiffrement AES dans la direction du chiffrement, en utilisant la clé k = k_e, dans laquelle les indices de bloc sont en ordre croissant. Symboliquement, chaque segment de flux de clé ressemble à

E(k, IV) || E(k, IV + 1 mod 2^128) || E(k, IV + 2 mod 2^128) ...

où la valeur initiale (IV) d'une valeur d'entier de 128 bits DEVRA être définie par le SSRC, l'indice de paquet SRTP i, et la clé salée SRTP de session k_s, sont comme suit.

IV = (k_s * 2^16) OUX (SSRC * 2^64) OUX (i * 2^16)

Chacun des trois termes dans la somme OUX ci-dessus est bourré avec autant de zéros en tête que nécessaire pour rendre l'opération bien définie, considérée comme une valeur de 128 bits.

L'inclusion du SSRC permet l'utilisation de la même clé pour protéger des flux SRTP distincts au sein de la même session RTP ; voir les avertissements de sécurité au paragraphe 9.1.

Dans le cas de SRTCP, le SSRC du premier en-tête du paquet composé DOIT être utilisé, i DEVRA être l'indice SRTCP de 31 bits et k_e, k_s DEVRONT être remplacés par la clé de session et le sel de chiffrement SRTCP.

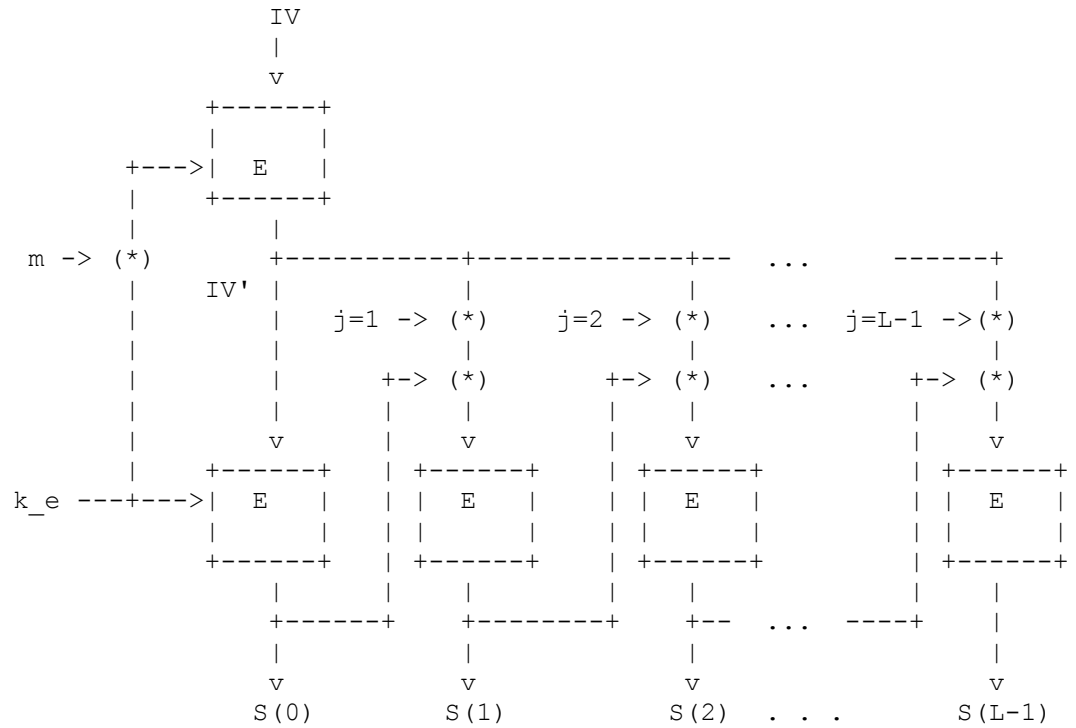
Noter que la valeur initiale, IV, est fixée pour chaque paquet et est formée en "réservant" 16 zéros dans les bits de moindre poids pour le compteur. Le nombre de blocs de flux de clés généré pour toute valeur fixe de IV NE DOIT PAS excéder 2^16 pour éviter la réutilisation du flux de clé ; voir ci-dessous. L'AES a une taille de bloc de 128 bits, de sorte que 2^16 blocs de résultat sont suffisants pour générer les 2^23 bits de flux de clé nécessaires pour chiffrer le plus grand paquet RTP possible (sauf pour les "jumbogrammes" IPv6 [RFC2675], qu'il est peu probable d'utiliser pour le trafic multimédia fondé sur RTP). Cette restriction sur la taille maximum en bits du paquet qui peut être chiffré assure la sécurité de la méthode de chiffrement en limitant l'efficacité des attaques probabilistes [BDJR].

Pour une clé en mode compteur particulière, chaque valeur IV utilisée comme entrée DOIT être distincte, afin d'éviter d'exposer la sécurité par une situation de double bourrage (paragraphe 9.1). Pour se plier à cette contrainte, une mise en œuvre DOIT s'assurer que la combinaison de l'indice de paquet SRTP de ROC || SEQ, et le SSRC utilisé dans la construction de l'IV sont distincts pour toute clé. Ne pas réussir à assurer cette unicité pourrait être catastrophique pour RTP sécurisé. C'est différent de la situation de RTP lui-même, qui peut être capable de tolérer de telles défaillances. Il est RECOMMANDÉ que, si un module de sécurité dédié est présent, les numéros de séquence RTP et le SSRC soient générés ou vérifiés par ce module (c'est-à-dire que le numéro de séquence et le traitement du SSRC dans un système SRTP doivent être protégés aussi bien que la clé).

4.1.2 AES en mode f8

Pour chiffrer les données UMTS (service universel de télécommunication avec les mobiles, comme réseaux de 3G) une solution (voir [f8-a] [f8-b]) connue sous le nom de algorithme f8 a été développée. En gros, le schéma proposé est une variante du mode de rebouclage de la sortie (OFB, *Output Feedback Mode*) [HAC], avec une fonction plus élaborée d'initialisation et de rebouclage. Comme en OFB normal, le cœur consiste en un chiffrement de bloc. On définit aussi ici l'utilisation de AES comme chiffrement de bloc à utiliser dans ce qu'on appelle le chiffrement RTP en "mode de fonctionnement f8". Le mode AES f8 DEVRA utiliser les mêmes tailles par défaut pour les clés de session et sel que l'AES en mode compteur.

La Figure 4 montre la structure du chiffrement de bloc, E, fonctionnant en mode f8.



(*) note OUX au bit près)

Figure 4 : Mode de fonctionnement f8

La figure représente le KG de la Figure 3, lorsque AES en mode f8 est utilisé.

4.1.2.1 Génération du flux de clés f8

Le vecteur d'initialisation (IV, *Initialization Vector*) DEVRA être déterminé comme décrit au paragraphe 4.1.2.2 (et au paragraphe 4.1.2.3 pour SRTCP).

Soit IV', S(j), et m qui notent n_b blocs de bits. Le flux de clés, S(0) ||... || S(L-1), pour un message de N bits DEVRA être défini en réglant IV' = E(k_e OUX m, IV), et S(-1) = 00..0. Pour j = 0, 1, .., L-1 où L = N/n_b (arrondi à l'entier le plus proche si il n'est pas déjà entier) calculer

S(j) = E(k_e, IV' OUX j OUX S(j-1))

Remarquer que le IV n'est pas utilisé directement. En fait, il est introduit à travers E sous une autre clé pour produire une valeur interne, "masquée" (notée IV') pour empêcher un attaquant d'obtenir connaissance des paires d'entrée/sortie.

Le rôle du compteur interne, j, est d'empêcher les cycles courts de flux de clés. La valeur du gabarit de clé m DEVRA être

m = k_s || 0x555..5,

c'est-à-dire, la clé de session salée, augmentée du schéma binaire 0101.. pour compléter la taille de clé désirée toute entière, n_e.

L'envoyeur NE DEVRAIT PAS générer plus de 2^32 blocs, ce qui est suffisant pour générer 2^39 bits de flux de clé. À la différence du mode compteur, il n'y a pas de seuil absolu au-dessus (en dessous) duquel il est garanti que f8 est non sûr (sûr). Le seuil ci-dessus a été choisi pour limiter, avec une marge de sécurité suffisante, la probabilité de comportement dégénératif dans la génération de flux de clé f8.

4.1.2.2 Formation du vecteur d'initialisation SRTP f8

L'objet de la formation d'IV suivante est de fournir un dispositif qu'on appellera authentification implicite d'en-tête (IHA, *implicit authentication header*) ; voir au paragraphe 9.5.

L'IV SRTP pour le bloc de 128 bits AES-f8 DEVRA être formé de la façon suivante :

IV = 0x00 || M || PT || SEQ || TS || SSRC || ROC

M, PT, SEQ, TS, SSRC DEVRONT être tirés de l'en-tête RTP ; ROC vient du contexte cryptographique.

La présence du SSRC au titre de l'IV permet d'utiliser AES-f8 lorsque une clé maîtresse est partagée entre plusieurs flux au sein de la même session RTP ; voir au paragraphe 9.1.

4.1.2.3 Formation du vecteur d'initialisation SRTCP f8

L'IV SRTCP pour un bloc AES-f8 de 128 bits DEVRA être formé de la façon suivante :

IV = 0..0 || E || indice SRTCP || V || P || RC || PT || longueur || SSRC

où V, P, RC, PT, longueur, SSRC DEVRONT être tirés du premier en-tête du paquet RTCP composé. E et indice SRTCP sont les champs de 1 bit et de 31 bits ajoutés au paquet.

4.1.3 Chiffrement NUL

Le chiffrement NUL est utilisé lorsque aucune confidentialité n'est requise pour RTP/RTCP. Le flux de clés peut être vu comme "000..0", c'est-à-dire que le chiffrement DEVRA simplement copier l'entrée de texte en clair dans la sortie de texte chiffré.

4.2 Authentification et intégrité de message

Tout au long de ce paragraphe, M note les données à protéger en intégrité. Dans le cas de SRTP, M DEVRA consister en la portion authentifiée du paquet (comme spécifié à la Figure 1) enchaînée avec le ROC, M = portion authentifiée || ROC ; dans le cas de SRTCP, M DEVRA seulement consister en la portion authentifiée (comme spécifié à la Figure 2).

Paramètres communs :

* AUTH_ALG est l'algorithme d'authentification
* k_a est la clé d'authentification de message de session
* n_a est la longueur en bits de la clé d'authentification
* n_tag est la longueur en bits de l'étiquette d'authentification de sortie
* SRTP_PREFIX_LENGTH est la longueur en octets du préfixe de flux de clé comme défini ci-dessus, un paramètre de AUTH_ALG

Les clés d'authentification de session distinctes pour SRTP/SRTCP sont par défaut déduites comme spécifié au paragraphe 4.3.

Les valeurs de n_a, n_tag, et SRTP_PREFIX_LENGTH DOIVENT être fixées pour toute valeur fixe particulière de la clé.

On décrit comme suit le processus de calcul des étiquettes d'authentification. L'envoyeur calcule l'étiquette de M et l'ajoute au paquet. Le receveur SRTP vérifie une paire message/étiquette d'authentification en calculant une nouvelle étiquette d'authentification sur M en utilisant l'algorithme et la clé choisis, et en la comparant à l'étiquette associée au message reçu. Si les deux étiquettes sont égales, la paire message/étiquette est alors valide ; autrement, elle est invalide et le message d'audit d'erreur "ÉCHEC D'AUTHENTIFICATION" DOIT être retourné.

4.2.1 HMAC-SHA1

La transformation d'authentification prédéfinie pour SRTP est HMAC-SHA1 [RFC2104]. Avec HMAC-SHA1, la SRTP_PREFIX_LENGTH (Figure 3) DEVRA être de 0. Pour SRTP (respectivement SRTCP) le HMAC DEVRA être appliqué à la clé d'authentification de session et à M comme spécifié ci-dessus, c'est-à-dire, HMAC(k_a, M). Le résultat HMAC DEVRA alors être tronqué aux n_tag bits de gauche.

4.3 Déduction de clé

4.3.1 Algorithme de déduction de clé

Sans considération de la transformation du chiffrement ou de l'authentification de message qui est employée (ce peut être une transformation SRTP prédéfinie ou une nouvelle, introduite selon la Section 6) les mises en œuvre SRTP interopérables DOIVENT utiliser la déduction de clé SRTP pour générer les clés de session. Une fois que le taux de déduction de clé est correctement signalé au début de la session, il n'est pas besoin d'autre communication entre les parties qui utilisent la déduction de clé SRTP.

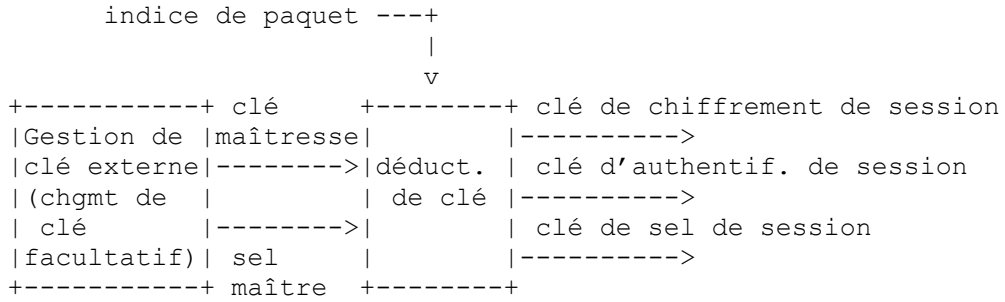


Figure 5 : Déduction des clés SRTP

Au moins une déduction de clé initiale DEVRA être effectuée par SRTP, c'est-à-dire, la première déduction de clé est EXIGÉE. D'autres applications de la déduction de clé PEUVENT être effectuées, conformément à la valeur du "taux_de_déduction_de_clé " dans le contexte cryptographique. La fonction de déduction de clé DEVRA être invoquée initialement avant le premier paquet, puis lorsque r > 0, une déduction de clé est effectuée chaque fois qu'un indice mod r égale zéro. Cela peut être vu comme un "rafraîchissement" des clés de session. La valeur du "taux_de_déduction_de_clé " DOIT rester fixe pour la durée de vie de la clé maîtresse associée.

Les mises en œuvre SRTP interopérables PEUVENT aussi déduire les clés de session salées pour les transformations de chiffrement, comme c'est fait dans les deux transformations prédéfinies.

Soit m et n des entiers positifs. Une famille de fonctions pseudo aléatoires (PRF, *pseudo-random function*) est un ensemble de fonctions de clés {PRF_n(k,x)} telles que pour la clé (secrète) aléatoire k, étant donnée x de m bits, PRF_n(k,x) est une chaîne de n bits, indistinguables par le calcul des chaînes aléatoires de n bits ; voir [HAC]. Pour les besoins de la déduction de clé dans SRTP, une PRF sûre avec m = 128 (ou plus) DOIT être utilisée, et une transformation de PRF par défaut est définie au paragraphe 4.3.3.

Soit "a DIV t" qui note la division d'entier par t, arrondie, et avec la convention que "a DIV 0 = 0" pour tout a. On fait aussi la convention de traiter "a DIV t" comme une chaîne binaire de même longueur que a, et donc "a DIV t" va en général avoir des zéros en tête.

La déduction de clé DEVRA être définie comme suit en termes de <étiquette>, une constante de 8 bits (voir ci-dessous) clé_maître et taux_de_déduction_de_clé , comme déterminé dans le contexte cryptographique, et indice, l'indice du paquet (c'est-à-dire, le ROC || SEQ de 48 bits pour SRTP):

* Soit r = indice DIV taux_de_déduction_de_clé (avec DIV comme défini ci-dessus).

* Soit clé_id = <étiquette> || r.
* Soit x = clé_id OUX sel_maître, où clé_id et sel_maître sont alignés de telle sorte que leurs bits de moindre poids s'accordent (alignement à droite).

<étiquette> DOIT être unique pour chaque type de clé à déduire. On définit actuellement <étiquette> 0x00 à 0x05 (voir ci-dessous) et de futures extensions PEUVENT spécifier de nouvelles valeurs dans la gamme 0x06 à 0xff pour d'autres objets. La clé SRTP de n-bit (ou sel) pour ce paquet DEVRA alors être déduite de la clé maîtresse, k_maître comme suit :

PRF_n(k_maître, x).

(La PRF peut spécifier en interne des formats et bourrages supplémentaires de x, voir par exemple, au paragraphe 4.3.3 la PRF par défaut.)

Les clés de session et le sel DEVRONT alors être déduits en utilisant :

- k_e (chiffrement SRTP) : <étiquette> = 0x00, n = n_e.
- k_a (authentification de message SRTP) : <étiquette> = 0x01, n = n_a.
- k_s (clé salée SRTP) : <étiquette> = 0x02, n = n_s.

où n_e, n_s, et n_a sont tirés du contexte cryptographique.

La clé maîtresse et le sel maître DOIVENT être aléatoires, mais le sel maître PEUT être public.

Noter que pour un taux_de_déduction_de_clé de 0, l'application de la déduction de clé DEVRA avoir lieu exactement une fois.

La définition de DIV ci-dessus est une pure convention de notation. Pour un t différent de 0 parmi l'ensemble des taux de déduction de clé admis, "a DIV t" peut être mis en œuvre comme un glissement à droite du logarithme base 2 de t. L'opération de déduction est encore facilitée si les taux sont choisis comme des puissances de 256, mais cette granularité a été considérée comme trop grossière pour que ce soit une exigence de cette spécification.

La limite supérieure sur le nombre de paquets qui peuvent être sécurisés en utilisant la même clé maîtresse (voir au paragraphe 9.2) est indépendante de la déduction de clé.

4.3.2 Déduction de la clé SRTCP

SRTCP DEVRA par défaut utiliser la même clé maîtresse (et sel maître) que SRTP. Pour faire cela en toute sécurité, les changements suivants DEVRONT être faits aux définitions du paragraphe 4.3.1 lors de l'application de la déduction de clé de session pour SRTCP.

Remplacer l'indice SRTP par la quantité de 32 bits : 0 || indice SRTCP (c'est-à-dire, excluant le bit E, le remplaçant par un 0 bit fixe) et en utilisant <étiquette> = 0x03 pour la clé de chiffrement SRTCP, <étiquette> = 0x04 pour la clé d'authentification SRTCP, et, <étiquette> = 0x05 pour la clé salée SRTCP.

4.3.3 PRF AES-CM

La PRF actuellement définie, avec pour clé la clé maîtresse à 128, 192, ou 256 bits, a une taille de bloc d'entrée m = 128 et peut produire des résultats de n bits pour n jusqu'à 2^23. PRF_n(k_maître,x) DEVRA être AES en mode compteur comme décrit au paragraphe 4.1.1, appliquée à la clé k_maître, et IV égal à (x*2^16) et avec le flux de clé de sortie tronqué au n premiers bits (les plus à gauche). (Exigeant n/128, arrondi, applications de AES.)

5. Transformations par défaut et de mise en œuvre obligatoire

Les transformations par défaut sont aussi des transformations de mise en œuvre obligatoire dans SRTP. Bien sûr, "de mise en œuvre obligatoire" n'implique pas "d'utilisation obligatoire". Le Tableau 1 résume les transformations prédéfinies. Les valeurs par défaut ci-dessous sont valides pour les transformations prédéfinies.

	mise en œuvre obl.	**facultative**	**par défaut**
chiffrement	AES-CM, NUL	AES-f8	AES-CM
intégrité du message	HMAC-SHA1	-	HMAC-SHA1
déduction de clé (PRF)	AES-CM	-	AES-CM

Tableau 1 : Transformations de mise en œuvre obligatoire, facultative et par défaut dans SRTP et SRTCP.

5.1 Chiffrement : AES-CM et NUL

AES fonctionnant en mode compteur d'entier segmenté, comme défini au paragraphe 4.1.1, DEVRA être l'algorithme de chiffrement par défaut. Les longueurs de clé par défaut DEVRONT être 128 bits pour la clé de chiffrement de session (n_e). La longueur de clé de sel de session par défaut (n_s) DEVRA être 112 bits.

Le chiffrement NUL DEVRA aussi être de mise en œuvre obligatoire.

5.2 Authentification/Intégrité de message : HMAC-SHA1

HMAC-SHA1, comme défini au paragraphe 4.2.1, DEVRA être le code d'authentification de message par défaut. La

longueur de clé d'authentification de session par défaut (n_a) DEVRA être 160 bits, la longueur d'étiquette d'authentification par défaut (n_tag) DEVRA être 80 bits, et la SRTP_PREFIX_LENGTH DEVRA être zéro pour HMAC-SHA1. De plus, pour SRTCP, le HMAC-SHA1 prédéfini NE DOIT PAS être appliqué avec une valeur de n_tag, ni de n_a, qui soient inférieures à ces valeurs par défaut. Pour SRTP, de plus petites valeurs ne sont PAS RECOMMANDÉES, mais PEUVENT être utilisées après un examen attentif des problèmes soulevés aux paragraphes 7.5 et 9.5.

5.3 Déduction : AES-CM et PRF

La déduction de clé AES fondée sur le mode compteur et la PRF définies aux paragraphes 4.3.1 à 4.3.3, utilisant une clé maîtresse de 128 bits, DEVRONT être la méthode par défaut pour générer les clés de session. La longueur de sel maître par défaut DEVRA être 112 bits et le taux de déduction de clé par défaut DEVRA être zéro.

6. Ajout des transformations SRTP

La Section 4 donne des exemples du niveau de détail nécessaire pour définir les transformations. Chaque fois qu'une nouvelle transformation est à ajouter à SRTP, une RFC d'accompagnement en cours de normalisation DOIT être rédigée pour définir exactement comment la nouvelle transformation peut être utilisée avec SRTP (et SRTCP). Une telle RFC d'accompagnement DEVRAIT éviter de se chevaucher avec le document qui spécifie le protocole SRTP. Noter cependant, qu'il PEUT être nécessaire d'étendre la définition du contexte cryptographique SRTP ou SRTCP avec de nouveaux paramètres (incluant des valeurs fixes ou par défaut) d'ajouter des étapes au traitement du paquet, ou même d'ajouter des champs aux paquets SRTP/SRTCP. La RFC d'accompagnement DEVRA expliquer tous problèmes connus concernant les interactions entre les transformations et les autres aspects de SRTP.

Chaque nouveau document de transformation DEVRAIT spécifier ses attributs de clés, par exemple, la taille des clés (minimum, maximum, recommandée) le format des clés, le traitement recommandé/exigé du matériel de clé d'entrée, les exigences/recommandations sur la durée de vie des clés, le changement de clés et la déduction des clés, et si le partage de clés entre SRTP et SRTCP est permis ou non, etc.

Un ajout de transformation d'intégrité de message DEVRAIT définir un minimum acceptable de taille de clé/étiquette pour SRTCP, équivalent en force aux valeurs minimum définies au paragraphe 5.2.

7. Raisons

La présente section explique les raisons qui sont derrière plusieurs dispositifs importants de SRTP.

7.1 Déduction de clé

La déduction de clé réduit la charge d'établissement des clés. Six clés différentes sont nécessaires par crypto contexte (clés et sels de chiffrement SRTP et SRTCP, clés d'authentification SRTP et SRTCP) mais elles sont déduites d'une seule clé maîtresse d'une façon cryptographiquement sûre. Donc, le protocole de gestion de clé a seulement besoin d'échanger une clé maîtresse (plus le sel maître lorsque nécessaire) et ensuite SRTP lui-même déduit toutes les clés de session nécessaires (via la première fonction d'application obligatoire de déduction de clé).

Plusieurs applications de la fonction de déduction de clé sont facultatives, mais ne donneront le bénéfice de la sécurité que lorsque elles sont activées. Elles empêchent un attaquant d'obtenir de grandes quantités de texte chiffré produit par une seule clé de session fixée. Si l'attaquant était capable de collecter une grande quantité de texte chiffré pour une certaine clé de session, cela pourrait l'aider à monter certaines attaques.

Plusieurs applications de la fonction de déduction de clé fournissent la sécurité vers l'avant et vers l'arrière en ce sens qu'une clé de session compromise ne compromet pas les autres clés de session déduites de la même clé maîtresse. Cela signifie que l'attaquant qui est capable de récupérer une certaine clé de session n'est de toutes façons pas capable d'avoir accès aux messages sécurisés sous les clés de session précédentes et suivantes (déduites de la même clé maîtresse). (Noter que, bien sûr, une fuite de la clé maîtresse révèle toutes les clés de session qui en sont déduites.)

Des problèmes surviennent avec les forts taux de rafraîchissement de clé, en particulier dans les grands établissements de diffusion groupée ; voir la Section 11.

7.2 Clé salée

Le sel maître garantit la sécurité contre les attaques de collision de clés hors ligne sur la déduction de clé qui pourraient autrement réduire la taille effective de clé [MF00].

La clé de session salée utilisée pour le chiffrement a été introduite pour protéger contre certaines attaques sur les chiffrements de flux additifs ; voir au paragraphe 9.2. La méthode d'inclusion explicite du sel dans l'IV a été choisie pour faciliter la mise en œuvre matérielle.

7.3 Intégrité de message à partir d'un hachage universel

La définition particulière du flux de clés donnée au paragraphe 4.1 (le préfixe de flux de clés) est destinée à fournir les fonctions particulières de hachage universel, convenables pour l'authentification de message dans le paradigme de Wegman-Carter [WC81]. De telles fonctions sont d'une sûreté prouvable, simpled, rapided, et particulièrement appropriées pour les processeurs de signaux numériques et autres processeurs avec des opérations de multiplication rapide.

Aucune transformation d'authentification n'est actuellement fournie dans SRTP autre que HMAC-SHA1. De futures transformations, comme les fonctions de hachage universel susmentionnées, PEUVENT être ajoutées suivant les lignes directrices de la Section 6.

7.4 Considérations sur l'authentification de l'origine des données

Noter que dans les communications par paires, la protection de l'intégrité et l'authentification de l'origine des données sont fournies ensemble. Cependant, dans les scénarios de groupe où les clés sont partagées entre les membres, l'étiquette MAC prouve seulement qu'un membre du groupe a envoyé le paquet, mais ne protège pas contre un membre qui se fait passer pour un autre. L'authentification de l'origine des données (DOA, *Data Origin Authentication*) pour les sessions de diffusion groupée et RTP de groupe est un problème difficile qui réclame une solution ; bien que des propositions prometteuses aient été explorées [PCST1], [PCST2], il faut encore les travailler pour une spécification rigoureuse de ces technologies. Donc, l'authentification de l'origine des données SRTP dans les groupes fera l'objet d'un complément d'étude.

La DOA peut être faite autrement en utilisant les signatures. Cependant, cela a un fort impact en termes de bande passante et en temps de traitement, donc, nous n'offrons pas cette forme d'authentification dans la transformation d'intégrité de paquet prédéfinie.

La présence de mélangeurs et traducteurs ne permet pas l'authentification de l'origine des données dans le cas où la charge utile RTP et/ou l'en-tête RTP sont manipulés. Noter que ces types d'entités médiatrices interrompent aussi la confidentialité de bout en bout (car la formation de l'IV dépend, par exemple, de la préservation de l'en-tête RTP). Un certain modèle de confiance peut choisir de faire confiance aux mélangeurs/traducteurs pour déchiffrer/rechiffrer le support (cela impliquerait de casser la sécurité de bout en bout, avec des implications sur la sécurité).

7.5 Authentification de message courte et de longueur zéro

Comme on l'a montré à la Figure 1, l'étiquette d'authentification est RECOMMANDÉE dans SRTP. Une pleine étiquette d'authentification de 80 bits DEVRAIT être utilisée, mais une étiquette plus courte ou même une étiquette de longueur zéro (c'est-à-dire, pas d'authentification de message) PEUT être utilisée dans certaines conditions pour prendre en charge l'un ou l'autre des deux environnements d'application suivants.

1. Une authentification forte peut être impraticable dans des environnements où la préservation de la bande passante est impérative. Un cas particulier important est celui des systèmes de communication sans fils, dans lesquels la bande passante est une ressource rare et coûteuse. Des études ont montré que pour certaines applications et technologies de liaisons, des octets supplémentaires peuvent résulter en une diminution significative de l'efficacité du spectre [SWO]. Des efforts considérables ont été faits pour concevoir des techniques de compression d'en-tête IP pour améliorer l'efficacité du spectre [RFC3095]. Une application vocale typique produit des échantillons de 20 octets, et les en-têtes RTP, UDP et IP ont besoin d'être compressés conjointement à un ou deux octets en moyenne afin d'obtenir une économie de bande passante sans fils acceptable [RFC3095]. Dans ce cas, une authentification forte imposerait une surcharge de presque cinquante pour cent.

2. L'authentification est impraticable pour les applications qui utilisent des liaisons de données avec des champs de largeur fixe qui ne peuvent pas s'accommoder de l'expansion due à l'étiquette d'authentification. C'est le cas pour certains importants canaux sans fils existants. Par exemple, la compression d'en-tête de zéro octet est utilisée pour

adapter la voix EVRC/SMV au canal support IS-95 traditionnel dans les services VoIP CDMA2000. Il a été trouvé que pas un seul octet supplémentaire ne pouvait être ajouté aux données, ce qui a motivé la création du profil zéro octet pour ROHC [RFC3242].

Une étiquette courte est sûre pour un ensemble restreint d'applications. Considérons par exemple une application de téléphonie vocale, telle q u'un codec audio G.729 avec un intervalle de mise en paquet de 20 millisecondes, protégé par une étiquette d'authentification de message de 32 bits. La probabilité qu'un paquet donné soit falsifié avec succès est seulement de un sur 2^32. Donc un adversaire ne peut pas contrôler plus de 20 millisecondes d'une sortie audio durant une période de 994 jours, en moyenne. À l'opposé, l'effet d'un seul paquet falsifié peut être bien plus grand si l'application est à états pleins. Un codec qui utilise une compression relative ou prédictive sur les paquets va propager l'état malveillant généré, affectant une plus longue durée du résultat.

Certainement pas toutes les applications SRTP ou de téléphonie ne satisfont aux critères pour les étiquettes d'authentification courtes ou de longueur zéro. Le paragraphe 9.5.1 expose les risques d'une authentification de message faible ou absente, et le paragraphe 9.5 décrit les circonstances dans lesquelles elle est acceptable et quand elle ne l'est pas.

8. Considérations sur la gestion des clés

De nouvelles normes de gestion de clé [RFC3830], [RFC4567], [RFC4568] apparaissent pour l'établissement d'un contexte cryptographique SRTP (par exemple, une clé maîtresse SRTP). Des méthodes aussi bien brevetées que libres de gestion de clé seront vraisemblablement utilisées pour les applications de téléphonie [RFC3830], [RFC4430] et des applications de diffusion groupée [RFC3547]. La présente section donne des lignes directrices sur les systèmes de gestion de clé qui desservent la session SRTP.

Pour l'initialisation, une mise en œuvre SRTP interopérable DEVRAIT recevoir le SSRC et PEUT recevoir le numéro de séquence initial RTP pour le flux RTP par la gestion de clé (donc, la gestion de clé est dépendante des paramètres opérationnels de RTP). L'envoi du numéro de séquence RTP dans la gestion de clé peut être utile par exemple, lorsque le numéro de séquence initial est proche du retour à zéro (pour éviter les problèmes de synchronisation) et pour communiquer le numéro de séquence actuel à un point d'extrémité qui rejoint la session (pour initialiser correctement sa liste de répétitions).

Si les transformations prédéfinies sont utilisées, SRTP permet le partage de la même clé maîtresse entre les flux SRTP/SRTCP qui appartiennent à la même session RTP.

D'abord, le partage entre des flux SRTP appartenant à la même session RTP est sûr si la conception du mécanisme de synchronisation, c'est-à-dire, de l'IV, évite la réutilisation des flux de clés (le bourrage répété, paragraphe 9.1). Ceci est pris en charge par le fait que RTP fournit des SSRC uniques pour les flux qui appartiennent à la même session RTP. Voir les détails au paragraphe 9.1.

Ensuite, le partage entre SRTP et le SRTCP correspondant est sûr. Le fait qu'un flux SRTP et son flux SRTCP associé portent tous deux le même SSRC ne constitue pas un problème pour le bourrage répété du fait de la déduction de clé. Donc, SRTP et SRTCP correspondant à une session RTP PEUVENT partager des clés maîtresses (comme ils le font par défaut).

Noter que l'authentification de message a aussi une dépendance à l'unicité de la SSRC qui est sans relation avec le problème de la réutilisation du flux de clés : le flux SRTP authentifié sous la même clé DOIT avoir une SSRC distincte afin d'identifier l'envoyeur du message. Cette exigence est nécessaire parce que la SSRC est le champ cryptographiquement authentifié qui est utilisé pour distinguer entre les différents flux SRTP. Si deux flux devaient utiliser des valeurs de SSRC identiques, un adversaire pourrait alors substituer des messages d'un flux à un autre sans détection.

SRTP/SRTCP NE DOIT PAS partager les clés maîtresses, dans aucune autres circonstances que celles citées ci-dessus, c'est-à-dire, entre SRTP et son SRTCP correspondant, et entre des flux appartenant à la même session RTP.

8.1 Changement de clé

La façon recommandée pour un système de gestion de clé particulier de fournir le changement de clé au sein de SRTP est en associant une clé maîtresse dans un contexte cryptographique à un MKI.

Cela permet une restitution facile de la clé maîtresse (voir les scénarios à la Section 11) mais a l'inconvénient d'ajouter des bits supplémentaires à chaque paquet. Comme noté au paragraphe 7.5, certaines liaisons sans fils ne se chargent pas des bits ajoutés, donc SRTP définit aussi une façon plus économique de déclancher le changement de clé, via l'utilisation de <From, To>, qui fonctionne dans des scénarios spécifiques simples (voir au paragraphe 8.1.1).

Les envoyeurs SRTP DEVRONT compter la quantité de trafic SRTP et SRTCP utilisé pour une clé maîtresse et invoquer la gestion de clé pour changer de clé si nécessaire (paragraphe 9.2). Ces interactions sont définies par l'interface de gestion de clé avec SRTP et ne sont pas définies par la présente spécification de protocole.

8.1.1 Utilisation de <From, To> pour le changement de clé

En plus de l'utilisation de MKI, SRTP définit un autre mécanisme facultatif pour la restitution de la clé maîtresse, le <From, To>. Le <From, To> spécifie la gamme des indices SRTP (une paire d'un numéro de séquence et d'un ROC) au sein de laquelle une certaine clé maîtresse est valide, et fait partie (lorsque elle est utilisée) du contexte cryptographique. En regardant l'indice SRTP de 48 bits du paquet SRTP en cours, la clé maîtresse correspondante peut être trouvée en déterminant à quel intervalle From-To elle appartient. Pour SRTCP, l'indice SRTP le plus récemment observé/utilisé (qui peut être obtenu du contexte cryptographique) est utilisé à cette fin, même si SRTCP a son propre indice (de 31 bits) (voir l'avertissement ci-dessous).

Cette méthode, comparée à celle du MKI, présente l'avantage d'identifier la clé maîtresse et de définir sa durée de vie sans ajouter de bit supplémentaire à chaque paquet. Ceci pourrait être utile, comme on l'a déjà noté, pour certaines liaisons sans fils que ne prennent pas en charge les bits ajoutés. Cependant, son utilisation DEVRAIT être limitée à des scénarios spécifiques, très simples. On recommande de limiter son usage à des sessions RTP qui sont un flux unidirectionnel ou bidirectionnel simple. Cela parce que en cas de flux multiples, il est difficile de déclancher le changement de clé sur la base du <From, To> d'un seul flux RTP. Par exemple, si plusieurs flux partagent une clé maîtresse, il n'y a pas de simple correspondance bijective entre l'espace de séquence d'indice d'un certain flux, et l'espace de séquence d'indice sur lequel les valeurs de <From, To> se fondent. Par conséquent, lorsque une clé maîtresse est partagée entre des flux, un de ces flux DOIT être désigné par la gestion de clé comme celui dont l'espace d'indice définit les points de changement de clé. Aussi, le déclanchement du changement de clé sur SRTCP se fonde sur le flux SRTP correspondant, c'est-à-dire que lorsque le flux SRTP change la clé maîtresse, ainsi fait le SRTCP correspondant. Cela devient évidemment de plus en plus complexe avec des flux multiples.

Les valeurs par défaut pour le <From, To> sont "du premier paquet observé" et "jusqu'à nouvel ordre". Cependant, la limite maximum des paquets SRTP/SRTCP qui sont envoyés sous chaque clé maîtresse/clé de session (paragraphe 9.2) NE DOIT PAS être excédée.

Au cas où le <From, To> est utilisé comme restitution de clé, le MKI n'est alors pas inséré dans le paquet (et son indicateur dans le contexte cryptographique est zéro). Cependant, utiliser le MKI n'exclut pas d'utiliser simultanément la durée de vie de clé de <From, To>. Cela peut, par exemple, être utile pour signaler au côté envoyeur à quel moment un MKI va être rendu actif.

8.2 Paramètres de gestion de clé

Le tableau ci-dessous fait la liste de tous les paramètres SRTP que peut fournir la gestion de clé. Pour référence, il fournit aussi un résumé des valeurs par défaut et de mise en œuvre obligatoire pour une mise en œuvre SRTP comme décrit à la Section 5.

Paramètre	**Prise en charge obligatoire**	**Par défaut**
Transfo. de chiffrement SRTP et SRTCP	AES_CM, NUL	AES_CM
(Autres valeurs possibles : AES_f8)		
Transfo. d'auth RTP et SRTCP	HMAC-SHA1	HMAC-SHA1
Paramètres d'auth. SRTP et SRTCP :		
n_tag (longueur d'étiquette)	80	80
longueur de préfixe SRTP	0	0
PRF de déduction de clé	AES_CM	AES_CM
Paramètres de matériel de clés (pour chaque clé maîtresse) :		
longueur de clé maîtresse	128	128
n_e (longueur de clé de session de chiffrement)	128	128
n_a (longueur de clé de session d'auth.)	160	160
Clé de sel maître		
longueur du sel maître	112	112
n_s (longueur de clé de sel de session)	112	112
taux de déduction de clé	0	0
Durée de vie de clé		
durée de vie max de paquets SRTP	2^48	2^48
durée de vie max de paquets SRTCP	2^31	2^31

durée de vie from-to <From, To>		
indicateur MKI		
longueur du MKI	0	0
valeur du MKI	0	0

Paramètres d'indice de contexte cryptographique :
- valeur de SSRC
- ROC
- SEQ
- Indice SRTCP
- Adresse de transport
- Numéro d'accès

Relation à d'autres profils RTP :		
Ordre des envoyeurs entre FEC et SRTP (voir la Section 10)	FEC-SRTP	FEC-SRTP

9. Considérations pour la sécurité

9.1 Collision de SSRC et bourrage répété

Tout résultat de flux de clés fixe, généré à partir des mêmes clés et indice DOIT n'être utilisé que pour un seul chiffrement. La réutilisation d'un tel flux de clé (appelé ironiquement un système de "bourrage répété" par les cryptographeurs) peut sérieusement compromettre la sécurité. Le projet VENONA de la NSA [C99] a fournit un exemple historique d'une telle compromission. Il est EXIGÉ que la gestion automatique de clé soit utilisée pour l'établissement et l'entretien du matériel de clés SRTP et SRTCP ; cette exigence est destinée à éviter la réutilisation du flux de clés, qui va plus probablement survenir avec la gestion manuelle de clé. De plus, dans SRTP, un "bourrage répété" est évité en exigeant que la clé, ou d'autres paramètres significatif du chiffrement, soit unique par flux et paquet RTP/RTCP. Les transformations SRTP prédéfinies réalisent l'unicité par paquet en incluant l'indice du paquet et l'unicité du flux en incluant le SSRC.

Les transformations prédéfinies (AES-CM et AES-f8) permettent que les clés maîtresses soient partagées à travers les flux qui appartiennent à la même session RTP par l'inclusion du SSRC dans l'IV. Une clé maîtresse NE DOIT PAS être partagée par des sessions RTP différentes.

Donc, le SSRC DOIT être unique pour tous les flux RTP au sein de la même session RTP qui partagent la même clé maîtresse. RTP fournit lui-même un algorithme pour détecter les collisions de SSRC au sein de la même session RTP. Donc, des collisions temporaires pourraient conduire à des bourrages répétés temporaires, dans le cas malencontreux où les SSRC se percuteraient à un moment où les flux auraient aussi des numéros de séquence identiques (ce qui se produit avec une probabilité d'environ 2^(-48)). Donc, la gestion de clé DEVRAIT veiller à éviter de telles collisions de SSRC en incluant les SSRC à utiliser dans la session comme paramètres de négociation, assurant de façon proactive leur unicité. C'est une exigence forte dans les scénarios où par exemple, plusieurs envoyeurs peuvent commencer à transmettre simultanément, avant que la collision de SSRC ne soit détectée au niveau RTP.

Noter aussi que même avec des SSRC distincts, une utilisation extensive de la même clé peut augmenter la probabilité de collision et de réussite d'attaques de compromis temps/mémoire.

Comme on l'a décrit, les clés maîtresses PEUVENT être partagées entre des flux qui appartiennent à la même session RTP, mais il est RECOMMANDÉ que chaque SSRC ait sa propre clé maîtresse. Lorsque les clés maîtresses sont partagées entre les participants SSRC et que les SSRC sont gérés par un module de gestion de clé comme recommandé ci-dessus, la politique RECOMMANDÉE pour une erreur de collision de SSRC est que le participant quitte la session SRTP car c'est un signe de dysfonctionnement

9.2 Usage des clés

La taille de clé effective est déterminée (avec une limite supérieure) par la taille de la clé maîtresse et, pour le chiffrement, par la taille de la clé salée. Tout chiffrement de flux ajouté est vulnérable aux attaques qui utilisent une connaissance statistique sur la source de texte en clair pour rendre possible une collision de clé et des attaques de compromis temps-mémoire [MF00], [H80], [BS00]. Ces attaques tirent parti des redondances dans le texte en clair, et donnent un moyen au cryptanalyste d'atténuer l'effort de calcul du déchiffrement sur de nombreuses clés, ou sur de nombreux octets de résultat, réduisant donc la taille effective de la clé de chiffrement. Une analyse détaillée de ces attaques et de leur applicabilité au chiffrement du trafic Internet est fournie dans [MF00]. En résumé, la taille effective de clé de SRTP lorsque utilisé dans un

système de sécurité dans lequel m clés distinctes sont utilisées, est égale à la taille de clé du chiffre moins le logarithme (base deux) de m. La protection contre de telles attaques peut être fournie simplement en augmentant la taille des clés utilisées, ce qui peut être accompli ici par l'utilisation de la clé salée. Noter que la clé salée DOIT être aléatoire mais PEUT être publique. Une taille de sel (suggérée) de 112 bits protège contre les attaques dans les scénarios où au plus 2^112 clés sont utilisées. Ceci est suffisant pour tous les besoins pratiques.

Les mises en œuvres DEVRAIENT utiliser les clés qui sont aussi longues que possible. Prière de noter que dans de nombreux cas, augmenter la taille de clé d'un chiffrement n'affecte pas le débit de ce chiffrement.

L'utilisation des indices SRTP et SRTCP dans les transformations prédéfinies fixe le nombre maximum de paquets qui peuvent être sécurisés avec la même clé. Cette limite est fixée à 2^48 paquets SRTP pour un flux SRTP, et à 2^31 paquets SRTCP, lorsque SRTP et SRTCP sont considérés indépendamment. Du fait, par exemple du changement de clé, atteindre cette limite peut ou non coïncider avec le retour à zéro des indices, et donc, l'envoyeur DOIT tenir le compte des paquets. Cependant, lorsque les clés de session pour les flux SRTP et SRTCP en rapport sont déduites de la même clé maîtresse (comportement par défaut du paragraphe 4.3) la limite supérieure qui doit être considérée est en pratique le minimum de deux quantités. C'est-à-dire, lorsque 2^48 paquets SRTP ou 2^31 paquets SRTCP ont été sécurisés avec la même clé (quel que soit le premier qui survient) la gestion de clé DOIT être invoquée pour fournir de nouvelles clés maîtresses (précédemment mémorisées et les clés utilisées NE DOIVENT PAS être réutilisées) ou la session DOIT être terminée. Si un envoyeur de RTCP découvre que l'envoyeur de SRTP (ou SRTCP) n'a pas mis à jour la clé maîtresse ou la clé de session avant d'envoyer 2^48 paquets SRTP (ou 2^31 paquets SRTCP) appartenant au même flux SRTP (SRTCP) il appartient à la politique de sécurité de l'envoyeur RTCP de décider de la conduite à tenir, par exemple, si un paquet BYE RTCP devrait être envoyé et/ou si l'événement devrait être enregistré dans le journal d'incidents.

Note : Dans la plupart des applications normales (supposant au moins un paquet RTCP tous les 128 000 paquets RTP) ce sera l'indice SRTCP qui atteindra le premier la limite supérieure, bien que le temps jusqu'à ce que cela arrive soit très long : même à 200 paquets SRTCP/s, l'espace d'indice de 2^31 de SRTCP est suffisant pour sécuriser approximativement 4 mois de communication.

Noter que si la clé maîtresse doit être partagée entre les flux SRTP au sein de la même session RTP (paragraphe 9.1) bien que les limites ci-dessus soient flux par flux (c'est-à-dire, par SSRC) l'envoyeur DOIT fonder sa décision de changement de clé sur les flux dont l'espace de numéro de séquence est le premier à être épuisé.

La déduction de clé limite la quantité de texte en clair qui est chiffré avec une clé de session fixe, et rend l'analyse possible à un attaquant, mais la déduction de clé n'étend pas la durée de vie de la clé maîtresse. Pour voir cela, considérons simplement notre exigence d'éviter le bourrage répété : deux paquets distincts DOIVENT être traités soit avec des IV distincts, soit avec des clés de session distinctes, et la distinction des IV et des clés de session sont toutes deux (pour les transformations prédéfinies) dépendantes de la distinction des indices de paquet.

Noter qu'avec la déduction de clé, la taille effective de clé est au plus celle de la clé maîtresse, même si la clé de session déduite est considérablement plus longue. Avec la transformation d'authentification prédéfinie, la clé d'authentification de session est de 160 bits, mais la clé maîtresse par défaut est seulement de 128 bits. Ce choix de conception a été fait pour se plier à certaines recommandations de la [RFC2104] afin qu'une mise en œuvre HMAC existante puisse être insérée sans problème dans SRTP. Comme la taille d'étiquette est par défaut de 80 bits, elle est, pour les applications visées, aussi considérée comme acceptable du point de vue de la sécurité. Aux utilisateurs qui ont des problèmes avec cela, il est RECOMMANDÉ d'utiliser plutôt une clé maîtresse de 192 bits dans la déduction de clé. Il a cependant été choisi de ne pas rendre obligatoire la clé de 192 bits car les mises en œuvre AES existantes à utiliser pour la déduction de clé peuvent ne pas toujours prendre en charge des longueurs de clé autres que de 128 bits. Comme AES n'est pas défini (ou proprement analysé) pour être utilisé avec des clés de 160 bits, il n'est PAS RECOMMANDÉ d'utiliser des schémas de bourrage de clé ad hoc pour bourrer de plus courtes clés jusqu'à 192 ou 256 bits.

9.3 Confidentialité de la charge utile RTP

Les chiffrements prédéfinis de SRTP sont des chiffrements de flux "cherchables", c'est-à-dire, des chiffrements capables de chercher efficacement des localisations arbitraires dans leur flux de clés (de sorte que le chiffrement ou le déchiffrement d'un paquet ne dépende pas des paquets précédents). En utilisant des chiffrements de flux cherchables, SRTP évite les attaques de déni de service qui sont possibles sur les chiffrements de flux qui n'ont pas cette propriété. Il est important d'être conscient que, comme avec tout chiffrement de flux, la longueur exacte de la charge utile est révélée par le chiffrement. Cela signifie qu'il est possible de déduire certains "bits de formatage" de la charge utile, car la longueur de la sortie du codec pourrait varier du fait de certains réglages de paramètres, etc. Cela implique à son tour que le bit correspondant du flux de clé peut être déduit. Cependant, si le chiffrement de flux est sûr (les modes compteur et f8 sont d'une sûreté prouvable sous certaines conditions [BDJR], [KSYH], [IK]) ; la connaissance de quelques bits du flux de clés ne va pas aider un attaquant à prédire les bits suivants du flux de clés. Donc, la longueur de la charge utile (et les

informations qu'on peut en déduire) va être divulguée, mais rien de plus.

Comme certains paquets RTP pourraient contenir des données très prévisibles, par exemple, le SID, il est important d'utiliser un chiffrement conçu pour résister aux attaques de texte en clair connu (qui est de pratique courante).

9.4 Confidentialité de l'en-tête RTP

Dans SRTP, les en-têtes RTP sont envoyés en clair pour permettre la compression d'en-tête. Cela signifie que des données comme le type de charge utile, l'identifiant de source de synchronisation, et l'horodatage sont disponibles à l'espionnage. De plus, comme RTP permet de futures extensions d'en-têtes, on ne peut pas prévoir quelles informations éventuellement sensibles pourraient aussi subir des "fuites".

SRTP est une méthode peu coûteuse, qui permet à la compression d'en-tête de réduire la bande passante. Il appartient aux politiques des points d'extrémité de décider des protocoles de sécurité à employer. Si on a réellement besoin de protéger les en-têtes, et il est permis de le faire par l'environnement, on devrait aussi examiner les solutions de remplacement, par exemple, IPsec [RFC2401].

9.5 Intégrité de l'en-tête et de la charge utile RTP

Les messages SRTP sont l'objet d'attaques contre leur intégrité et l'identification de source, et ces risques sont discutés au paragraphe 9.5.1. Pour se protéger contre ces attaques, chaque flux SRTP DEVRAIT être protégé par HMAC-SHA1 [RFC2104] avec une étiquette de sortie de 80 bits et une clé de 160 bits, ou un code d'authentification de message avec une force équivalente. RTP sécurisé NE DEVRAIT PAS être utilisé sans l'authentification de message, sauf dans les circonstances décrites dans cette section. Il est important de noter que les algorithmes de chiffrement, y compris AES en mode compteur et f8, n'assurent pas l'authentification de message. SRTCP NE DOIT PAS être utilisé avec une authentification faible (ou NULLE) .

SRTP PEUT être utilisé avec une authentification faible (par exemple, une étiquette d'authentification de 32 bits) ou sans authentification (l'algorithme d'authentification NUL). Ces options permettent à SRTP d'être utilisé pour fournir la confidentialité dans des situations où :

* une authentification faible ou nulle est un risque de sécurité acceptable, et
* il est impraticable de fournir une authentification de message forte.

Ces conditions sont décrites ci-dessous et au paragraphe 7.5. Noter que les deux conditions DOIVENT tenir pour que l'authentification faible ou nulle soit utilisée. Les risques associés à la mise en œuvre des options d'authentification faible ou nulle doivent être examinés par un audit de sécurité avant leur utilisation pour une application ou environnement particuliers étant données les risques, qui sont exposés au paragraphe 9.5.1.

L'authentification faible est acceptable lorsque l'application RTP est telle que l'effet d'une petite fraction de falsification réussie est négligeable. Si l'application est sans état, alors, l'effet d'un seul paquet RTP falsifié est limité au décodage de ce paquet particulier. Dans ces conditions, la taille de l'étiquette d'authentification DOIT assurer que seule une fraction négligeable du paquets passé à l'application RTP par le receveur SRTP peut être falsifiée. Cette fraction est négligeable lorsque un adversaire, si on lui donne le contrôle des paquets falsifiés, n'est pas capable d'avoir un impact significatif sur le résultat de l'application RTP (voir l'exemple du paragraphe 7.5).

Une authentification faible ou nulle PEUT être acceptable lorsque il est peu probable qu'un adversaire puisse modifier le texte chiffré de telle sorte que son déchiffrement donne une valeur intelligible. Un cas important est lorsque il est difficile à un adversaire d'acquérir les données du texte RTP en clair, car pour de nombreux codecs, un adversaire qui ne connaît pas le signal d'entrée ne peut pas manipuler le signal de sortie d'une façon contrôlée. Dans de nombreux cas, il peut être difficile à l'adversaire de déterminer la valeur réelle du texte en clair. Par exemple, un appareil d'espionnage dissimulé peut être nécessaire pour connaître un signal audio ou vidéo en direct. Le signal de l'adversaire doit avoir une qualité équivalente ou supérieure à celle du signal attaqué, car autrement, l'adversaire n'aurait pas assez d'informations pour coder ce signal avec le codec utilisé par la victime. La prédiction du texte en clair peut aussi être particulièrement difficile pour une application interactive comme un appel téléphonique.

Une authentification faible ou nulle NE DOIT PAS être utilisée lorsque l'application RTP prend des décisions de transmission de données ou de contrôle d'accès sur la base des données RTP. Dans de tels cas, un attaquant peut être capable de subvertir la confidentialité en causant la transmission des données par le receveur à un attaquant. Voir à la Section 3 de [B96] une exemple réel de telles attaques.

L'authentification nulle NE DOIT PAS être utilisée lors d'une attaque en répétition, dans laquelle un adversaire mémorise les paquets puis les répète plus tard dans la session, qui pourrait avoir un impact non négligeable sur le receveur. Un

exemple d'une attaque en répétition réussie est la mémorisation du résultat d'une caméra de surveillance pendant un certain temps, suivie plus tard par l'injection de ce résultat à la station de surveillance pour éviter la surveillance. Le chiffrement ne protège pas contre cette attaque, et une authentification non nulle est EXIGÉE afin de la déjouer.

Si la falsification de messages existentiels est un problème, c'est-à-dire, lorsque la précision des données reçues est d'une importance non négligeable, l'authentification nulle NE DOIT PAS être utilisée.

9.5.1 Risques d'authentification de message faible ou nulle

Durant un audit de sécurité pour considérer la possibilité d'utilisation d'une authentification faible ou nulle, il est important de garder présent à l'esprit les attaques suivantes qui sont possibles lorsque aucun algorithme d'authentification de message n'est utilisé.

Un attaquant qui ne peut pas prédire le texte en clair est quand même capable de modifier le message envoyé entre l'envoyeur et le receveur de sorte qu'il déchiffre en une valeur de texte clair aléatoire, ou qu'il envoie un flux de paquets bogués au receveur qui va déchiffrer des valeurs de texte clair aléatoires. Cette attaque est essentiellement un déni de service, bien qu'en l'absence de l'authentification de message, l'application RTP va avoir des entrées qui sont corrélées au bit près à la vraie valeur. Certains codecs multimédia et des systèmes d'exploitation courants vont tomber en panne lorsque de telles données sont acceptées comme données vidéo valides. Cette attaque de déni de service peut être une bien plus grande menace que celle d'un attaquant qui élimine, retarde ou dérange l'ordre des paquets.

Un attaquant qui ne peut pas prédire le texte en clair peut toujours répéter un message précédent avec la certitude que le receveur va l'accepter. Les applications qui ont des codecs sans état peuvent être robustes contre ce type d'attaque, mais pour les autres applications plus complexes, ces attaques peuvent être beaucoup plus graves.

Un attaquant qui peut prédire le texte en clair peut modifier le texte chiffré de telle sorte qu'il se déchiffre en une valeur de son choix. Avec un chiffrement de flux additif, un attaquant va toujours être capable de changer des bits individuels.

Un attaquant peut être capable de subvertir la confidentialité à cause du manque d'authentification lors de la transmission des données ou de la prise de décision de contrôle d'accès sur du texte en clair déchiffré mais non authentifié. Cela parce que le receveur peut être trompé au point d'envoyer les données à l'attaquant, ce qui conduit à une violation indirecte de la confidentialité (voir la Section 3 de [B96]). Cela parce que les décisions de transmission des données sont prises sur le texte en clair déchiffré ; les informations du texte en clair vont déterminer à quel sous-réseau (ou processus) est transmis le texte en clair en mode tunnel ESP [RFC2401] (respectivement, en mode transport). Lorsque RTP sécurisé est utilisé sans authentification de message, il devrait être vérifié que l'application ne prend pas de décision de transmission des données ou de contrôle d'accès sur la base du texte en clair déchiffré.

Certains modes de fonctionnement de chiffrement qui exigent du bourrage, par exemple, la chaîne de bloc de chiffrement (CBC, *cipher block chaining*) standard, sont très sensibles aux attaques contre la confidentialité si certains types de bourrage sont utilisés en l'absence de protection de l'intégrité. L'attaque [V02] montre que c'est bien sûr le cas pour le bourrage RTP standard, comme exposé dans la référence à la Figure 1, quand il est utilisé avec le mode CBC. Les ajouts de transformations ultérieures à SRTP DOIVENT donc examiner attentivement le risque d'utilisation de ce bourrage sans une protection appropriée de l'intégrité.

9.5.2 Authentification implicite d'en-tête

La formation d'IV du mode f8 donne une authentification implicite (IHA) de l'en-tête RTP, même lorsque l'authentification de message n'est pas utilisée. Lorsque IHA est utilisée, un attaquant qui modifie la valeur de l'en-tête RTP va causer la production de valeurs de texte en clair aléatoires lors du processus de déchiffrement chez le receveur. Bien que cette protection ne soit pas équivalente à l'authentification de message, elle peut être utile à certaines applications.

10. Interaction avec les mécanismes de correction d'erreur directe

Le traitement par défaut lors de l'utilisation de la correction d'erreur directe (FEC, *forward error correction*) (par exemple, RFC2733) avec SRTP DEVRA être d'effectuer un traitement de FEC avant le traitement SRTP du côté envoyeur et d'effectuer un traitement SRTP avant le traitement de FEC sur le côté receveur. Tout changement à cet ordre (en l'inversant, ou en plaçant la FEC entre le chiffrement SRTP et l'authentification SRTP) DEVRA être signalé hors bande.

11. Scénarios

SRTP peut être utilisé comme protocole de sécurité pour le trafic RTP/RTCP dans de nombreux scénarios différents. SRTP a un certain nombre d'options de configuration, en particulier en ce qui concerne l'usage des clés, et peut avoir un impact sur les performances globales de l'application selon la façon dont il est utilisé. Donc, l'utilisation de SRTP dépend du type de scénario et d'application avec lesquels il est utilisé. Dans ce qui suit, on illustre brièvement quelques cas d'utilisation de SRTP, et on donne quelques lignes directrices sur les réglages recommandés de ses options.

11.1 Envoi individuel

Un exemple typique serait celui d'un appel vocal ou d'une application de vidéo à la demande.

Considérons un flux RTP bidirectionnel, comme une session RTP. Il est possible aux deux parties de partager la même clé maîtresse dans les deux directions selon les principes du paragraphe 9.1. Le premier tour de la déduction de clé partage la clé maîtresse en une ou toutes les clés de session suivantes (selon les fonctions de sécurité fournies) :

SRTP_encr_key, SRTP_auth_key, SRTCP_encr_key, et SRTCP_auth key.

(Pour simplifier, on omet la discussion des sels, qui sont aussi déduits.) Dans ce scénario, il va suffire dans la plupart des cas d'avoir une seule clé maîtresse avec la durée de vie par défaut. Cela garantit une durée de vie suffisamment longue des clés et un ensemble minimum de clés en place pour la plupart des besoins pratiques. Aussi, dans ces cas, la protection de RTCP peut s'appliquer en douceur. Avec ces hypothèses, l'utilisation du MKI peut être omise. Comme la déduction de clés en combinaison avec de grosses différences dans les taux de paquet dans les directions respectives peut exiger la mémorisation simultanée de plusieurs clés de session, si la mémorisation pose un problème, on recommande d'utiliser une déduction de clé à taux faible.

Les mêmes considérations peuvent être étendues au scénario d'envoi individuel avec plusieurs sessions RTP, où chaque session aurait une clé maîtresse distincte.

11.2 Diffusion groupée (un envoyeur)

Tout comme avec RTP (non protégé) un problème d'adaptabilité se pose dans les gros groupes du fait de la possibilité de très grosses quantités de rapports de receveur SRTCP que l'envoyeur peut avoir besoin de traiter. Dans SRTP, l'envoyeur peut devoir conserver l'état (le contexte cryptographique) pour chaque receveur, ou plus précisément, pour le SRTCP utilisé pour protéger les rapports de receveur. La redondance augmente proportionnellement à la taille du groupe. En particulier, le changement de clé exige une attention particulière.

Considérons d'abord un petit groupe de receveurs. Plusieurs réglages sont possibles pour la distribution des clés maîtresses entre les receveurs. Dans une seule session RTP, une possibilité est que les receveurs partagent la même clé maîtresse comme au paragraphe 9.1 pour sécuriser tout leur trafic RTCP respectif. Cette clé maîtresse partagée pourrait alors être la même qu'utilisée par l'envoyeur pour protéger son trafic SRTP sortant. Autrement, ce pourrait être une clé maîtresse partagée seulement parmi les receveurs et utilisée seulement pour leur trafic SRTCP. Les deux solutions exigent que les receveurs se fassent confiance.

Quand on considère la mémorisation des clés dans SRTCP, il est recommandé d'utiliser un taux de déduction faible (ou zéro) (sauf pour l'initial qui est obligatoire) afin que l'envoyeur n'ait pas besoin de stocker trop de clés de session (chaque flux SRTCP pourrait autrement avoir une clé de session différente à un moment donné, car les sources SRTCP envoient à différents moments). Donc, dans les cas où la déduction de clé est voulue pour SRTP, le contexte cryptographique pour SRTP peut être tenu à part du contexte cryptographique de SRTCP, afin qu'il soit possible d'avoir un taux_de_déduction_de_clé de 0 pour SRTCP et une valeur différente de zéro pour SRTP.

L'utilisation du MKI pour le changement de clé est RECOMMANDÉ pour la plupart des applications (voir au paragraphe 8.1).

Si il y a plus d'un flux SRTP/SRTCP (au sein de la même session RTP) qui partagent la clé maîtresse, la limite supérieure de 2^48 paquets SRTP / 2^31 paquets SRTCP signifie que, avant qu'un des flux atteigne son nombre maximum de paquets, le changement de clés DOIT être déclanché sur TOUS les flux qui partagent la clé maîtresse. (Du strict point de vue de la sécurité, seuls les flux qui atteignent le maximum auraient besoin de changer leurs clés, mais alors, les flux ne partageraient plus la clé maîtresse, ce qui est l'intention.) Une politique locale du côté envoyeur devrait forcer le changement de clé de façon que la limite maximum de paquet ne soit atteinte sur aucun des flux. L'utilisation du MKI pour changer les clés est RECOMMANDÉE.

Dans les grosses diffusions groupées avec un seul envoyeur, les mêmes considérations que pour le petit groupe de diffusion groupée tiennent. Le plus gros problème dans ce scénario est la charge supplémentaire qui pèse sur le côté envoyeur, du fait que l'état (des contextes cryptographiques) doit être entretenu pour chaque receveur, à renvoyer les rapports de receveur RTCP. Au minimum, une fenêtre de répétition devrait être insérée pour être tenue pour chaque source RTCP.

11.3 Changement de clé et contrôle d'accès

Le changement d clé peut survenir à cause du contrôle d'accès (par exemple, lorsque un membre est retiré durant une session RTP en diffusion groupée) ou pour de pures raisons cryptographiques (par exemple, la clé est en fin de vie). Lors de l'utilisation des transformations SRTP par défaut, la clé maîtresse DOIT être remplacée avant qu'aucun des espaces d'indice ne soit épuisé pour aucun des flux protégés par la même clé maîtresse.

Comment la gestion de clé change les clés des mises en œuvre SRTP est hors de notre propos, mais il est clair qu'il y a des façons directes de gérer les clés pour un groupe de diffusion groupée. Dans une diffusion groupée à un envoyeur, par exemple, il est normalement de la responsabilité de l'envoyeur de déterminer quand une nouvelle clé est nécessaire. L'envoyeur est l'entité qui peut garder trace de quand le nombre maximum de paquets a été atteint, car les receveurs peuvent se joindre à la session et la quitter à tout moment, il peut y avoir des pertes et des retards de paquet, etc.. Dans les scénarios autres que de diffusion groupée à un seul envoyeur, d'autres méthodes peuvent être utilisées. Ici, on doit prendre en considération que l'échange de clé peut être une opération coûteuse, qui prend plusieurs secondes pour un seul échange. Donc, un certain temps avant que la clé maîtresse ne soit épuisée/n'arrive à expiration, la gestion de clé hors bande est initiée, résultant en une nouvelle clé maîtresse qui est partagée avec le ou les receveurs. En tous cas, pour conserver la synchronisation lorsque on passe à la nouvelle clé, la politique de groupe peut choisir entre utiliser le MKI et le <From, To>, comme décrit au paragraphe 8.1.

Pour les besoins du contrôle d'accès, les périodes de <From, To> sont réglées à la granularité désirée, selon le taux de paquets. Un taux élevé de changement de clés peut être problématique pour SRTCP dans certains scénarios de grands groupes. Comme on l'a mentionné, il y a des problèmes potentiels à utiliser l'indice SRTP, plutôt que l'indice SRTCP, pour déterminer la clé maîtresse. En particulier, pour de courtes périodes durant le changement des clés maîtresses, il se peut que les paquets SRTCP ne soient pas sous la clé maîtresse actuelle du SRTP correspondant. Donc, l'utilisation du MKI pour le changement de clés dans de tels scénarios produira de meilleurs résultats.

11.4 Résumé des scénarios de base

La description de ces scénarios met en lumière certaines recommandations sur l'utilisation de SRTP, principalement en relation avec le changement de clés et la diffusion groupée à grande échelle :

- Ne pas utiliser le changement de clé rapide avec le dispositif <From, To>. Il peut en particulier, causer des problèmes de récupération de la clé SRTCP correcte, si un paquet SRTCP arrive près du moment du changement. Le MKI DEVRAIT être utilisé dans ce cas.
- Si plusieurs flux SRTP dans la même session RTP partagent la même clé maîtresse, modérer aussi le taux de changement de clés PEUT poser les mêmes problèmes, et le MKI DEVRAIT être utilisé.
- Bien qu'offrant une sécurité accrue, un taux_de_déduction_de_clé différent de zéro N'EST PAS RECOMMANDÉ lorsque on essaye de minimiser le nombre de clés utilisées avec plusieurs flux.

12. Considérations relatives à l'IANA

La spécification RTP établit un registre des noms de profils à utiliser par les protocoles de contrôle de niveau supérieur, tels que le protocole de description de session (SDP, *Session Description Protocol*) pour se référer aux méthodes de transport. Le présent profil enregistre le nom "RTP/SAVP".

SRTP utilise des transformations cryptographiques que signale un protocole de gestion de clé. Il relève de chaque protocole de gestion de clé particulier d'enregistrer les transformations cryptographiques ou suites de transformations auprès de l'IANA. Le protocole de gestion de clé porte ces numéros de protocole, non SRTP, et chaque protocole de gestion de clé choisit le schéma et la syntaxe de numérisation qu'il requiert.

La spécification d'un protocole de gestion de clé pour SRTP est hors de propos ici. Cependant, le paragraphe 8.2 donne des lignes directrices sur les paramètres qui ont besoin d'être définis pour les transformations par défaut et obligatoires.

13 Remerciements

David Oran (Cisco) et Rolf Blom (Ericsson) sont co-auteurs du présent document mais leurs précieuses contributions sont mentionnées ici pour garder à la liste des auteurs une taille raisonnable.

Les auteurs souhaitent de plus remercier Magnus Westerlund, Brian Weis, Ghyslain Pelletier, Morgan Lindqvist, Robert Fairlie-Cuninghame, Adrian Perrig, le groupe de travail AVT et en particulier son président Colin Perkins et Stephen Casner, les directeurs de la zone Transport et Sécurité, et Eric Rescorla pour leurs relectures et leur soutien.

14. Références

14.1 Références normatives

[AES] NIST, "Advanced Encryption Standard (AES)", FIPS PUB 197, http://www.nist.gov/aes/

[RFC2104] H. Krawczyk, M. Bellare et R. Canetti, "HMAC : Hachage de clés pour l'authentification de message", février 997.

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

[RFC2401] S. Kent et R. Atkinson, "Architecture de sécurité pour le protocole Internet", novembre 1998. (*Obsolète, voir* RFC4301)

[RFC2828] R. Shirey, "Glossaire de la sécurité sur l'Internet", FYI 36, mai 2000.

[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.

[RFC3551] H. Schulzrinne et S. Casner, "Profil RTP pour conférences audio et vidéo avec contrôle minimal", STD 65, juillet 2003.

14.2 Références pour information

[AES-CTR] Lipmaa, H., Rogaway, P. et D. Wagner, "CTR-Mode Encryption", NIST, http://csrc.nist.gov/encryption/modes/workshop1/papers/lipmaa-ctr.pdf

[B96] Bellovin, S., "Problem Areas for the IP Security Protocols," dans le compte-rendu du sixième symposium Usenix Unix Security, pp. 1-16, San Jose, CA, juillet 1996 http://www.research.att.com/~smb/papers/index.html

[BDJR] Bellare, M., Desai, A., Jokipii, E. et P. Rogaway, "A Concrete Treatment of Symmetric Encryption: Analysis of DES Modes of Operation", Proceedings 38th IEEE FOCS, pp. 394-403, 1997.

[BS00] Biryukov, A. et A. Shamir, "Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers", Proceedings, ASIACRYPT 2000, LNCS 1976, pp. 1-13, Springer Verlag.

[C99] Crowell, W. P., "Introduction to the VENONA Project", http://www.nsa.gov:8080/docs/venona/index.html .

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

[f8-a] 3GPP TS 35.201 V4.1.0 (2001-12) Technical Specification 3rd Generation Partnership Project; "Technical Specification Group Services and System Aspects; 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 1: f8 and f9 Specification" (Release 4).

[f8-b] 3GPP TR 33.908 V4.0.0 (2001-09) Technical Report 3rd Generation Partnership Project; "Technical Specification Group Services and System Aspects; 3G Security; General Report on the Design, Specification and Evaluation of 3GPP Standard Confidentiality and Integrity Algorithms" (Release 4).

[HAC] Menezes, A., Van Oorschot, P. et S. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1997, ISBN 0-8493-8523-7.

[H80] Hellman, M. E., "A cryptanalytic time-memory trade-off", IEEE Transactions on Information Theory, juiller 1980, pp. 401-406.

[IK] T. Iwata et T. Kohno: "New Security Proofs for the 3GPP Confidentiality and Integrity Algorithms", Proceedings of FSE 2004.

[KSYH] Kang, J-S., Shin, S-U., Hong, D. et O. Yi, "Provable Security of KASUMI and 3GPP Encryption Mode f8", Proceedings Asiacrypt 2001, Springer Verlag LNCS 2248, pp. 255-271, 2001.

[MF00] McGrew, D. et S. Fluhrer, "Attacks on Encryption of Redundant Plaintext and Implications on Internet Security", Compte-rendu du septième atelier annuel sur des domaines choisis en cryptographie (SAC 2000), Springer-Verlag.

[PCST1] Perrig, A., Canetti, R., Tygar, D. et D. Song, "Efficient et Secure Source Authentication for Multicast", dans le compte-rendu du Network and Distributed System Security Symposium NDSS 2001, pp. 35-46, 2001.

[PCST2] Perrig, A., Canetti, R., Tygar, D. et D. Song, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels", dans le compte-rendu de IEEE Security and Privacy Symposium S&P2000, pp. 56-73, 2000.

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

[RFC2675] D. Borman, S. Deering, R. Hinden, "Jumbogrammes IPv6", août 1999. *(P.S.)*

[RFC3095] C. Bormann et autres, "Compression d'en-tête robuste (ROHC) : cadre et quatre profils", juillet 2001. (*MàJ par* RFC3759, RFC4815) *(P.S.)*

[RFC3242] L-E. Jonsson, G. Pelletier, "Compression d'en-tête robuste (ROHC) : profil assisté de couche liaison pour IP/UDP/RTP", avril 2002. (*Obsolète, voir* RFC4362) *(P.S.)*

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

[RFC3830] J. Arkko et autres, "MIKEY : Chiffrement Internet multimédia", août 2004. (*MàJ par* RFC4738) *(P.S.)*

[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.)*

[RFC4568] F. Andreasen et autres, "Définition d'attributs de sécurité dans le protocole de description de session (SDP) pour les flux de support", juillet 2006. *(P.S.)*

[SWO] Svanbro, K., Wiorek, J. et B. Olin, "Voice-over-IP-over- wireless", Proc. PIMRC 2000, London, septembre 2000.

[V02] Vaudenay, S., "Security Flaws Induced by CBC Padding -Application to SSL, IPsec, WTLS...", Advances in Cryptology, EUROCRYPT'02, LNCS 2332, pp. 534-545.

[WC81] Wegman, M. N., et J.L. Carter, "New Hash Functions and Their Use in Authentication and Set Equality", JCSS 22, 265-279, 1981.

Appendice A Pseudocode pour la détermination des indices

Ce qui suit est un exemple de pseudo-code pour l'algorithme qui détermine l'indice i d'un paquet SRTP avec le numéro de séquence SEQ. Dans ce qui suit, on suppose une arithmétique signée.

```
if (s_l < 32,768)
  if (SEQ - s_l > 32,768)
    set v to (ROC-1) mod 2^32
  else
```

```
      set v to ROC
    endif
  else
    if (s_l - 32,768 > SEQ)
      set v to (ROC+1) mod 2^32
    else
      set v to ROC
    endif
  endif
  return SEQ + v*65,536
```

Appendice B Vecteurs d'essai

Toutes les valeurs sont en hexadécimal.

B.1 Vecteurs d'essai AES-f8

Longueur de préfixe SRTP : 0

En-tête de paquet RTP : 806e5cba50681de55c621599

Charge utile de paquet RTP :
70736575646f72616e646f6d6e6573732069732074686520e6578742062657374207468696e67

ROC :	d462564a
clé :	234829008467be186c3de14aae72d62c
sel de clé :	32f2870d
gabarit de clé (m) :	32f2870d55555555555555555555555
clé OUX gabarit de clé :	11baae0dd132eb4d3968b41ffb278379
IV :	006e5cba50681de55c621599d462564a
IV' :	595b699bbd3bc0df26062093c1ad8f73
j = 0	
IV' oux j :	595b699bbd3bc0df26062093c1ad8f73
S(-1) :	00000000000000000000000000000000IV' oux S(-1) oux j :
	595b699bbd3bc0df26062093c1ad8f73
S(0) :	71ef82d70a172660240709c7fbb19d8e
texte en clair :	70736575646f72616e646f6d6e657373
texte chiffré :	019ce7a26e7854014a6366aa95d4eefd
j = 1IV' oux j :	595b699bbd3bc0df26062093c1ad8f72
S(0) :	71ef82d70a172660240709c7fbb19d8e
IV' oux S(0) oux j :	28b4eb4cb72ce6bf020129543a1c12fc
S(1) :	3abd640a60919fd43bd289a09649b5fc
texte en clair :	20697320746865206e65787420626573
texte chiffré :	1ad4172a14f9faf455b7f1d4b62bd08f
j = 2	
IV' oux j :	595b699bbd3bc0df26062093c1ad8f71
S(1) :	3abd640a60919fd43bd289a09649b5fc
IV' oux S(1) oux j :	63e60d91ddaa5f0b1dd4a93357e43a8d
S(2) :	220c7a8715266565b09ecc8a2a62b11b
texte en clair :	74207468696e67
texte chiffré :	562c0eef7c4802

B.2 Vecteurs d'essai AES-CM

Longueur de segment de flux de clé :	044 512 octets (65 282 blocs AES)
Clé de session :	B7E151628AED2A6ABF7158809CF4F3C
Compteur de retour à zéro :	00000000

Numéro de séquence : 0000
SSRC : 00000000
Sel de session : F0F1F2F3F4F5F6F7F8F9FAFBFCFD0000 (déjà déplacé)
Décalage : F0F1F2F3F4F5F6F7F8F9FAFBFCFD0000

Compteur	**Flus de clés**
F0F1F2F3F4F5F6F7F8F9FAFBFCFD0000	E03EAD0935C95E80E166B16DD92B4EB4
F0F1F2F3F4F5F6F7F8F9FAFBFCFD0001	D23513162B02D0F72A43A2FE4A5F97AB
F0F1F2F3F4F5F6F7F8F9FAFBFCFD0002	41E95B3BB0A2E8DD477901E4FCA894C0
...	...
F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF	EC8CDF7398607CB0F2D21675EA9EA1E4
F0F1F2F3F4F5F6F7F8F9FAFBFCFDFF00	362B7C3C6773516318A077D7FC5073AE
F0F1F2F3F4F5F6F7F8F9FAFBFCFDFF01	6A2CC3787889374FBEB4C81B17BA6C44

Nota bene : ce cas d'essai est biaisé de telle sorte que la dernière partie du segment de flux de clé coincide avec le cas d'essai de la Section F.5.1 de [CTR].

B.3 Vecteurs d'essai de déduction de clé

Ce paragraphe donne des données d'essai pour la fonction de déduction de clé par défaut, qui utilise AES-128 en mode compteur. Dans ce qui suit, on passe par la déduction initiale de clé pour le chiffrement AES-128 en mode compteur, qui exige une clé de chiffrement de session de 16 octets et un sel de session de 14 octets, et une fonction d'authentification qui exige une clé d'authentification de session de 94 octets. Ces valeurs sont appelées la clé de chiffrement, le sel de chiffrement, et la clé d'authentification dans ce qui suit. Comme c'est la déduction initiale de clé et que le taux de déduction de clé est égal à zéro, la valeur de (indice DIV taux_de_déduction_de_clé) est zéro (en fait, une chaîne de six octets de zéros). Dans ce qui suit, on abrège taux_de_déduction_de_clé en kdr.

Les entrées à la fonction de déduction de clé sont les 16 octets de la clé maîtresse et les 14 octets de sel maître:
clé maîtresse : E1F97A0D3E018BE0D64FA32C06DE4139
sel maître : 0EC675AD498AFEEBB6960B3AABE6

On montre d'abord comment est générée la clé de chiffrement. Le bloc d'entrée pour AES-CM est généré en ouxiant le sel maître avec l'enchaînement de l'étiquette de clé de chiffrement 0x00 avec (indice DIV kdr) puis en bourrant sur la droite avec deux octets nuls (ce qui met en œuvre l'opération de multiplication par 2^16, voir au paragraphe 4.3.3). La valeur résultante est alors chiffrée avec AES-CM en utilisant la clé maîtresse pour obtenir la clé de chiffrement.

indice DIV kdr : 000000000000
étiquette : 00
sel maître : 0EC675AD498AFEEBB6960B3AABE6
oux : 0EC675AD498AFEEBB6960B3AABE6 (x, PRF input)
x*2^16 : 0EC675AD498AFEEBB6960B3AABE60000 (AES-CM input)
clé de chiffrement : C61E7A93744F39EE10734AFE3FF7A087 (AES-CM output)

Ensuite, on montre comment est généré le sel de chiffrement. Le bloc d'entrée pour AES-CM est généré par l'opération OUX sur le sel maître avec l'enchaînement de l'étiquette de sel de chiffrement. Cette valeur est bourrée et chiffrée comme ci-dessus.

indice DIV kdr : 000000000000
étiquette : 02
sel maître : 0EC675AD498AFEEBB6960B3AABE6oux : 0EC675AD498AFEE9B6960B3AABE6 (x, entrée de PRF)
x*2^16 : 0EC675AD498AFEE9B6960B3AABE60000 (entrée AES-CM)
30CBBC08863D8C85D49DB34A9AE17AC6 (résultat AES-CM)
sel de chiffrement : 30CBBC08863D8C85D49DB34A9AE1

On montre maintenant comment est générée la clé d'authentification. Le bloc d'entrée pour AES-CM est généré comme précédemment, mais en utilisant l'étiquette de clé d'authentification.

indice DIV kdr : 000000000000
étiquette : 01
sel maître : 0EC675AD498AFEEBB6960B3AABE6
oux : 0EC675AD498AFEEAB6960B3AABE6 (x, PRF input)

x*2^16 : 0EC675AD498AFEEAB6960B3AABE60000 (AES-CM input)

Ci dessous, la clé d'authentification est montrée à gauche, tandis que les blocs d'entrée AES correspondants sont à droite.

Clé d'authentification	**Blocs d'entrée AES**
CEBE321F6FF7716B6FD4AB49AF256A15	0EC675AD498AFEEAB6960B3AABE60000
6D38BAA48F0A0ACF3C34E2359E6CDBCE	0EC675AD498AFEEAB6960B3AABE60001
E049646C43D9327AD175578EF7227098	0EC675AD498AFEEAB6960B3AABE60002
6371C10C9A369AC2F94A8C5FBCDDDC25	0EC675AD498AFEEAB6960B3AABE60003
6D6E919A48B610EF17C2041E47403576	0EC675AD498AFEEAB6960B3AABE60004
6B68642C59BBFC2F34DB60DBDFB2	0EC675AD498AFEEAB6960B3AABE60005

Adresse des auteurs

Les questions et commentaires devraient être adressés aux auteurs et à avt@ietf.org :

Mark Baugher
Cisco Systems, Inc.
5510 SW Orchid Street
Portland, OR 97219 USA
téléphone : +1 408-853-4418
mél : mbaugher@cisco.com

Elisabetta Carrara
Ericsson Research
SE-16480 Stockholm
Sweden
téléphone : +46 8 50877040
mél : elisabetta.carrara@ericsson.com

David A. McGrew
Cisco Systems, Inc.
San Jose, CA 95134-1706
USA
téléphone : +1 301-349-5815
mél : mcgrew@cisco.com

Mats Naslund
Ericsson Research
SE-16480 Stockholm
Sweden
téléphone : +46 8 58533739
mél : mats.naslund@ericsson.com

Karl Norrman
Ericsson Research
SE-16480 Stockholm
Sweden
téléphone : +46 8 4044502
mél : karl.norrman@ericsson.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 pourraient 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 le 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 droits de reproduction, 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 l'Internet Society.