Groupe de travail Réseau
Request for Comments : 4771
Catéorie : En cours de normalisation
Traduction Claude Brière de L'Isle

V. Lehtovirta, M. Naslund et K. Norrman
Ericsson
janvier 2007

Transformation d'intégrité portant un compteur de débordement pour le protocole de transport sûr en temps réel (SRTP)

Résumé
Le présent document définit une transformation d'intégrité pour le protocole de transport sûr en temps réel (SRTP, *Secure Real-time Transport Protocol)* (RFC 3711) qui permet que le compteur de retour à zéro (ROC, *roll-over counter*) soit transmis dans les paquets SRTP au titre de l'étiquette d'authentification. Le besoin d'envoi du ROC dans les paquets SRTP découle des situations où le receveur se joint à une session SRTP en cours et a besoin de se synchroniser de façon rapide et robuste. Le mécanisme améliore aussi le fonctionnement de SRTP dans les cas où il y a un risque de perdre la synchronisation entre envoyeur et receveur.

Table des matières

1. Introduction

Quand un receveur se joint à une session SRTP [RFC3711] en cours, la signalisation hors bande doit fournir au receveur la valeur de ROC qu'utilise actuellement l'envoyeur. Par exemple, elle peut être transférée dans la charge utile d'en-tête commun d'un message MIKEY [RFC3830]. Dans certains cas, le receveur ne sera pas capable de synchroniser son ROC avec celui utilisé par l'envoyeur, même si il le lui a signalé hors bande. Des exemples où apparaissent des échecs de synchronisation sont :

1. Le receveur reçoit le ROC dans un message MIKEY avec une clé requise pour un service continu particulier. Il ne se joint cependant pas au service avant quelques heures, et à ce moment le numéro de séquence (SEQ) de l'envoyeur a fait un tour complet pour revenir à zéro, et donc l'envoyeur, pendant ce temps a augmenté la valeur du ROC. Quand l'usager se joint au service, il prend le SEQ du premier paquet SRTP venu et ajoute le ROC pour construire l'indice. Si la protection de l'intégrité est utilisée, le paquet va être éliminé. Si il n'y a pas de protection de l'intégrité, le paquet peut (si le taux de déduction de clé n'est pas zéro) être déchiffré en utilisant la mauvaise clé de session, car le ROC est utilisé

comme entrée dans la déduction de clé de session. Dans l'un et l'autre cas, le receveur n'aura pas son ROC synchronisé avec l'envoyeur, et il n'est pas possible de récupérer sans signalisation hors bande.

2. Si le receveur quitte la session (parce qu'il quitte la zone de couverture radio ou à cause d'une action de l'utilisateur) et ne recommence pas à recevoir du trafic provenant du service après que 2^15 paquets ont été envoyés, le receveur va être hors de synchronisation (pour les mêmes raisons que dans l'exemple 1).

3. Le receveur rejoint un service quand le SEQ est récemment revenu à zéro (disons, SEQ = 0x0001). L'envoyeur génère un message MIKEY et inclut la valeur courante de ROC (disons, ROC = 1) dans le message MIKEY. Le message MIKEY atteint le receveur, qui lit la valeur de ROC et initialise son ROC local à 1. Maintenant, si un paquet SRTP antérieur au retour à zéro, c'est-à-dire, avec un SEQ inférieur à 0 (disons, SEQ = 0xffff) a été retardé et atteint le receveur comme premier paquet SRTP qu'il voit, le receveur va initialiser son plus haut numéro de séquence reçu, s_l, à 0xffff. Ensuite, le receveur va recevoir des paquets SRTP avec des numéros de séquence supérieurs à zéro, et va en déduire que le SEQ a fait un tour. Donc, le receveur va incorrectement mettre à jour le ROC et être hors de synchronisation.

4. Similaire à (3), comme le SEQ initial est choisi au hasard par l'envoyeur, il se peut qu'il soit choisi à une valeur très proche de 0xffff. Dans ce cas, si les premiers paquets sont perdus, le receveur peut de la même façon se trouver hors de synchronisation.

Ces problèmes ont été reconnus dans, par exemple, le 3GPP2 et le 3GPP, où SRTP est utilisé pour la protection de supports en direct dans leurs solutions respectives de diffusion groupée/diffusion [BCMCS], [MBMS]. Le problème 4 existe réellement par nature vue la façon dont l'initialisation de SEQ est faite dans RTP.

Une approche possible pour régler ce problème pourrait être de porter le ROC dans le champ d'identifiant de clé maîtresse (MKI, *Master Key Identifier*) de chaque paquet SRTP. Cela a l'avantage que le receveur sait immédiatement l'indice entier pour un paquet. Malheureusement, le MKI n'a pas de signification dans la RFC 3711 (autre que de spécifier la clé maîtresse) et une mise en œuvre régulièrement conforme à la RFC 3711 ne serait pas capable d'utiliser les informations portées dans le MKI. De plus, le champ MKI n'est pas protégé en intégrité ; donc, il faut faire attention à éviter les attaques évidentes contre la synchronisation.

Dans le présent document, on présente une solution où le ROC est porté dans l'étiquette d'authentification d'une transformation d'intégrité spéciale dans les paquets SRTP choisis.

L'avantage de cette approche est que la fonctionnalité de synchronisation rapide et robuste peut être réalisée comme une transformation d'intégrité séparée, en utilisant les facilités existantes dans SRTP. De plus, quand le ROC est transmis au receveur, il a besoin d'être protégé en intégrité pour éviter des attaques persistantes de déni de service (DoS) ou des erreurs de transmission qui pourraient mettre le receveur hors synchronisation. (Une attaque de DoS est considérée comme persistante si elle peut durer après que l'attaquant a quitté la zone ; dans ce cas particulier, un attaquant pourrait modifier le ROC dans un paquet et la victime serait désynchronisée jusqu'à ce que le nouveau ROC soit transmis). La discussion ci-dessus conduit à la conclusion qu'il y a du sens à porter le ROC dans l'étiquette d'authentification d'une transformation d'intégrité.

1.1 Terminologie

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 le BCP 14, RFC 2119.

2. Transformation

La transformation, ci-après appelée transformation portant un compteur de retour à zéro (RCC, *Roll-over Counter Carrying Transform*) fonctionne comme suit.

L'envoyeur traite le paquet RTP conformément à la RFC 3711. Quand il applique la transformation d'intégrité de message, l'envoyeur vérifie si le SEQ est égal à 0 modulo un entier constant non zéro R. Si c'est le cas, l'envoyeur calcule le MAC de la même façon que quand on utilise la transformation d'intégrité par défaut (c'est-à-dire, HMAC-SHA1(auth_key, Authenticated_portion || ROC)). Ensuite, l'envoyeur tronque le MAC de 32 bits pour générer MAC_tr, c'est-à-dire, MAC_tr est l'étiquette de longueur moins les 32 bits de poids fort du MAC. Ensuite l'envoyeur construit l'étiquette comme TAG =

ROC_sender || MAC_tr, où ROC_sender est la valeur de son ROC local, et ajoute l'étiquette au paquet. Voir dans la section des considérations sur la sécurité ci-dessous la discussion des effets du raccourcissement du MAC. En particulier, noter qu'une longueur d'étiquette de 32 bits ne donne aucune sécurité.

Si le SEQ n'est pas égal à 0 mod R, l'envoyeur procède juste au traitement du paquet en accord avec la RFC 3711 sans effectuer les actions du paragraphe précédent.

La valeur R est le taux auquel le ROC est inclus dans les paquets SRTP. Comme le ROC consomme quatre octets, cela donne la possibilité de l'utiliser avec économie.

Quand le receveur reçoit un paquet SRTP, il le traite en accord avec la RFC 3711, excepté que durant le traitement de l'authentification, le ROC_local est remplacé par le ROC_sender (pris dans le paquet). Cela fonctionne comme suit. Dans l'étape où la protection de l'intégrité est à vérifier, si le SEQ est égal à 0 modulo R, le receveur extrait le ROC_sender del'étiquette TAG et vérifie que le MAC calculé (de la même façon que si la transformation d'intégrité par défaut était utilisée) sur la portion authentifiée du paquet (comme défini dans la [RFC3711]) mais enchaînée avec le ROC_sender au lieu de l'enchaîner avec le local_ROC. Le receveur génère MAC_tr pour la vérification de MAC de la même façon que l'a fait l'envoyeur. Noter que la clé de session utilisée dans le calcul de MAC dépend du ROC, et durant la déduction de la clé d'intégrité de session, le ROC qui se trouve dans le paquet considéré DOIT être utilisé. Si la vérification réussit, le receveur règle son ROC local égal au ROC porté dans le paquet. Si le MAC ne se vérifie pas, le paquet DOIT être éliminé. La raison de l'utilisation du ROC provenant du paquet dans le calcul du MAC est que si le receveur a une valeur de ROC incorrecte, la vérification du MAC va échouer, de sorte que le receveur ne va pas corriger son ROC.

Si le SEQ n'est pas égal à 0 mod R, le receveur procède juste au traitement du paquet conformément à la RFC 3711 sans effectuer les actions de l'alinéa précédent.

Comme le protocole de contrôle de transport sûr en temps réel (SRTCP, *Secure Real-time Transport Control Protocol*) porte déjà l'indice entier dans la bande, il n'y a pas de raison d'appliquer cette transformation à SRTCP. Donc, la transformation DEVRA seulement être appliquée à SRTP, et NE DEVRA PAS être utilisée avec SRTCP.

3. Modes de transformation

La transformation ci-dessus ne fournit de protection de l'intégrité que pour les paquets tqui portent le ROC (c'est ce qu'on va appeler le mode 1). Dans les cas où il est besoin de protéger l'intégrité de tous les paquets, les paquets qui n'ont pas SEQ égal à 0 mod R DOIVENT être protégés en utilisant la transformation d'intégrité par défaut (qu'on va appeler le mode 2).

Dans certaines circonstances, il peut être acceptable de n'utiliser la protection de l'intégrité sur aucun des paquets ; c'est ce qu'on appelera le mode 3. Sans protection de l'intégrité des paquets portant le ROC, une attaque de DoS, qui va durer jusqu'à la réception du prochain ROC correct, est possible. Il faut s'assurer de lire avec attention les considérations sur la sécurité de la Section 5 avant d'utiliser le mode 3.

Dans le cas où aucune protection de l'intégrité n'est offerte, c'est-à-dire le mode 3, ce qui suit s'applique. La couche SRTP du receveur DEVRAIT ignorer la valeur de ROC provenant du paquet si la couche d'application peut lui indiquer que le ROC local est synchronisé avec celui de l'envoyeur (donc, le paquet va être traité en utilisant le ROC local). Noter que le ROC reçu DOIT quand même être retiré du paquet avant de continuer le traitement. Dans ce scénario, la rétroaction de la couche d'application à la couche SRTP n'a pas besoin d'être paquet par paquet, et peut consister simplement en une valeur booléenne réglée par la couche d'application et lue par la couche SRTP.

On note donc la différence suivante. Utiliser le mode 2 protège l'intégrité de tous les paquets RTP, mais seulement ajoute le ROC à ceux qui ont le SEQ divisible par R. Utiliser le mode 1 et régler R égal à un va aussi protéger l'intégrité de tous les paquets, mais va en plus de cela ajouter le ROC à chaque paquet. Les modes 1 et 2 DOIVENT calculer le MAC de la même façon que la transformation d'authentification pré-définie pour SRTP, c'est-à-dire, HMAC-SHA1.

Pour se conformer à la présente spécification, la mise en œuvre des modes 1, 2, et 3 est OBLIGATOIRE. Cependant, il appartient à la politique locale de décider quels modes il est permis d'utiliser.

4. Négociation de paramètre

La RCC exige que quelques paramètres soient signalés hors bande. Les paramètres qui doivent être en place avant que la transformation puisse être utilisée sont le mode de transformation d'intégrité et le taux, R, auquel le ROC va être transmis. Cela peut être fait en utilisant, par exemple, MIKEY [RFC3830].

Pour effectuer la négociation de paramètre en utilisant MIKEY, trois transformations d'intégrité ont été enregistrées -- RCCm1, RCCm2, et RCCm3 dans le Tableau 6.10.1.c de la [RFC3830] -- pour les trois modes définis.

Tableau 1. Transformations d'intégrité

Algorithme d'authentification SRTP	Valeur
RCCm1	2
RCCm2	3
RCCm3	4

De plus, le paramètre R a été enregistré dans le Tableau 6.10.1.a de la [RFC3830].

Tableau 2. Paramètre de transformation d'intégrité

Type	Signification	Valeurs possibles
13	taux de transmission de ROC	entier de 16 bits

Le taux de transmission de ROC, R, est donné dans l'ordre des octets du réseau. R DOIT être un entier non signé non zéro. Si le taux de transmission de ROC n'est pas inclus dans la négociation, par défaut la valeur de 1 DEVRA être utilisée.

Pour avoir la capacité d'utiliser des transformations d'intégrité différentes pour SRTP et SRTCP, ce qui est nécessaire en connexion avec l'utilisation de RCC, les paramètres supplémentaires suivants ont été enregistrés dans le Tableau 6.10.1.a de la [RFC3830]:

Tableau 3. Paramètres d'intégrité

Type	Signification	Valeurs possibles
14	algorithme d'authentification SRTP	voir ci-dessous
15	algorithme d'authentification SRTCP	voir ci-dessous
16	longueur de clé d'authentification de session SRTP	voir ci-dessous
17	longueur de clé d'authentification de session SRTCP	voir ci-dessous
18	longueur d'étiquette d'authentification SRTP	voir ci-dessous
19	longueur d'étiquette d'authentification SRTCP	voir ci-dessous

Les valeurs possibles pour les algorithmes d'authentification (types 14 et 15) sont les mêmes que pour le paramètre "Algorithme d'authentification" (type 2) dans leTableau 6.10.1.a de la RFC 3830 avec l'ajout des valeurs trouvées dans le Tableau 1 ci-dessus.

Les valeurs possibles pour les longueurs de clé d'authentification de session (types 16 et 17) sont les mêmes que pour le paramètre "Longueur de clé d'authentification de session" (type 3) dans le Tableau 6.10.1.a de la RFC 3830.

Les valeurs possibles pour les longueurs d'étiquette d'authentification (types 18 et 19) sont les mêmes que pour le paramètre "Longueur d'étiquette d'authentification " (type 11) du Tableau 6.10.1.a de la RFC 3830 avec l'ajout que la longueur du ROC DOIT être incluse dans le paramètre "Longueur d'étiquette d'authentification". Cela signifie que la longueur minimum d'étiquette quand on utilise RCC est 32 bits.

Pour éviter des ambiguités quand on introduit ces nouveaux paramètres qui ont des fonctionnalités qui chevauchent celles des paramètres existant dans le Tableau 6.10.1.a de la RFC 3830, l'approche suivante DOIT être suivie : si un des types de paramètres 14 à 19 (spécifiant un comportement spécifique de SRTP ou SRTCP) et un paramètre général correspondant (type 2, 3, ou 11) sont tous deux présents dans la politique, le paramètre le plus spécifique DEVRA avoir la préséance. Par exemple, si le paramètre "Algorithme d'authentification" (type 2) est réglé à HMAC-SHA-1, et si "Algorithme d'authentification SRTP" (type 14) est réglé à RCCm1, SRTP va utiliser l'algorithme RCCm1, mais comme il n'y a pas d'algorithme spécifique choisi pour SRTCP, le plus général spécifié (HMAC-SHA-1) est utilisé.

5. Considérations sur la sécurité

Une méthode analogue existe déjà dans SRTCP (l'indice SRTCP est porté dans chaque paquet sous protection de l'intégrité). À notre connaissance, la seule considération de sécurité introduite ici est que l'indice SRTP entier (ROC || SEQ) va devenir public car il est transféré sans chiffrement. (Dans le fonctionnement normal de SRTP, seule la partie SEQ-part de l'indice est divulguée.) Cependant, la RFC 3711 n'identifie pas de besoin de chiffrement de l'indice SRTP.

Il est important de réaliser que seulement chaque Rième paquet est protégé en intégrité dans le mode 1, donc sauf si R = 1, le mécanisme devrait être vu pour ce qu'il est : un moyen d'améliorer la synchronisation entre envoyeur et receveur, et non un remplacement de la protection de l'intégrité.

L'utilisation du mode 3 (NULL-MAC) introduit une vulnérabilité qui n'est pas présente dans la RFC 3711 ; à savoir que si un attaquant modifie le ROC, la modification va passer indétectée chez le receveur, et le receveur va perdre la synchronisation cryptographique jusqu'à ce que le prochain ROC correct soit reçu. Cela implique qu'un attaquant peut effectuer une attaque de DoS en modifiant seulement chaque Rième paquet. À cause de cela, le mode 3 DOIT seulement être utilisé après une évaluation de risque appropriée du réseau sous-jacent. À côté des considerations des paragraphes 9.5 et 9.5.1 de la RFC 3711, les exigences supplémentaires du réseau de transport sous-jacent doivent être satisfaites :

- Le réseau de transport doit seulement consister en domaines de confiance. Cela signifie que chacun sur le chemin de la source à la destination est estimé ne pas modifier les paquets ou en injecter.

- Le réseau de transport doit être protégé contre l'injection de paquets, c'est-à-dire, on doit être sûr que seuls les paquets présents sur le chemin de la source à la ou les destinations proviennent de sources de confiance.

- Si les paquets, sur leur chemin de la source à la ou les destinations, voyagent en dehors d'un domaine de confiance, leur intégrité doit être assurée (par exemple, en utilisant un réseau privé viruel (VPN, *Virtual Private Network*) une connexion ou une liaison louée de confiance).

Dans le cas (supposé courant) où la dernière liaison pour la ou les destinations est une liaison sans fil, la possibilité qu'un attaquant y injecte des paquets fasifiés doit être considérée avec attention avant d'utiliser le mode 3. En particulier, si il est utilisé dans un dispositif de diffusion, de nombreuses destinations seraient affectées par l'attaque. Cependant, sauf si R est grand, cette attaque de DoS serait similaire dans ses effets à un brouillage radio, qui serait plus facile à réaliser.

On doit aussi noter que si le ROC est modifié par un attaquant et qu'aucune protection de l'intégrité n'est utilisée, le résultat du déchiffrement ne sera pas utile aux couches supérieures, et elles doivent être capables de s'accomoder de données qui vont apparaître au hasard. Dans le cas où la protection de l'intégrité est utilisée sur les paquets contenant le ROC, et si le ROC est modifié par un attaquant (et si le receveur a déjà une approximation du ROC, par exemple, en l'ayant eu précédemment) le paquet va être éliminé et le receveur ne va pas être capable de le déchiffrer correctement. Noter, cependant, que la situation est meilleure dans ce dernier cas, car le receveur peut alors essayer différentes valeurs de ROC dans le voisinage de la valeur approximative qu'il a déjà.

Comme la RCC est supposée être utilisée dans un dispositif de diffusion où l'appartenance au groupe va se fonder sur l'accès à une clé symétrique de groupe, il est important de souligner ce qui suit. Avec une protection de l'intégrité fondée sur une clé symétrique, il peut être aussi facile, sinon plus, d'obtenir l'accès à la clé d'intégrité (souvent une combinaison d'une activité peu coûteuse de prendre un abonnement et de casser la sécurité d'un terminal pour en extraire la clé d'intégrité) que d'être capable d'émettre.

Un mot d'avertisement concernant le choix de la longueur de l'étiquette d'authentification : noter que à la différence des étiquettes de MAC courantes, il y a une claire distinction entre l'étiquette d'authentification du RCC et le MAC RCC. L'étiquette est le conteneur qui détient le MAC (et pour certains paquets aussi le ROC) et le MAC est le résultat de l'algorithme de MAC (c'est-à-dire, HMAC-SHA1). La longueur de l'étiquette d'authentification avec la transformation RCC inclut les quatre octets de ROC dans certains paquets. Cela signifie que pour une longueur d'étiquette de n octets, il y a seulement de la place pour un MAC de longueur n - 4, c'est-à-dire, une longueur d'étiquette de n octets ne fournit pas une protection de l'intégrité complète de n octets sur tous les paquets. Il y a cinq cas :

1. RCCm1 est utilisé et la longueur d'étiquette est n. Pour ces paquets où SEQ = 0 mod R, le ROC est porté dans l'étiquette et occupe quatre octets. Cela laisse n - 4 octets pour le MAC.

2. RCCm1 est utilisé et la longueur d'étiquette est n. Pour ces paquets où SEQ != 0 mod R, il n'y a pas de ROC porté dans l'étiquette. Pour RCCm1 il n'y a pas de MAC sur les paquets qui ne portent pas le ROC, donc ni la longueur du MAC ni la longueur de l'étiquette ne sont pertinentes.

3. RCCm2 est utilisé et la longueur d'étiquette est n. Pour ces paquets où SEQ = 0 mod R, le ROC est porté dans l'étiquette et occupe quatre octets. Cela laisse n - 4 octets pour le MAC (c'est équivalent au cas 1).

4. RCCm2 est utilisé et la longueur d'étiquette est n. Pour ces paquets où SEQ != 0 mod R, il n'y a pas de ROC porté dans l'étiquette. Cela laisse n octets pour le MAC.

5. RCCm3 est utilisé. RCCm3 n'utilise aucun MAC, mais le ROC occupe quand même quatre octets dans l'étiquette pour les paquets avec SEQ = 0 mod R, donc la longueur d'étiquette DOIT être réglée à quatre. Pour les paquets avec SEQ != 0 mod R, ni la longueur du MAC ni la longueur de l'étiquette ne sont pertinentes.

La conclusion est que dans les cas 1 et 3, la longueur du MAC est inférieure à la longueur de l'étiquette d'authentification. Pour obtenir la même (ou inférieure) probabilité de réussite de falsification de MAC sur tous les paquets quand on utilise RCCm1 ou RCCm2, qu'avec la transformation d'intégrité par défaut dans la RFC 3711, la longueur d'étiquette doit être réglée à 14 octets, ce qui signifie que la longueur du MAC_tr est 10 octets.

Il est recommandé de régler la longueur d'étiquette à 14 octets quand RCCm1 ou RCCm2 est utilisé, et la longueur d'étiquette DOIT être réglée à quatre octets quand RCCm3 est utilisé.

6. Considérations relatives à l'IANA

En accord avec la Section 10 de la RFC 3830, le consensus de l'IETF est requis pour enregistrer des valeurs dans la gamme 0 à 240 dans l'espace de noms d'algorithme d'authentification SRTP et l'espace de noms de type SRTP.

Les valeurs 2 pour RCCm1, 3 pour RCCm2, et 4 pour RCCm3 ont été enregistrées dans l'espace de noms d'algorithme d'authentification SRTP comme spécifié dans le Tableau 1 de la Section 4.

La valeur 13 pour le taux de transmission de ROC a été enregistrée dans l'espace de noms de type SRTP comme spécifié au Tableau 2 de la Section 4.

Les valeurs 14 à 19 ont été enregistrées dans l'espace de noms de type SRTP conformément au Tableau 3 de la Section 4.

7. Remerciements

Merci à Nigel Dallard, Lakshminath Dondeti, et David McGrew de leurs commentaires et discussions fructueuses.

8. Références

8.1 Références normatives

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

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

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

8.2 Références pour information

[MBMS] 3GPP TS 33.246, "3G Security; Security of Multimedia Broadcast/ Multicast Service (MBMS)", octobre 2006.

[BCMCS] 3GPP2 X.S0022-0, "Broadcast and Multicast Service in cdma2000 Wireless IP Network", février 2005.

Adresse des auteurs

Vesa Lehtovirta
Ericsson Research
02420 Jorvas
Finland
téléphone : 9 2993314
mél : vesa.lehtovirta@ericsson.com

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

Karl Norrman
Ericsson Research
SE-16480 Stockholm
Sweden
téléphone : 8 4044502
mél : karl.norrman@ericsson.com

Déclaration complète de droits de reproduction

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