Groupe de travail Réseau
Request for Comments : 3329
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

J. Arkko, V. Torvinen, &G. Camarillo, Ericsson
A. Niemi & T. Haukka, Nokia
janvier 2003

Accord de mécanisme de sécurité pour le protocole d'initialisation de session (SIP)

Statut de ce mémoire
Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

La présente traduction incorpore les errata n° 2169, 2170 et 3799.

Résumé
Le présent document définit une nouvelle fonctionnalité pour la négociation des mécanismes de sécurité utilisés entre un agent d'utilisateur du protocole d'initialisation de session (SIP, *Session Initiation Protocol*) et son entité SIP de prochain bond. Cette nouvelle fonctionnalité s'ajoute aux méthodes existantes de choix des mécanismes de sécurité entre les entités SIP.

Table des Matières

1. Introduction

Traditionnellement, les protocoles de sécurité incluaient des facilités pour s'accorder sur les mécanismes, algorithmes, et autres paramètres de sécurité utilisés. C'est destiné à augmenter la souplesse, car différents mécanismes conviennent habituellement pour les différents scénarios. Aussi, l'évolution des mécanismes de sécurité a souvent introduit de nouveaux algorithmes, ou des problèmes cachés dans ceux qui existent, rendant nécessaire la négociation des mécanismes.

L'objet de la présente spécification est de définir une fonction de négociation pour le protocole d'initialisation de session (SIP) [RFC3261]. Cette négociation est destinée à ne fonctionner qu'entre un UA et son entité SIP de premier bond.

1.1 Motifs

Sans une méthode de choix sécurisée entre les mécanismes de sécurité et/ou leurs paramètres, SIP est vulnérable à certaines attaques. L'authentification et la protection de l'intégrité en utilisant plusieurs méthodes et algorithmes est vulnérable aux attaques par interposition (MitM, *Man-in-the-Middle*) (par exemple, voir la [RFC2617]).

Il est aussi difficile ou parfois même impossible de savoir si un mécanisme de sécurité spécifique est vraiment indisponible à une entité SIP homologue, ou si en fait une attaque par interposition est en cours.

Dans certains petits réseaux, ces questions ne sont pas très pertinentes, car les administrateurs de tels réseaux peuvent déployer les versions de logiciel appropriées et mettre en place des politiques pour utiliser exactement le bon type de sécurité. Cependant, SIP est aussi supposé être déployé dans des centaines de millions de petits appareils avec peu ou pas de possibilités de coordonner des politiques de sécurité, de mettre à niveau les logiciels, ce qui nécessite que des fonctions de négociation soient disponibles depuis le tout début de l'installation (par exemple, voir la [RFC4083]).

1.2 Objectifs de conception

1. Les entités impliquées dans le processus d'accord de sécurité ont besoin de découvrir exactement quels mécanismes de sécurité appliquer, de préférence sans allers-retours supplémentaires excessifs.

2. Le choix des mécanismes de sécurité eux-mêmes doit être sécurisé. Traditionnellement, tous les protocoles de sécurité utilisent une forme de négociation sûre. Par exemple, après l'établissement de clés mutuelles par Diffie-Hellman, IKE envoie des hachages des données envoyées précédemment incluant les mécanismes de chiffrement offerts [RFC2409]. Cela permet aux homologues de détecter si l'offre initiale, non protégée, a été altérée.

3. Les entités impliquées dans le processus d'accord de sécurité ont besoin d'être capables d'indiquer la réussite ou l'échec du processus d'accord de sécurité.

4. Le processus d'accord de sécurité NE DEVRAIT PAS introduire d'état supplémentaire à entretenir par les entités impliquées.

1.3 Conventions

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMETE", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, [RFC2119].

2. Solution

2.1 Généralités sur le fonctionnement

Le flux de messages ci-dessous illustre comment fonctionne le mécanisme défini dans le présent document :

1. Client ----------liste de clients----------> Serveur
2. Client <---------liste de serveurs---------- Serveur
3. Client --------(activer la sécurité)--------- Serveur
4. Client ----------liste de serveurs---------> Serveur
5. Client <----------ok ou erreur---------- Serveur

Figure 1 : Flux de messages d'accord de sécurité

Étape 1 : Les clients qui souhaitent utiliser la présente spécification peuvent envoyer une liste des mécanismes de sécurité

qu'ils prennent en charge avec la première demande au serveur.

Étape 2 : Les serveurs qui souhaitent utiliser la présente spécification mettent au défi le client d'effectuer la procédure d'accord de sécurité. Les mécanismes et les paramètres de sécurité pris en charge par le serveur sont envoyés avec ce défi.

Étape 3 : Le client procède alors au choix du mécanisme de sécurité qui a la plus forte préférence qu'ils ont en commun et à l'activation de la sécurité choisie.

Étape 4 : Le client contacte à nouveau le serveur, en utilisant maintenant le mécanisme de sécurité choisi. La liste des mécanismes de sécurité pris en charge du serveur est retournée comme réponse au défi.

Étape 5 : Le serveur vérifie sa propre liste de mécanismes de sécurité afin de s'assurer que la liste d'origine n'a pas été modifiée.

Cette procédure est sans état pour les serveurs (sauf si les mécanismes de sécurité utilisés exigent que le serveur conserve un état).

Les listes du client et du serveur sont toutes deux statiques (c'est-à-dire qu'ils ne peuvent pas la changer en fonction des entrées de l'autre). Les nœuds PEUVENT cependant tenir plusieurs listes statiques, une pour chaque interface, par exemple.

Entre les étapes 1 et 2, le serveur PEUT établir si nécessaire un mécanisme de sécurité auto descriptif. Noter qu'avec ce type de mécanismes de sécurité, le serveur est nécessairement à états pleins. Le client établirait le mécanisme de sécurité non auto descriptif entre les étapes 2 et 4.

2.2 Syntaxe

On définit trois nouveaux champs d'en-tête SIP, à savoir Security-Client, Security-Server et Security-Verify. La notation utilisée dans les définitions du BNF augmenté pour les éléments de syntaxe dans ce paragraphe est celle utilisée dans SIP [RFC3261], et tous les éléments non définis dans ce paragraphe sont comme défini dans SIP et les documents auxquels il se réfère :

```
security-client  = "Security-Client" HCOLON sec-mechanism *(COMMA sec-mechanism)
security-server  = "Security-Server" HCOLON sec-mechanism *(COMMA sec-mechanism)
security-verify  = "Security-Verify" HCOLON sec-mechanism *(COMMA sec-mechanism)
sec-mechanism    = mechanism-name *(SEMI mech-parameters)
mechanism-name   = ( "digest" / "tls" / "ipsec-ike" / "ipsec-man" / jeton )
mech-parameters  = ( preference / digest-algorithm / digest-qop / digest-verify / extension )
preference       = "q" EQUAL qvalue
qvalue           = ( "0" [ "." 0*3DIGIT ] ) / ( "1" [ "." 0*3("0") ] )
digest-algorithm = "d-alg" EQUAL jeton
digest-qop       = "d-qop" EQUAL jeton
digest-verify    = "d-ver" EQUAL LDQUOT 32LHEX RDQUOT
extension        = generic-param
```

Noter que qvalue est déjà défini dans le BNF de SIP [RFC3261]. On a copié ici ses définitions pour être complet.

Les paramètres décrits par le BNF ci-dessus ont la sémantique suivante :

Mechanism-name
Ce jeton identifie le mécanisme de sécurité pris en charge par le client, lorsque il apparaît dans un champ d'en-tête Security-Client; ou par le serveur, lorsque il apparaît dans un champ d'en-tête Security-Server ou Security-Verify. Les jetons mechanism-name sont enregistrés auprès de l'IANA. La présente spécification définit quatre valeurs :

* "tls" pour TLS [RFC2246].
* "digest" pour le résumé HTTP [RFC2617].
* "ipsec-ike" pour IPsec avec IKE [RFC2401].
* "ipsec-man" pour IPsec chiffré manuellement sans IKE.

Preference
La valeur "q" indique une préférence relative pour le mécanisme en cause. Plus la valeur est élevée, plus le mécanisme est préféré. Tous les mécanismes de sécurité DOIVENT avoir des valeurs "q" différentes. C'est une erreur de fournir deux mécanismes avec la même valeur "q".

Digest-algorithm
Ce paramètre facultatif n'est défini ici que pour le résumé HTTP [RFC2617] afin d'empêcher l'attaque par enchères sur le paramètre d'algorithme de résumé HTTP. Le contenu du champ PEUT avoir les mêmes valeurs que défini dans la [RFC2617] pour le champ "algorithm".

Digest-qop
Ce paramètre facultatif n'est défini ici que pour le résumé HTTP [RFC2617] afin d'empêcher l'attaque par enchères sur le paramètre d'algorithme de résumé HTTP. Le contenu du champ PEUT avoir les mêmes valeurs que défini dans la [RFC2617] pour le champ "qop".

Digest-verify
Ce paramètre facultatif n'est défini ici que pour le résumé HTTP [RFC2617] afin d'empêcher l'attaque par enchères sur l'accord du mécanisme de sécurité de SIP (le présent document). Le contenu du champ est compté exactement de la même façon que "request-digest" dans la [RFC2617] excepté que le champ d'en-tête Security-Server est inclus dans le paramètre A2. Si la valeur de la directive "qop" est "auth" ou n'est pas spécifiée, alors A2 est :

A2 = Method ":" digest-uri-value ":" security-server

Si la valeur de "qop" est "auth-int", alors A2 est :

A2 = Method ":" digest-uri-value ":" H(entity-body) ":" security-server

Toutes les espaces blanches linéaires dans le champ d'en-tête Security-Server DOIVENT être remplacées par un seul caractère SP avant de calculer ou interpréter le paramètre digest-verify. Les paramètres Method, digest-uri-value, entity-body, et tous les autres paramètres de résumé HTTP sont comme spécifié dans la [RFC2617].

Noter que la présente spécification n'introduit aucune extension ni changement au résumé HTTP [RFC2617]. La présente spécification réutilise seulement les mécanismes existants du résumé HTTP pour protéger la négociation du mécanisme de sécurité entre les entités SIP.

2.3 Fonctionnement du protocole

Ce paragraphe traite des détails du protocole impliqué dans la négociation entre un UA SIP et son entité SIP de prochain bond. Tout au long du texte, on fait référence à l'entité SIP de prochain bond comme au mandataire de premier bond ou mandataire extérieur. Cependant, le lecteur DEVRAIT se souvenir qu'un serveur d'agent d'utilisateur peut aussi être le prochain bond d'un client d'agent d'utilisateur.

2.3.1 Initié par le client

Si un client en arrive à utiliser TLS pour contacter le serveur parce que il a suivi les règles spécifiées dans la [RFC3263], le client NE DOIT PAS utiliser la procédure d'accord de sécurité de la présente spécification. Si un client en arrive à utiliser des connexions non TLS à cause des règles de la [RFC3263], le client PEUT utiliser l'accord de sécurité de la présente spécification pour détecter une falsification du DNS, ou pour négocier une autre sécurité que TLS.

Un client qui souhaite utiliser l'accord de sécurité de la présente spécification DOIT ajouter un champ d'en-tête Security-Client à une demande adressée à son mandataire de premier bond (c'est-à-dire, la destination de la demande est le mandataire de premier bond). Ce champ d'en-tête contient une liste de tous les mécanismes de sécurité que le client prend en charge. Le client NE DEVRAIT PAS ajouter de paramètres de préférence à cette liste. Le client DOIT ajouter les deux champs d'en-tête Require et Proxy-Require avec la valeur "sec-agree" à cette demande.

Le contenu du champ d'en-tête Security-Client PEUT être utilisé par le serveur pour inclure des informations nécessaires dans sa réponse.

Un serveur qui reçoit une demande non protégée qui contient un champ d'en-tête Require ou Proxy-Require avec la valeur "sec-agree" DOIT répondre au client par une répons 494 (Accord de sécurité exigé). Le serveur DOIT ajouter un champ d'en-tête Security-Server à cette réponse donnant la liste des mécanismes de sécurité qu'il prend en charge. Le serveur DOIT ajouter sa liste à la réponse même si il n'y a pas de mécanisme de sécurité commun dans la liste du client et dans

celle du serveur. La liste du serveur NE DOIT PAS dépendre du contenu de la liste du client.
Le serveur DOIT comparer la liste reçue dans le champ d'en-tête Security-Client avec la liste à envoyer dans le champ d'en-tête Security-Server. Lorsque le client reçoit cette réponse, il va choisir le mécanisme de sécurité commun qui a la plus forte valeur "q". Donc, le serveur DOIT ajouter les informations nécessaires pour que le client puisse initier ce mécanisme (par exemple, un champ d'en-tête Proxy-Authenticate pour le résumé HTTP).

Lorsque le client reçoit une réponse avec un champ d'en-tête Security-Server, il DOIT choisir le mécanisme de sécurité qui dans la liste du serveur a la plus forte valeur "q" parmi tous les mécanismes qui sont connus du client. Ensuite, il DOIT initier ce mécanisme de sécurité particulier comme décrit au paragraphe 3.5. Cette initiation PEUT être menée à bien sans impliquer aucun échange de message SIP (par exemple, en établissant une connexion TLS).

Si un attaquant a modifié le champ d'en-tête Security-Client dans la demande, le serveur ne peut pas inclure dans sa réponse les informations nécessaires pour établir le mécanisme de sécurité commun avec la plus forte valeur de préférence (par exemple, il manque le champ d'en-tête Proxy-Authenticate). Un client qui détecte un tel manque d'informations dans la réponse DOIT considérer le processus actuel d'accord de sécurité comme interrompu, et PEUT essayer de le recommencer en envoyant une nouvelle demande avec un champ d'en-tête Security-Client comme décrit ci-dessus.

Toutes les demandes SIP envoyées ensuite par le client à ce serveur DEVRAIENT faire usage du mécanisme de sécurité initié à l'étape précédente. Ces demandes DOIVENT contenir un champ d'en-tête Security-Verify qui reflète la liste du serveur reçue précédemment dans le champ d'en-tête Security-Server. Ces demandes DOIVENT aussi avoir les deux champs d'en-tête Require et Proxy-Require avec la valeur "sec-agree".

Le serveur DOIT vérifier que les mécanismes de sécurité énumérés dans le champ d'en-tête Security-Verify des demandes entrantes correspondent à sa liste statique des mécanismes de sécurité pris en charge.

Noter que, suivant les règles de comparaison de champ d'en-tête SIP standard définies dans la [RFC3261], les deux listes doivent contenir les mêmes mécanismes de sécurité dans le même ordre pour être considérées comme équivalentes. De plus, pour chaque mécanisme de sécurité particulier, les paramètres doivent avoir les mêmes valeurs dans les deux listes.

Le serveur peut poursuivre le traitement d'une certaine demande si, et seulement si la liste n'a pas été modifiée. Si on détecte des modifications sur la liste, le serveur DOIT répondre au client avec un code 494 (Accord de sécurité exigé). Cette réponse DOIT inclure la liste non modifiée des mécanismes de sécurité pris en charge par le serveur. Si la liste n'a pas été modifiée, et si le serveur est un mandataire, il DOIT retirer la valeur "sec-agree" des deux champs d'en-tête Require et Proxy-Require, et retirer ensuite les champs d'en-tête si il ne reste aucune valeur.

Une fois que la sécurité a été négociée entre deux entités SIP, ces mêmes entités SIP PEUVENT utiliser la même sécurité lorsque elles communiquent ensemble dans des rôles SIP différents. Par exemple, si un UAC et son mandataire de sortie négocient une certaine sécurité, ils PEUVENT essayer d'utiliser la même sécurité pour les demandes entrantes (c'est-à-dire que l'UA va agir comme un UAS).

L'utilisateur d'un UA DEVRAIT être informé des résultats de l'accord sur le mécanisme de sécurité. L'usager PEUT décliner l'offre d'un mécanisme de sécurité particulier, et interrompre la communication SIP avec l'homologue.

2.3.2 Initié par le serveur

Un serveur décide d'utiliser l'accord de sécurité décrit dans le présent document sur la base d'une politique locale. Si un serveur reçoit une demande provenant de l'interface réseau qui est configurée pour utiliser ce mécanisme, il DOIT vérifier que la demande a seulement une entrée Via. Si il y a plusieurs entrées Via, le serveur n'est pas l'entité SIP de premier bond, et il NE DOIT PAS utiliser ce mécanisme. Pour une telle demande, le serveur DOIT retourner une réponse 502 (Mauvaise passerelle).

Un serveur qui décide d'utiliser ce mécanisme d'accord DOIT mettre au défi les demandes non protégées avec une entrée Via sans considération de la présence ou de l'absence de champs d'en-tête Require, Proxy-Require ou Supported dans les demandes entrantes.

Un serveur qui exige, selon sa politique, d'utiliser la présente spécification et qui reçoit une demande qui n'a pas l'étiquette d'option sec-agree dans un champ d'en-tête Require, Proxy-Require ou Supported DOIT retourner une réponse 421 (Extension exigée). Si la demande avait l'étiquette d'option sec-agree dans un champ d'en-tête Supported, il DOIT retourner une réponse 494 (Accord de sécurité exigé). Dans les deux situations, le serveur DOIT aussi inclure dans la réponse un champ d'en-tête Security-Server faisant la liste de ses capacités et un champ d'en-tête Require contenant une étiquette d'option "sec-agree". Le serveur DOIT aussi ajouter les informations nécessaires afin que le client puisse initier le mécanisme de sécurité préféré (par exemple, un champ d'en-tête Proxy-Authenticate pour un résumé HTTP).

Les clients qui prennent en charge l'extension définie dans le présent document DEVRAIENT ajouter un champ d'en-tête Supported avec une valeur de "sec-agree".

2.4 Initiation du mécanisme de sécurité

Une fois que le client a choisi un mécanisme de sécurité dans la liste reçue dans le champ d'en-tête Security-Server provenant du serveur, il initialise ce mécanisme. Les différents mécanismes exigent des procédures d'initialisation différentes.

Si "tls" est choisi, le client utilise les procédures du paragraphe 8.1.2 de la [RFC3261] pour déterminer l'URI à entrer dans les procédures du DNS de la [RFC3263]. Cependant, si l'URI est un URI SIP, il DOIT traiter le schéma comme si c'était sips, et non sip. Si le schéma d'URI n'est pas sip, la demande DOIT être envoyée en utilisant TLS.

Si "digest" est choisi, la réponse 494 (Accord de sécurité exigé) va contenir un défi d'authentification par résumé HTTP. Le client DOIT utiliser l'algorithme et les paramètres qop dans le champ d'en-tête Security-Server pour remplacer les mêmes paramètres dans le défi de résumé HTTP. Le client DOIT aussi utiliser le paramètre digest-verify dans le champ d'en-tête Security-Verify pour protéger le champ d'en-tête Security-Server comme spécifié en 2.2.

Pour utiliser "ipsec-ike", le client tente d'établir une connexion IKE avec la partie hôte de l'URI de demande dans la première demande au serveur. Si la tentative de connexion IKE échoue, la procédure d'accord DOIT être considérée comme ayant échoué, et DOIT être terminée.

Noter que "ipsec-man" ne va fonctionner que si les entités SIP communicantes savent quelles clés et autres paramètres utiliser. Il sort du domaine d'application de la présente e spécification de décrire comment ces informations peuvent être portées à la connaissance des homologues. Toutes les règles pour une mise en œuvre minimale, comme un algorithme de mise en œuvre obligatoire, s'appliquent comme défini dans les [RFC2401], [RFC2402], et [RFC2406].

Dans les deux mécanismes fondés sur IPsec, on s'attend à ce qu'aient été configurées les entrées de politique appropriées pour protéger SIP ou qu'elles seront créées avant de tenter d'utiliser la procédure d'accord de sécurité, et que les communications SIP utilisent les numéros d'accès et les adresses conformément à ces entrées de politique. Il sort du domaine d'application de la présente spécification de décrire comment ces informations peuvent être portées à la connaissance des homologues, mais elles devraient normalement être configurées en même temps qu'ont été entrés les accréditifs IKE ou les SA manuelles.

2.5 Durée des associations de sécurité

Une fois qu'un mécanisme de sécurité a été négocié, le serveur et le client ont tous deux besoin de savoir jusqu'à quand il peut être utilisé. Tous les mécanismes décrits dans le présent document ont des façons différentes de signaler la fin d'une association de sécurité. Lorsque TLS est utilisé, la terminaison de la connexion indique qu'une nouvelle négociation est nécessaire. IKE négocie la durée d'une association de sécurité. Si les accréditifs fournis par un client qui utilise un résumé ne sont plus valides, le serveur va refaire un défi au client. On suppose que lorsque IPsec-man est utilisé, le même mécanisme hors bande utilisé pour distribuer les clés est utilisé pour définir la durée de l'association de sécurité.

2.6 Résumé de l'utilisation des champs d'en-tête

Les champs d'en-tête définis dans le présent document PEUVENT être utilisés pour négocier les mécanismes de sécurité entre un UAC et d'autres entités SIP incluant UAS, mandataire, et registraire. Des informations sur l'utilisation des en-têtes en relation avec les méthodes SIP et le traitement par le mandataire sont résumées dans le Tableau 1.

Champ d'en-tête	**où**	**mandataire**	**ACK**	**BYE**	**CAN**	**INV**	**OPT**	**REG**
Security-Client	R	ard	-	o	-	o	o	o
Security-Server	421,494	-	o	-	o	o	o	
Security-Verify	R	ard	-	o	-	o	o	o

Champ d'en-tête	**où**	**mandataire**	**SUB**	**NOT**	**PRK**	**IFO**	**UPD**	**MSG**
Security-Client	R	ard	o	o	-	o	o	o
Security-Server	421,494	o	o	-	o	o	o	
Security-Verify	R	ard	o	o	-	o	o	o

Tableau 1 : Résumé de l'utilisation des champs d'en-tête

La colonne "où" décrit les types de demandes et réponse dans lesquels le champ d'en-tête PEUT être utilisé. L'en-tête PEUT ne pas apparaître dans d'autres types de messages SIP. Les valeurs dans la colonne "où" sont :
* R : le champ d'en-tête PEUT apparaître dans les demandes.
* 421, 494 : une valeur numérique indique les codes de réponse que peut utiliser le champ d'en-tête.

La colonne "mandataire" décrit les opérations que PEUT effectuer un mandataire sur un champ d'en-tête :
* a : un mandataire peut ajouter ou enchaîner le champ d'en-tête si il n'est pas présent.
* r : un mandataire DOIT être capable de lire le champ d'en-tête, et donc ce champ d'en-tête ne doit pas être chiffré.
* d : un mandataire peut supprimer une valeur de champ d'en-tête.

Les six colonnes suivantes se rapportent à la présence d'un champ d'en-tête dans une méthode :
* o : le champ d'en-tête est facultatif.

3. Rétro compatibilité

L'utilisation de cette extension dans une interface réseau est une affaire de politique locale. Des interfaces réseau différentes PEUVENT suivre des politiques différentes, et par conséquent, l'utilisation de cette extension PEUT par nature dépendre de la situation. Les mises en œuvre d'UA et de serveur DOIVENT être configurables pour fonctionner avec ou sans cette extension.

Un serveur qui est configuré à utiliser ce mécanisme PEUT aussi accepter des demandes de clients qui utilisent TLS sur la base des règles définies dans la [RFC3263]. Les demandes provenant de clients qui ne prennent pas en charge cette extension, et ne prennent pas en charge TLS, ne peuvent pas être acceptées. Cela casse évidemment l'interopérabilité avec certains clients SIP. Donc, cette extension DEVRAIT être utilisée dans des environnements où on est d'une certaine manière assuré que chaque client met en œuvre cette extension ou est capable d'utiliser TLS. Cette extension PEUT aussi être utilisée dans des environnements où une communication non sécurisée n'est pas acceptable si l'option de n'être pas capable de communiquer est aussi acceptée.

4. Exemples

Les exemples suivants illustrent l'utilisation du mécanisme défini ci-dessus.

4.1 Initié par le client

Un UA négocie le mécanisme de sécurité à utiliser avec son mandataire de sortie sans savoir à l'avance quels mécanismes le mandataire prend en charge. La méthode OPTIONS peut être utilisée ici pour demander les capacités de sécurité du mandataire. De cette façon, la sécurité peut être initiée même avant l'envoi du premier INVITE via le mandataire.

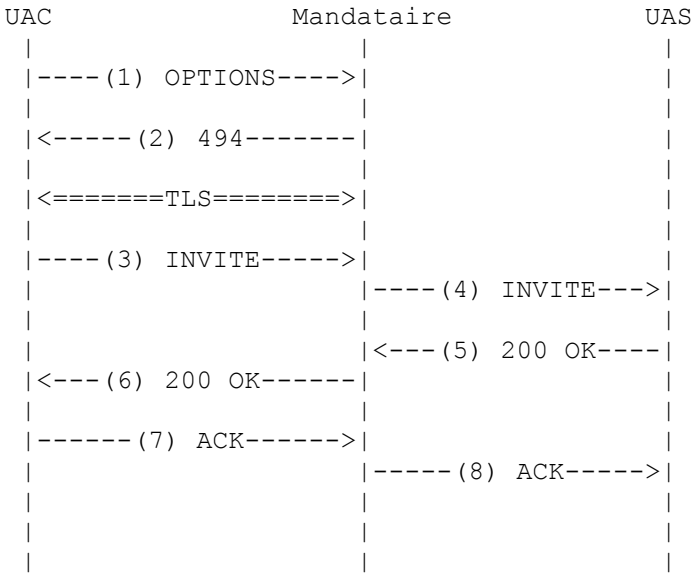

Figure 2 : Négociation initiée par le client

L'UAC envoie une demande OPTIONS à son mandataire de sortie, indiquant en même temps qu'il est capable de négocier les mécanismes de sécurité et qu'il prend en charge TLS et HTTP Digest (1).

Le mandataire de sortie répond à l'UAC par sa propre liste de mécanismes de sécurité - IPsec et TLS (2).Le seul mécanisme de sécurité commun est TLS, de sorte qu'ils établissent une connexion TLS entre eux. Lorsque la connexion est bien établie; l'UAC envoie une demande INVITE sur la connexion TLS qui vient d'être établie (3). Cet INVITE contient la liste de sécurité du serveur. Le serveur la vérifie, et comme elle correspond à sa liste statique, il traite l'INVITE et le transmet au prochain bond.

Si cet exemple était présenté sans l'en-tête Security-Server dans l'étape 2, l'UAC ne saurait pas quelle sorte de sécurité est prise en charge par l'autre, et serait forcé de faire des essais à tâtons.

Plus sérieusement, si l'en-tête Security-Verify était omis dans l'étape 3, le processus entier serait vulnérable à des attaques par interposition. Un attaquant pourrait envoyer un faux message "Accès ICMP injoignable" en réponse aux essais, ou retirer une plus forte option de sécurité de l'en-tête dans l'étape 1, réduisant ainsi de façon substantielle la sécurité.

(1) OPTIONS sip:proxy.example.com SIP/2.0
Security-Client: tls
Security-Client: digest
Require: sec-agree
Proxy-Require: sec-agree

(2) SIP/2.0 494 Security Agreement Required
Security-Server: ipsec-ike;q=0.1
Security-Server: tls;q=0.2

(3) INVITE sip:proxy.example.com SIP/2.0
Security-Verify: ipsec-ike;q=0.1
Security-Verify: tls;q=0.2
Route: sip:callee@domain.com
Require: sec-agree
Proxy-Require: sec-agree

La réponse 200 OK (6) à l'INVITE et le ACK (7) sont aussi envoyés sur la connexion TLS.

4.2 Initié par le serveur

Dans cet exemple de la Figure 3 le client envoie un INVITE vers l'appelé en utilisant un mandataire de sortie. Cet INVITE ne contient aucun champ d'en-tête Require.

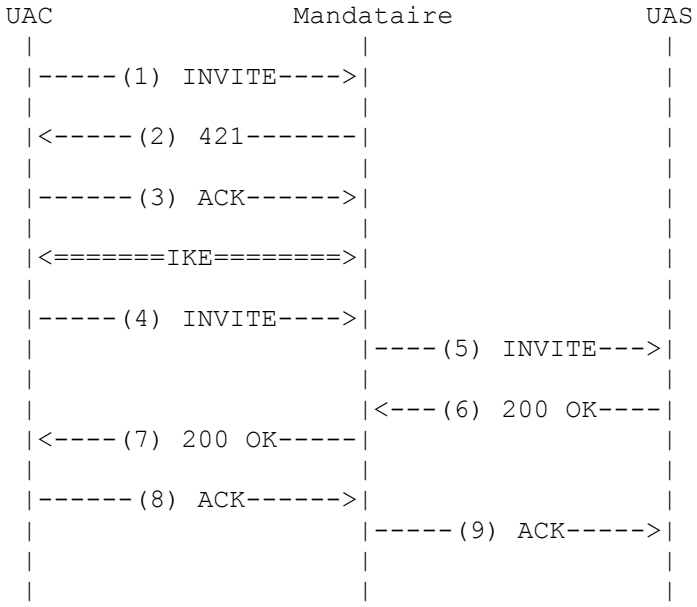

Figure 3 : Négociation de sécurité initiée par le serveur

Le mandataire, conformément à sa politique locale, n'accepte pas l'INVITE. Il retourne un code 421 (Extension exigée) avec un champ d'en-tête Security-Server qui mentionne IPsec-IKE et TLS. Comme l'UAC prend en charge IPsec-IKE, il effectue l'échange de clés et établit une association de sécurité avec le mandataire.

Le second INVITE (4) contient un champ d'en-tête Security-Verify qui reflète le champ d'en-tête Security-Server reçu dans le 421. Le INVITE (4), le 200 OK (7) et le ACK (8) sont envoyés en utilisant l'association de sécurité qui a été établie.

(1) INVITE sip:uas.example.com SIP/2.0

(2) SIP/2.0 421 Extension exigée
Security-Server: ipsec-ike;q=0.1
Security-Server: tls;q=0.2

(4) INVITE sip:uas.example.com SIP/2.0
Security-Verify: ipsec-ike;q=0.1
Security-Verify: tls;q=0.2

5. Considérations sur la sécurité

La présente spécification traite de la possibilité de choisir de façon sûre entre divers mécanismes de sécurité SIP. En particulier, la méthode présentée ici permet l'utilisation des réseaux actuels, par exemple, avec une adaptation sûre au résumé HTTP, par exemple avec IPsec, sans exiger une modification simultanée de tous les équipements. La méthode présentée dans cette spécification n'est sûre que si le plus faible mécanisme proposé offre au moins la protection de l'intégrité et contre la répétition pour le champ d'en-tête Security-Verify.

Les implications de cela pour la sécurité sont subtiles, mais ont une importance fondamentale dans la construction de grands réseaux qui changent au fil du temps. Étant donné que les hachages sont aussi produits en utilisant des algorithmes acceptés dans les premiers messages non protégés, on pourrait se demander quelle différence cela fait pour la sécurité. En supposant que la protection de l'intégrité est obligatoire et que seuls des algorithmes sûrs sont utilisés, on a quand même besoin d'empêcher les attaques par interposition de modifier d'autres paramètres, comme si le chiffrement est ou non fourni. Supposons d'abord deux homologues capables d'utiliser une sécurité forte et faible. Si les offres initiales ne sont pas du tout protégées, tout attaquant peut facilement "dégrader" les offres en retirant les options fortes. Cela forcerait les deux homologues à utiliser entre eux la sécurité faible. Mais si les offres sont protégées d'une certaine façon – comme par un hachage, ou en les répétant plus tard lorsque la sécurité choisie est réellement activée -- la situation est différente. Il ne serait pas suffisant pour l'attaquant de modifier un seul message. L'attaquant devrait plutôt modifier à la fois le message d'offre et le message qui contient le hachage/répétition. Plus important, l'attaquant aurait à inventer la sécurité faible qui est présente dans le second message, et devrait le faire en temps réel entre les offres envoyées et les messages ultérieurs. Autrement, les homologues remarqueraient que le hachage est incorrect. Si l'attaquant est capable de caser la sécurité faible, la méthode de sécurité et/ou l'algorithme NE DEVRAIENT PAS être utilisés.

En conclusion, la différence de sécurité rend possible une attaque triviale par opposition a obliger l'attaquant à casser les algorithmes. Un exemple de cas où cela a des conséquences sérieuses est lorsque un réseau est déployé avec une protection d'intégrité (comme le résumé HTTP de la [RFC2617]) et qu'ensuite de nouveaux appareils sont ajoutés qui prennent aussi en charge le chiffrement (comme TLS [RFC2246]). Dans cette situation, une procédure de négociation non sécurisée permet à des attaquants de forcer de façon triviale les nouveaux appareils à n'utiliser que la protection d'intégrité.

Les attaques possibles contre l'accord de sécurité incluent que :

1. Les attaquants pourraient essayer de modifier la liste des mécanismes de sécurité du serveur dans la première réponse. Cela serait révélé au serveur lorsque le client retourne la liste reçue en utilisant la sécurité.
2. Les attaquants pourraient aussi essayer de modifier la liste répétée dans la seconde demande provenant du client. Cependant, si le mécanisme de sécurité choisi utilise le chiffrement, cela PEUT n'être pas possible, et si il utilise la protection d'intégrité, toute modification sera détectée par le serveur.
3. Les attaquants pourraient essayer de modifier la liste des mécanismes de sécurité du client dans le premier message. Le client choisit le mécanisme de sécurité sur la base de sa connaissance de ses propres capacités et de la liste du serveur, donc, le choix du client ne sera pas affecté par une telle modification. Cependant, le choix du serveur pourrait quand même être affecté par ce qui suit :
   * Si la modification a affecté le choix du serveur, serveur et client vont finir par choisir des mécanismes de sécurité différents aux étapes 3 ou 4 de la Figure 1. Comme ils seront incapables de communiquer l'un avec l'autre, cela va être détecté comme une attaque potentielle. Dans cette situation, le client va soit réessayer, soit abandonner.
   * Si la modification n'a pas affecté le choix du serveur, il n'y a pas d'effet.
4. Finalement, les attaquants PEUVENT aussi essayer de répéter de vieux messages d'accord de sécurité. Chaque

mécanisme de sécurité DOIT fournir la protection contre la répétition. En particulier, les mises en œuvre du résumé HTTP DEVRAIENT utiliser avec soin les options existantes de protection de réponse comme d'inclure un horodatage au paramètre de nom occasionnel, et d'utiliser des compteurs de nom occasionnel [RFC2617].

Tous les clients qui mettent en œuvre la présente spécification DOIVENT choisir HTTP Digest, TLS, IPsec, ou toute méthode plus forte pour la protection de la seconde demande.

6. Considérations relatives à l'IANA

La présente spécification définit un nouvel espace de nom de nom de mécanisme au paragraphe 2.2 qui exige un corps de coordination central. L'organisme responsable de cette coordination est l'Autorité d'allocation des numéros de l'Internet (IANA, *Internet Assigned Numbers Authority*).

Le présent document définit quatre noms de mécanisme à enregistrer initialement, à savoir "digest", "tls", "ipsec-ike", et "ipsec-man". En plus de ces noms de mécanismes, le nom de mécanisme "ipsec-3gpp" est aussi enregistré (voir l'Appendice A). Suivant les politiques mentionnées dans la [RFC2434], d'autres noms de mécanisme sont alloués sur la base du consensus de l'IETF.

Les enregistrements auprès de l'IANA DOIVENT inclure le jeton mechanism-name qui est enregistré, et un pointeur sur une RFC publiée qui décrit les détails du mécanisme de sécurité correspondant.

6.1 Informations pour l'enregistrement

L'IANA enregistre les nouveaux noms de mécanismes à http://www.iana.org/assignments/sip-parameters sous "Security Mechanism Names". Comme le présent document spécifie cinq noms de mécanismes, l'enregistrement initial de l'IANA pour les noms de mécanismes va contenir les informations montrées au Tableau 2. Il montre aussi le type d'informations tenues par l'IANA.

Nom de mécanisme :	**Référence :**
digest	[RFC3329]
tls	[RFC3329]
ipsec-ike	[RFC3329]
ipsec-man	[RFC3329]
ipsec-3gpp	[RFC3329]

Tableau 2 : Enregistrement initial de l'IANA

6.2 Gabarit d'enregistrement

To : ietf-sip-sec-agree-mechanism-name@iana.org
Objet : Enregistrement d'un nouveau mécanisme d'accord de sécurité SIP
Nom du mécanisme : (Valeur de jeton conforme à la syntaxe décrite au paragraphe 2.2.)
Spécifications publiées : (Les descriptions de nouveaux mécanismes d'accord de sécurité SIP exigent la publication d'une RFC.)

6.3 Noms de champ d'en-tête

La présente spécification enregistre trois nouveaux champs d'en-tête, à savoir Security-Client, Security-Server et Security-Verify. Ces en-têtes sont définis par les informations suivantes, qui ont été incluses dans le sous registre des en-têtes SIP sous http://www.iana.org/assignments/sip-parameters.

Nom d'en-tête : Security-Client
Forme compacte : (aucune)

Nom d'en-tête : Security-Server
Forme compacte : (aucune)

Nom d'en-tête : Security-Verify
Forme compacte : (aucune)

6.4 Codes de réponse

La présente spécification enregistre un nouveau code de réponse, à savoir 494 (Accord de sécurité exigé). Le code de réponse est défini par les informations suivantes, qui ont été incluses au sous registre des méthodes et codes de réponse SIP sous http://www.iana.org/assignments/sip-parameters.

Numéro de code de réponse : 494
Phrase de cause par défaut : Accord de sécurité exigé

6.5 Étiquettes d'option

La présente spécification définit une nouvelle étiquette d'option, à savoir sec-agree. L'étiquette d'option est définie par les information suivantes, qui ont été incluses dans le sous registre des étiquettes d'option sous http://www.iana.org/assignments/sip-parameters .

Nom : sec-agree
Description : Cette étiquette d'option indique la prise en charge du mécanisme d'accord de sécurité. Lorsque elle est utilisée dans les en-têtes Require, ou Proxy-Require, elle indique que les serveurs mandataires sont obligés d'utiliser le mécanisme d'accord de sécurité. Lorsque elle est utilisée dans l'en-tête Supported, elle indique que le client d'agent d'utilisateur prend en charge le mécanisme d'accord de sécurité. Lorsque elle est utilisée dans l'en-tête Require dans la réponse 494 (Accord de sécurité exigé) ou 421 (Extension exigée) elle indique que le client d'agent d'utilisateur DOIT utiliser le mécanisme d'accord de sécurité.

7. Contributeurs

Sanjoy Sen et Lee Valerius de Nortel Networks ont contribué à ce document.

8. Remerciements

En plus des contributeurs, les auteurs souhaitent remercier Allison Mankin, Rolf Blom, James Undery, Jonathan Rosenberg, Hugh Shieh, Gunther Horn, Krister Boman, David Castellanos-Zamora, Miguel Garcia, Valtteri Niemi, Martin Euchner, Eric Rescorla et les membres du groupe 3GPP SA3 des intéressantes discussions sur cette problématique.

9. Références normatives

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

[RFC2246] T. Dierks et C. Allen, "Protocole TLS version 1.0", janvier 1999.

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

[RFC2402] S. Kent et R. Atkinson, "En-tête d'authentification IP", novembre 1998. (*Obsolète, voir* RFC4302, 4305)

[RFC2406] S. Kent et R. Atkinson, "Encapsulation de charge utile de sécurité IP (ESP)", novembre 1998. *(Obsolète, voir* RFC4303*)*

[RFC2409] D. Harkins et D. Carrel, "L'échange de clés Internet (IKE)", novembre 1998. *(Obsolète, voir la* RFC4306*)*

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

[RFC2617] J. Franks et autres, "Authentification HTTP : Authentification d'accès de base et par résumé", RFC 2617, juin 1999. *(DS.)*

[RFC3261] J. Rosenberg et autres, "SIP : Protocole d'initialisation de session", juin 2002. (*Mise à jour par* RFC3265, RFC3853, RFC4320, RFC4916, RFC5393, RFC6665)

[RFC3263] J. Rosenberg, H. Schulzrinne, "Protocole d'initialisation de session (SIP) : Localisation des serveurs SIP",

juin 2002. (*Remplace* RFC2543) *(P.S.)*

10. Références pour information

[RFC2403] C. Madson, R. Glenn, "Utilisation de HMAC-MD5-96 au sein d'ESP et d'AH", novembre 1998. *(P.S.)*

[RFC2404] C. Madson, R. Glenn, "Utilisation de HMAC-SHA-1-96 au sein d'ESP et d'AH", novembre 1998. *(P.S.)*

[RFC2451] R. Pereira, R. Adams, "Algorithmes de chiffrement ESP en mode CBC", novembre 1998. *(P.S.)*

[RFC4083] M. Garcia-Martin, "Exigences du projet en partenariat de troisième génération (3GPP) pour la version 5 des entrées du protocole d'initialisation de session (SIP)", mai 2005. *(Information)*

[TS33.203] 3rd Generation Partnership Project, "Access security for IP-based services, Release 5", TS 33.203 v5.3.0, septembre 2002.

Appendice A Syntaxe de ipsec-3gpp

Le présent Appendice étend le cadre d'accord de sécurité décrit dans le présent document à un nouveau mécanisme de sécurité : "ipsec-3gpp". Ce mécanisme de sécurité et ses paramètres associés sont utilisés dans le sous-système multimédia IP 3GPP [TS33.203]. Les définitions de BNF augmenté ci-dessous suivent la syntaxe de SIP [RFC3261].

```
mechanism-name   = ( "ipsec-3gpp" )
mech-parameters    = ( algorithm / protocol /mode / encrypt-algorithm / spi / port1 / port2 )
algorithm          = "alg" EQUAL ( "hmac-md5-96" / "hmac-sha-1-96" )
protocol           = "prot" EQUAL ( "ah" / "esp" )
mode               = "mod" EQUAL ( "trans" / "tun" )
encrypt-algorithm  = "ealg" EQUAL ( "des-ede3-cbc" / "null" )
spi                = "spi" EQUAL spivalue
spivalue           = 1*10DIGIT; 0 to 4294967295
port1              = "port1" EQUAL port
port2              = "port2" EQUAL port
port               = 1*DIGIT
```

Les paramètres décrit par le BNF ci-dessus ont la sémantique suivante :

Algorithm
Ce paramètre définit l'algorithme d'authentification utilisé. Il PEUT avoir une valeur de "hmac-md5-96" pour HMAC-MD5-96 [RFC2403], ou de "hmac-sha-1-96" pour HMAC-SHA-1-96 [RFC2404]. Le paramètre algorithm est obligatoire.

Protocol
Ce paramètre définit le protocole IPsec. Il PEUT avoir une valeur de "ah" pour AH [RFC2402], ou de "esp" pour ESP [RFC2406]. Si le paramètre Protocol n'est pas présent, le protocole sera ESP par défaut.

Mode
Ce paramètre définit le mode dans lequel le protocole IPsec est utilisé. Il PEUT avoir une valeur de "trans" pour le mode transport, ou une valeur de "tun" pour le mode tunnelage. Si aucun paramètre Mode n'est présent, le protocole IPsec est utilisé en mode transport.

Encrypt-algorithm
Ce paramètre définit l'algorithmz de chiffrement utilisé. Il PEUT avoir une valeur de "des-ede3-cbc" pour 3DES [RFC2451], ou de "null" pour pas de chiffrement. Si aucun paramètre Encrypt-algorithm n'est présent, le chiffrement n'est pas utilisé.

Spi : Ce paramètre définit le numéro SPI utilisé pour les messages entrants.

Port1 : Ce paramètre définit le numéro d'accès de destination pour les messages entrants qui sont protégés.

Port2 : Ce paramètre définit le numéro d'accès de source pour les messages sortants qui sont protégés. Port 2 est facultatif.

Les entités SIP communicantes ont besoin de savoir à l'avance quelles clés utiliser. On suppose aussi que la mise en œuvre IPsec sous-jacente prend en charge des sélecteurs qui permettent à tous les protocoles de transport pris en charge par SIP d'être protégés avec une seule SA. La durée de l'association de sécurité est la même que dans l'intervalle d'expiration du lien d'enregistrement correspondant.

Adresse des auteurs

Jari Arkko
Ericsson
Jorvas, FIN 02420
Finlet
téléphone : +358 40 507 9256
mél : jari.arkko@ericsson.com

Vesa Torvinen
Ericsson
Joukahaisenkatu 1
Turku, FIN 20520
Finlet
téléphone : +358 40 723 0822
mél : vesa.torvinen@ericsson.fi

Gonzalo Camarillo
Advanced Signalling Research Lab.
Ericsson
FIN-02420 Jorvas
Finlet
téléphone : +358 40 702 3535
mél : Gonzalo.Camarillo@ericsson.com

Aki Niemi
NOKIA Corporation
P.O.Box 321, FIN 00380
Finlet
téléphone : +358 50 389 1644
mél : aki.niemi@nokia.com

Tao Haukka
Nokia Corporation
P.O. Box 50
FIN - 90570 Oulu
Finlet
téléphone : +358 40 517 0079
mél : tao.haukka@nokia.com

Déclaration complète de droits de reproduction

Remerciement

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