Groupe de travail Réseau
Request for Comments : 4567
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

J. Arkko, F. Lindholm, M. Naslund & K. Norrman, Ericsson
E. Carrara, Royal Institute of Technology
juillet 2006

Extensions de gestion de clé pour le protocole de description de session (SDP) et le protocole de flux directs en temps réel (RTSP)

Statut du présent mémoire
Le présent document spécifie un protocole de normalisation Internet pour la communauté Internet, et appelle à discussion et suggestions en vue de son amélioration. Prière de se rapporter à l'édition en cours des "Internet Official Protocol Standards" (normes officielles du protocole Internet) (STD 1) pour connaître l'état de la normalisation et le statut du présent protocole. La distribution du présent mémoire n'est soumise à aucune restriction.
(La présente traduction incorpore les errata 56, 809, et 2247)

Résumé
Le présent document définit des extensions générales pour le protocole de description de session (SDP, *Session Description Protocol*) et le protocole de flux directs en temps réel (RTSP, *Real Time Streaming Protocol*) pour porter les messages, comme spécifié par un protocole de gestion de clés, afin de sécuriser le support. Ces extensions sont présentées comme un cadre, pour être utilisé par un ou plusieurs protocoles de gestion de clés. À ce titre, leur utilisation n'a de signification que complétée par un protocole de gestion de clés approprié.

Des lignes directrices générales sont aussi données sur la façon dont le cadre devrait être utilisé avec SIP et SAP. L'utilisation avec le protocole de gestion de clés pour l'Internet (MIKEY, *Multimedia Internet KEYing*) est aussi définie.

Table des matières

1. Introduction

Des travaux récents ont défini un profil de sécurité pour la protection des applications en temps réel fonctionnant sur RTP, [RFC3711]. Cependant, un protocole de sécurité a besoin d'une solution de gestion de clés pour échanger les clés et les paramètres de sécurité, gérer et rafraîchir les clés, etc.

Un protocole de gestion de clés est exécuté avant l'exécution du protocole de sécurité. Le but principal du protocole de gestion de clés est d'établir, d'une façon sûre et fiable, une association de sécurité pour le protocole de sécurité. Cela inclut une ou plusieurs clés de chiffrement et l'ensemble des paramètres nécessaires pour le protocole de sécurité, par exemple, les algorithmes de chiffrement et d'authentification à utiliser. Le protocole de gestion de clés a des similitudes avec, par exemple, SIP [RFC3261] et RTSP [RFC2326] en ce sens qu'il négocie les informations nécessaires afin d'être capable d'établir la session.

Les sections qui suivent se concentrent sur la description d'un nouvel attribut SDP et une extension d'en-tête RTSP pour prendre en charge la gestion de clés, et montrent comment cela peut être intégré à SIP et RTSP. Le cadre résultant est complété par un ou plusieurs protocoles de gestion de clés, qui utilisent les extensions fournies.

Certaines des raisons de la création d'un cadre avec la possibilité d'inclure la gestion de clés dans l'établissement de session sont :

* Tout comme les informations de codec sont une description de comment coder et décoder le flux audio (ou vidéo) les données de gestion de clés sont une description de comment chiffrer et déchiffrer les données.

* La possibilité de négocier la sécurité pour la session multimédia entière en même temps.

* La connaissance du support à l'établissement de la session rend facile de lier la gestion de clés à la session multimédia.

* Cette approche peut être plus efficace que d'établir la sécurité plus tard, car cette approche peut forcer à des allers-retours supplémentaires, éventuellement aussi un établissement séparé pour chaque flux, impliquant donc plus de délais pour l'établissement réel de la session de supports.

* La possibilité de négocier le matériel de chiffrement de bout en bout sans appliquer de protection d'extrémité à extrémité du SDP (des mécanismes de sécurité bond par bond peuvent être utilisés à la place, ce qui peut être utile si des mandataires intermédiaires ont besoin d'accéder au SDP).

Il existe actuellement dans SDP [RFC4566] un champ pour transporter les clés, le champ "k=". Cependant, ce n'est pas suffisant pour un protocole de gestion de clés car il y a beaucoup plus de paramètres à transporter, et le champ "k=" n'est pas extensible. L'approche utilisée est d'étendre la description SDP par un certain nombre d'attributs qui transportent l'offre/réponse de gestion de clés et aussi pour l'associer aux sessions de supports. SIP utilise le modèle d'offre/réponse [RFC3264] qui va suffire aux extensions à SDP. Cependant, RTSP [RFC2326] n'utilise pas le modèle offre/réponse avec SDP, de sorte qu'un nouvel en-tête RTSP est introduit pour porter les données de gestion de clés. La [RFC4568] utilise l'approche d'étendre SDP, pour porter les paramètres de sécurité pour les flux de supports. Cependant, le mécanisme défini dans la [RFC4568] exige la protection de SDP de bout en bout par un protocole de sécurité comme S/MIME, afin de d'obtenir la protection de bout en bout. La solution décrite ici ne concerne que la protection de bout en bout des paramètres de gestion de clés et par conséquent n'exige pas de moyen de protection externe de bout en bout. Il est important de noter que seuls les paramètres de gestion de clés sont protégés.

Ce document définit aussi l'utilisation du cadre décrit avec le protocole de gestion de clés multimédia Internet (MIKEY) [RFC3830].

1.1 Conventions de notation

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

2. Applicabilité

La [RFC4568] fournit des capacités similaires de distribution de clés de chiffrement, et elle est destinée à être utilisée lorsque le matériel de chiffrement est protégé en même temps que la signalisation.

À l'opposé, la présente spécification attend des points d'extrémité qu'ils aient des clés préconfigurées ou une infrastructure de sécurité commune. Elle fournit sa propre sécurité et est indépendante de la protection de la signalisation (si il en est). Par suite, elle peut être appliquée dans des environnements où la protection de la signalisation n'est pas activée, ou est utilisée bond par bond (c'est-à-dire, des scénarios où le SDP n'est pas protégé de bout en bout). La présente spécification va assurer, indépendamment de la protection de signalisation appliquée, l'établissement de la sécurité de bout en bout pour le support.

3. Extensions à SDP et RTSP

Cette Section décrit les attributs communs qui peuvent être inclus dans SDP ou RTSP quand un protocole de gestion de clés intégré est utilisé. Les valeurs d'attribut suivent les lignes directrices générales de SDP et RTSP (voir les [RFC4566] et [RFC2326]).

Pour SDP et RTSP, la méthode générale d'ajout du protocole de gestion de clés est d'introduire de nouveaux attributs, un identifiant pour identifier le protocole de gestion de clés spécifique, et un champ de données où sont placées les données du protocole de gestion de clés. Les données du protocole de gestion de clés contiennent les informations nécessaires pour établir le protocole de sécurité, par exemple, les clés et les paramètres de chiffrement. Tous les paramètres et clés sont protégés par le protocole de gestion de clés.

Les données de gestion de clés DOIVENT être codées en base64 [RFC3548] et se conformer à la grammaire base64 définie dans la [RFC4566]. L'identifiant de protocole de gestion de clés, KMPID, est défini comme ci-dessous dans la grammaire du format Backus-Naur augmenté (ABNF) [RFC4234].

KMPID = 1*(ALPHA / CHIFFRE)

Les valeurs pour l'identifiant, KMPID, sont enregistrées et définies conformément à la Section 9. Noter que le KMPID est sensible à la casse, et il est RECOMMANDÉ que les valeurs enregistrées soient des lettres minuscules.

3.1 Extensions à SDP

Ce paragraphe donne la grammaire ABNF (comme utilisée dans la [RFC4566]) pour les extensions de gestion de clés à SDP.

Noter que les nouvelles définitions sont conformes à la définition d'un champ d'attribut, c'est-à-dire,

attribut = (att-field ":" att-value) / att-field

L'ABNF pour les extensions de gestion de clés (conformes à att-field et att-value) est le suivant :

key-mgmt-attribute = key-mgmt-att-field ":" key-mgmt-att-value

key-mgmt-att-field = "key-mgmt"
key-mgmt-att-value = 0*1SP prtcl-id SP keymgmt-data

prtcl-id = KMPID ; par exemple, "mikey"

keymgmt-data = base64
SP = %x20

où KMPID est défini à la Section 3 de ce mémoire, et base64 est défini dans SDP [RFC4566]. Prtcl-id se réfère à l'ensemble de valeurs définies pour KMPID à la Section 9.

L'attribut PEUT être utilisé au niveau session, au niveau support, ou aux deux niveaux. Un attribut défini au niveau support outrepasse un attribut défini au niveau session. En d'autres termes, si l'attribut de niveau support est présent, l'attribut de niveau session DOIT être ignoré pour ce support. Le paragraphe 4.1 décrit en détails comment les attributs sont utilisés et comment SDP est traité dans les différents scénarios d'usage. Le choix du niveau dépend, par exemple, du protocole de

gestion de clés particulier. Certains protocoles peuvent n'être pas capables de déduire assez de matériel de chiffrement pour toutes les sessions ; de plus, une protection différente pour chaque session pourrait éventuellement être exigée. Le protocole concerné ne pourrait faire cela qu'en le spécifiant au niveau support. D'autres protocoles, comme MIKEY, ont ces capacités (car il peuvent exprimer plusieurs politiques de sécurité et déduire plusieurs clés) de sorte qu'ils peuvent utiliser le niveau session.

3.2 Extensions à RTSP

Pour prendre en charge les attributs de gestion de clés, l'en-tête RTSP suivant est défini :

KeyMgmt = "KeyMgmt" ":" key-mgmt-spec 0*("," key-mgmt-spec)

key-mgmt-spec = "prot" "=" KMPID ";" ["uri" "=" %x22 URI %x22 ";"] ["data" "=" %22 base64 %22 ";"]

où KMPID est défini à la Section 3 du présent mémoire, "base64" est défini dans la [RFC4566], et "URI" est défini à la Section 3 de la [RFC3986].

Le paramètre "uri" identifie le contexte pour lequel s'appliquent les données de gestion de clés, et l'URI RTSP DEVRA correspondre à un URI (de session ou de support) présent dans la description de la session. Si l'URI de contrôle agrégé RTSP est inclus, il indique que le message de gestion de clés est au niveau session (et de façon similaire l'URI de contrôle de support RTSP qu'il s'applique au niveau support). Si aucun paramètre "uri" n'est présent dans une spécification de gestion de clé *(key-mgmt-spec)* la spécification s'applique au contexte identifié par l'URI de demande RTSP.

L'en-tête KeyMgmt PEUT être utilisé dans les messages et directions décrits dans le tableau ci-dessous :

Méthode	**Direction**	**Exigence**
Réponse DESCRIBE	S->C	RECOMMANDÉ
SETUP	C->S	EXIGÉ
Réponse SETUP	S->C	EXIGÉ (erreur)

Note : le paragraphe 4.2 décrit en détails comment sont utilisées les extensions RTSP.

On définit un nouveau code d'état RTSP pour rapporter une erreur due à une défaillance durant le traitement de la gestion de clés (paragraphe 4.2):

Code d'état = "463" ; défaillance de gestion de clé

Une réponse 463 PEUT contenir un en-tête Gestion de clé *(KeyMgmt)* avec un message de protocole de gestion de clés qui précise la nature de l'erreur.

4. Usage avec SDP, SIP, RTSP, et SAP

Cette Section donne les règles et recommandations sur la façon et le moment d'inclure l'attribut défini de gestion de clés quand SIP et/ou RTSP sont utilisés avec SDP.

Quand un protocole de gestion de clés est intégré à SIP/SDP et RTSP, les exigences générales suivantes s'appliquent à la gestion de clés:

* Pour l'instant, il DOIT être possible d'exécuter le protocole de gestion de clés sur au plus un échange de messages demande/réponse. Une future atténuation de cette exigence est possible mais introduirait une complexité significative pour les mises en œuvre qui prennent en charge des mécanismes à plusieurs allers-retours.

* Il DOIT être possible à partir de l'application SIP/SDP et RTSP, en utilisant l'API de gestion de clés, de recevoir les données et informations de gestion de clés disant si un message est accepté ou non.

Le contenu des messages de gestion de clés dépend du protocole de gestion de clés utilisé. Cependant, le contenu de tels messages de gestion de clés peut être en gros supposé comme suit : l'initiateur de la gestion de clés (par exemple, l'offreur) inclut les données de gestion de clés dans un premier message, contenant la description du support auquel elle devrait s'appliquer. Ces données consistent en général en les paramètres de sécurité (incluant le matériel de chiffrement)

nécessaires pour sécuriser la communication, ainsi que les informations nécessaires à l'authentification (pour s'assurer que le message est authentique).

Du côté du répondant, le protocole de gestion de clés vérifie la validité du message de gestion de clés, ainsi que la disponibilité des paramètres offerts, et ensuite fournit les données de gestion de clés à inclure dans la réponse. Cette réponse peut normalement authentifier le répondant à l'initiateur, et aussi déclarer si l'offre initiale a été ou non acceptée. Certains protocoles peuvent exiger que le répondant inclue un choix des paramètres de sécurité qu'il veut prendre en charge. Là encore, le contenu réel de telles réponses dépend du protocole de gestion de clés.

La Section 7 décrit une réalisation du protocole MIKEY qui utilise ces mécanismes. Les procédures à utiliser lors de la transposition de nouveaux protocoles de gestion de clés dans ce cadre sont décrites à la Section 6.

4.1 Utilisation de SDP

Cette section décrit les règles de traitement des différentes applications qui utilisent SDP pour la gestion de clés.

4.1.1 Traitement général

Le traitement quand SDP est utilisé est légèrement différent selon la façon dont SDP est transporté, de si il utilise un modèle d'offre/réponse ou une annonce. Le traitement peut être divisé en quatre étapes différentes :

1) comment créer l'offre initiale,
2) comment traiter une offre reçue,
3) comment créer une réponse,
4) comment traiter une réponse reçue.

On notera que les deux dernières étapes ne peuvent pas toujours être applicables, car il y a des cas ou une réponse ne peut pas ou ne veut pas être renvoyée.

Le traitement général pour créer une offre initiale DEVRA suivre les actions suivantes :

* L'identifiant du protocole de gestion de clés utilisé DOIT être placé dans le champ prtcl-id de SDP. Un tableau des identifiants de protocoles légaux est tenu par l'IANA (voir la Section 9).
* Le champ keymgmt-data DOIT être créé comme suit : le protocole de gestion de clés DOIT être utilisé pour créer le message de gestion de clés. Ce message DEVRA être codé en base64 [RFC3548] par l'application SDP et ensuite encapsulé dans l'attribut keymgmt-data. Noter cependant que la sémantique du message encapsulé dépend du protocole de gestion de clés utilisé.

Le traitement général pour une offre reçue DEVRA suivre les actions suivantes :

* Le protocole de gestion de clés est identifié selon le champ prtcl-id. Un tableau des identifiants de protocoles légaux est tenu par l'IANA (voie la Section 9).
* Les données de gestion de clés provenant du champ keymgmt-data DOIVENT être extraites, et décodées de base64 pour reconstruire le message original, et ensuite passées pour traitement au protocole de gestion de clés. Noter que selon le protocole de gestion de clés, des paramètres supplémentaires peuvent aussi être demandés par l'API spécifique, comme l'adresse/accès du réseau de source/destination pour le support spécifié (cependant, cela va être mis en œuvre selon l'API spécifique réelle). Les paramètres supplémentaires dont un protocole de gestion de clés peut avoir besoin (à part ceux définis ici) DOIVENT être documentés, en décrivant leur utilisation, ainsi que les interactions de ce protocole de gestion de clés avec SDP et RTSP.
* Si des erreurs surviennent, ou si l'offre de gestion de clés est rejetée, la session DEVRA être interrompue. Les messages d'erreur possibles dépendent du protocole spécifique d'établissement de session.

À ce stade, la gestion de clés aura accepté ou rejeté les paramètres offerts. Ceci PEUT causer un message de réponse, selon le protocole de gestion de clés et le scénario d'application.

Si une réponse doit être générée, les actions générales suivantes DEVRONT être effectuées :

* L'identifiant du protocole de gestion de clés utilisé DOIT être placé dans le champ prtcl-id.
* Le champ keymgmt-data DOIT être créé comme suit : le protocole de gestion de clés DOIT être utilisé pour créer le message de gestion de clés. Ce message DEVRA être codé en base64 [RFC3548] par l'application SDP et ensuite encapsulé dans l'attribut keymgmt-data. La sémantique du message encapsulé dépend du protocole de gestion de clés utilisé.

Le traitement général pour une réponse reçue DEVRA suivre les actions ci-après :

* Le protocole de gestion de clés est identifié conformément au champ prtcl-id.

* Les données de gestion de clés provenant du champ keymgmt-data DOIVENT être extraites, décodées de base64 pour reconstruire le message original, et ensuite passées au protocole de gestion de clés pour traitement.
* Si l'offre de gestion de clés est rejetée et si l'intention est de la renégocier, cela DOIT être fait par un autre échange offre/réponse. Il est RECOMMANDÉ de NE PAS interrompre la session dans ce cas, mais de renégocier en utilisant un autre échange offre/réponse. Par exemple, dans la [RFC3261], la "précondition de sécurité " comme définie dans la [RFC5027] résout le problème pour une initialisation de session. Les procédures de la [RFC5027] sortent du domaine d'application du présent document. Dans une session établie, un échange offre/réponse supplémentaire utilisant un re-INVITE ou UPDATE comme approprié PEUT être utilisé.
* Si des erreurs se produisent, ou si l'offre de gestion de clés est rejetée et si on n'a pas l'intention de la renégocier, la session DEVRA être interrompue. Si possible, un message d'erreur indiquant l'échec DEVRAIT être renvoyé.

Autrement, si toutes les étapes sont réussies, l'établissement normal s'effectue.

4.1. Utilisation de SDP avec offre/réponse et SIP

Ce paragraphe définit des règles de traitement supplémentaires qui s'ajoutent aux règles générales définies au paragraphe 4.1.1, applicables seulement aux applications qui utilisent SDP avec le modèle offre/réponse de la [RFC3264] (et en particulier SIP).

Quand une offre initiale est créée, la procédure spécifique d'offre/réponse DEVRA être appliquée :
* avant de créer le champ de données de gestion de clés, la liste des identifiants de protocole DOIT être fournie par l'application SDP à chaque protocole de gestion de clés, comme défini au paragraphe 4.1.4 (pour combattre les attaques en dégradation).

Pour une offre SDP reçue qui contient les attributs de gestion de clés, la procédure spécifique d'offre/réponse DEVRA être appliquée :
* avant le, ou conjointement au, passage des données de gestion de clés au protocole de gestion de clés, la liste complète des identifiants de protocole provenant du message d'offre est fournie par l'application SDP au protocole de gestion de clés (comme défini au paragraphe 4.1.4).

Quand une réponse est créée, la procédure spécifique d'offre/réponse DEVRA être appliquée :
* Si la gestion de clés rejette l'offre et si l'intention est de la renégocier, la réponse DEVRAIT inclure la cause de l'échec dans un message inclus provenant du protocole de gestion de clés. La renégociation DOIT être faite par un autre échange offre/réponse (par exemple, en utilisant la [RFC5027]). Dans une session établie, elle peut aussi être faite par un re-INVITE ou UPDATE comme approprié.
* Si la gestion de clés rejette l'offre et si la session doit être interrompue, le répondant DEVRAIT retourner un message "488 Non acceptable ici", incluant aussi facultativement un ou plusieurs en-têtes d'avertissement (un "306 Attribut non compris" quand un des paramètres n'est pas pris en charge, et un "399 Avertissements divers" avec des informations arbitraires à présenter à un utilisateur humain ou enregistrées ; voir le paragraphe 20.43 de la [RFC3261]). D'autres détails sur la cause de l'échec PEUVENT être décrits dans un message inclus provenant du protocole de gestion de clés. La session est alors interrompue (et il appartient à la politique locale ou à l'utilisateur final de décider comment continuer).

Noter que l'attribut gestion de clés (relatif au même protocole de gestion de clés) PEUT être présent au niveau session et au niveau support. Par conséquent, le processus DEVRA être répété pour chaque attribut gestion de clés détecté. En cas d'échec du traitement par la gestion de clés d'un de ces attributs (par exemple, échec d'authentification, paramètres non pris en charge, etc.) au niveau session ou support, l'établissement de la session entière DEVRA être interrompu, y compris les parties de la session qui avaient achevé avec succès leur part de la gestion de clés.

Si plus d'un protocole de gestion de clés est pris en charge, plusieurs instances de l'attribut gestion de clés PEUVENT être incluses dans l'offre initiale quand on utilise le modèle offre/réponse, chacune transportant un protocole de gestion de clés différent, indiquant donc les solutions de remplacement prises en charge.

Si l'offreur inclut plus d'un attribut protocole de gestion de clés au niveau session (même chose au niveau support) ceux-ci DEVRAIENT être cités dans l'ordre de préférence (le premier étant le préféré). Celui qui répond choisit le protocole de gestion de clés qu'il souhaite utiliser, et ne traite que lui, aussi bien au niveau session qu'au niveau support, selon sa localisation. Si celui qui répond ne prend en charge aucun des protocoles de gestion de clés suggérés par l'offreur, il l'indique à l'offreur afin qu'une nouvelle offre/réponse puisse être déclenchée ; autrement, il peut retourner un message d'erreur "488 Non acceptable ici", suivant lequel l'envoyeur DOIT interrompre la procédure d'établissement en cours.

Noter que le placement de multiples offres de gestion de clés dans un seul message présente l'inconvénient d'allonger le message et que la charge de calcul pour l'offreur va augmenter de façon considérable. Sauf à suivre les lignes directrices du

paragraphe 4.1.4, des lignes multiples peuvent ouvrir la voie à des attaques en dégradation. Noter aussi que l'option d'offre multiple a été ajoutée pour optimiser les frais généraux de signalisation dans le cas où l'initiateur connaît certaines clés (par exemple, une clé publique) qu'a celui qui répond, mais n'est pas sûr du protocole que celui qui répond prend en charge. Le mécanisme n'est pas destiné à négocier les options au sein d'un seul et même protocole.

L'offreur DOIT inclure les données de gestion de clés dans une offre qui contient la description du support à laquelle elle s'applique.

Le changement de clés DOIT être traité comme une nouvelle offre, avec les nouveaux paramètres proposés. Celui qui répond traite cela comme une nouvelle offre où la gestion de clés est le sujet du changement. L'échange de changement de clés DOIT être finalisé avant que le protocole de sécurité puisse changer les clés. Le même protocole de gestion de clés utilisé dans l'offre originale DEVRA aussi être utilisé dans la nouvelle offre portant le changement de clés. Si la nouvelle offre portant le changement de clés échoue (par exemple, la vérification de l'authentification échoue) celui qui répond DEVRAIT envoyer un message "488 Non acceptable ici", incluant un ou plusieurs en-têtes d'avertissement (au moins un 306). L'offreur DOIT alors interrompre la session.

Noter que, dans les scénarios de diffusion groupée, à la différence de l'envoi individuel, il n'y a qu'une seule vue du flux [RFC3264], donc il DOIT y avoir un accord complet sur les paramètres de sécurité.

Après la production de l'offre, l'offreur DEVRAIT être prêt à recevoir des supports, car ils peuvent arriver avant la réponse. Cependant, cela pose des problèmes, car l'offreur ne connaît pas encore le choix de celui qui répond en termes de, par exemple, algorithmes, éventuellement si la clé est connue. Cela peut causer des délais ou une coupure peut se produire ; si c'est inacceptable, l'offreur DEVRAIT utiliser des mécanismes qui sortent du domaine d'application du présent document, par exemple, les préconditions de sécurité pour SIP [RFC5027].

4.1.3 Utilisation de SDP avec SAP

Il y a des cas où SDP est utilisé sans se conformer au modèle d'offre/réponse ; à la place, il y a une distribution SDP unidirectionnelle (c'est-à-dire, sans canal de retour) comme quand il est utilisé avec SAP et HTTP.

Le traitement suit les deux premières étapes du traitement SDP général (voir le paragraphe 4.1.1). On peut noter que dans ce cas le traitement diffère du cas de l'offre/réponse en ce qu'un seul protocole de gestion de clés DEVRA être offert (c'est-à-dire, aucune négociation ne va être possible). Cela implique que l'attaque en dégradation n'est pas un problème ; donc, la contre mesure n'est pas nécessaire. Le protocole de gestion de clés utilisé DOIT prendre en charge les messages unidirectionnels.

4.1.4 Prévention des attaques en dégradation

La possibilité de prise en charge de plusieurs protocoles de gestion de clés peut, si elle n'est pas traitée de façon appropriée, introduire des attaques en dégradation. Spécifiquement, un interposé pourrait "éplucher" des offres à chiffrement fort (en supprimant les lignes de gestion de clés du message) laissant seulement les plus faibles comme choix de celui qui répond. Pour éviter cela, la liste des identifiants de protocoles de gestion de clés proposés DOIT être authentifiée. L'authentification DOIT être faite séparément par chaque protocole de gestion de clés.

En conséquence, il DOIT être spécifié (dans la spécification du protocole de gestion de clés lui-même ou dans un document d'accompagnement) comment la liste des identifiants de protocole de gestion de clés peut être traitée pour être authentifiée de l'offreur à celui qui répond par le protocole de gestion de clés spécifique. Noter que même si un seul protocole de gestion de clés est utilisé, il DOIT quand même authentifier son propre identifiant de protocole.

La liste des identifiants de protocoles DOIT alors être donnée à chacun des protocoles de gestion de clés proposés (offerts) par l'application avec des identifiants séparés par ";". Tous les identifiants de protocole offerts DOIVENT être inclus, dans le même ordre que celui où ils apparaissent dans la description SDP correspondante.

La liste des protocoles peut être formellement décrite par

prtcl-list = KMPID *(";" KMPID)

où KMPID est défini à la Section 3.

Par exemple, si les protocoles offerts sont MIKEY et deux protocoles restants encore à inventer KEYP1, KEYP2, le SDP est :

```
v=0
o=alice 2891092738 2891092738 IN IP4 lost.exemple.com
s=discussion secrète
t=0 0
c=IN IP4 lost.exemple.com
a=key-mgmt:mikey AQAFgM0XflABAAAAAAAAAAAAAAsAyO...
a=key-mgmt:keyp1 727gkdOshsuiSDF9sdhsdKnD/dhsoSJokdo7eWD...
a=key-mgmt:keyp2 DFsnuiSDSh9sdh Kksd/dhsoddo7eOok727gWsJD...
m=audio 39000 RTP/SAVP 98
a=rtpmap:98 AMR/8000
m=video 42000 RTP/SAVP 31
a=rtpmap:31 H261/90000
```

La liste des protocoles, "mikey;keyp1;keyp2", va être générée à partir de la description SDP et utilisée en entrée à chaque protocole de gestion de clés spécifié (avec les données pour ce protocole). Chacun des trois protocoles inclut cette liste des identifiants de protocole dans sa couverture d'authentification (conformément à la spécification de son protocole).

Si plus d'un protocole est pris en charge par l'offreur, il est RECOMMANDÉ que tous les protocoles acceptables soient inclus dans la première offre, plutôt que de faire ensuite d'autres offres sur un seul en réponse aux messages d'erreur ; voir les "Considérations sur la sécurité".

La protection d'intégrité de bout en bout des attributs key-mgmt tous ensemble, fournie en externe à la gestion de clés elle-même, protège aussi contre cette attaque en dégradation. C'est, par exemple, le cas si SIP utilise S/MIME [RFC3851] pour protéger de bout en bout l'intégrité de la description SDP. Cependant, comme cette protection de bout en bout n'est pas une hypothèse du cadre, les mécanismes définis dans cette section DEVRONT être appliqués.

4.2 Usage de RTSP

RTSP n'utilise pas le modèle offre/réponse, comme le fait SIP. Cela pose quelques problèmes, car il n'est pas possible (sans modifier RTSP) de renvoyer une réponse. Pour résoudre cela, un nouvel en-tête a été introduit (paragraphe 3.2). Cela suppose aussi que la gestion de clés a aussi une certaine forme de lien avec le support afin que la réponse au serveur soit traité comme nécessaire.

Le serveur DEVRA être l'initiateur de l'échange de gestion de clés pour les sessions en mode PLAY, c'est-à-dire, qui transportent des supports du serveur au client. Le texte qui suit décrit le comportement pour le mode PLAY. Pour tout autre mode, le comportement n'est pas défini dans la présente spécification.

Pour obtenir une description de session, le client contacte initialement le serveur via un message DESCRIBE. Le message initial de gestion de clés provenant du serveur RTSP est envoyé au client dans le SDP du 200 OK en réponse au DESCRIBE. Noter que seulement un protocole de gestion de clés DEVRA être utilisé par niveau session/support. Un serveur PEUT permettre que SDP avec des attributs de gestion de clés soit distribué au client par d'autres moyens que RTSP, mais ceci n'est pas spécifié ici.

Le paramètre "uri" de l'en-tête KeyMgmt est utilisé pour indiquer sur quel contexte le message porté s'applique pour le protocole de gestion de clés. Pour les messages de gestion de clés sur le SDP niveau session, la réponse DOIT contenir l'URI de commande agrégée RTSP pour l'indiquer. Pour les messages de gestion de clés initialement sur SDP niveau support, le message de réponse de gestion de clés dans l'en-tête KeyMgmt PEUT utiliser l'URL de RTSP niveau support. Pour les sessions RTSP qui n'utilisent pas la commande agrégée, c'est-à-dire, où aucun URI de commande de niveau session n'est défini, le protocole de gestion de clés DEVRA seulement être invoqué sur des flux de supports individuels. Dans ce cas aussi, la réponse de gestion de clés DEVRA être sur un flux de supports individuel (c'est-à-dire, un en-tête de gestion de clés en-tête RTSP par support).

Quand il répond au message initial de gestion de clés, le client utilise le nouvel en-tête RTSP (KeyMgmt) pour renvoyer une réponse. La façon de le faire dépend du contexte d'usage :

* Les réponses au protocole de gestion de clés pour l'établissement initial des paramètres de sécurité pour une session RTSP agrégée DEVRONT être envoyées dans le premier SETUP de la session. Cela signifie que si la gestion de clés est déclarée pour la session entière mais est établie de façon non agrégée (c'est-à-dire, un support par session RTSP) chaque SETUP DOIT porter la même réponse pour le contexte de niveau session. Lorsque on effectue un établissement du second support ou d'un suivant dans une session RTSP, les mêmes paramètres de gestion de clés établis pour le premier support s'appliquent aussi à ces établissements.

* Les réponses de gestion de clés pour l'établissement initial des paramètres de sécurité pour un support individuel DEVRONT seulement être incluses dans SETUP pour le flux de supports correspondant.

Si un serveur reçoit un message SETUP dans lequel il attend un message de gestion de clés, mais qu'aucun n'est inclus, un "403 Interdit" DEVRAIT être retourné au client, ce qui DOIT interrompre l'établissement en cours.

Lorsque le serveur crée un message initial SDP, la procédure DEVRA être la même que décrit au paragraphe 4.1.1.

Le client qui traite le message initial SDP provenant du serveur DEVRA suivre les mêmes procédures que décrit au paragraphe 4.1.1, excepté que, si il y a une erreur, la session est interrompue (aucune erreur n'est renvoyée).

Le client DEVRA créer la réponse, en utilisant l'en-tête de gestion de clés en-tête dans RTSP comme suit :
* L'identifiant du protocole de gestion de clés utilisé (par exemple, MIKEY) DOIT être placé dans le champ "prot" de l'en-tête. Les valeurs de prot sont tenues par l'IANA (Section 9).
* Le champ keymgmt-data DOIT être créé comme suit : le protocole de gestion de clés DOIT être utilisé pour créer le message de gestion de clés. Ce message DEVRA être codé en base64 par l'application RTSP et ensuite encapsulé dans le champ "data" de l'en-tête. La sémantique du message encapsulé dépend du protocole de gestion de clés utilisé.
* Inclure, si nécessaire, l'URL pour indiquer le contexte dans le paramètre "uri".

Le serveur DEVRA traiter un en-tête de gestion de clés reçu dans RTSP comme suit :
* Le protocole de gestion de clés est identifié selon le champ "prot".
* Les données de gestion de clés provenant du champ "data" DOIVENT être extraites, décodées de base64 pour reconstruire le message d'origine, et ensuite passées au protocole de gestion de clés pour traitement.
* Si le protocole de gestion de clés réussit, le traitement peut se poursuivre en accord avec les règles normales.
* Autrement, si la gestion de clés échoue (par exemple, à cause de l'échec de l'authentification ou d'un paramètre non pris en charge) une erreur est renvoyée comme réponse au SETUP en utilisant le code d'erreur RTSP 463 (voir au paragraphe 3.2) et la session est interrompue. Il appartient au protocole de gestion de clés de spécifier (dans le message de code d'état RTSP ou par des messages de gestion de clés) les détails sur le type d'erreur qui s'est produite.

Le changement de clés dans RTSP fera l'objet d'études futures, étant donné que les mécanismes de mise à jour de support dans RTSP ne sont pas encore spécifiés au moment de la rédaction du présent document.

5. Exemples de scénarios

Les exemples suivants utilisent MIKEY [RFC3830] comme protocole de gestion de clés à intégrer dans SDP et RTSP.

5.1 Exemple 1 (SIP/SDP)

Un appel SIP a lieu entre Alice et Bob. Alice envoie un message INVITE consistant en l'offre suivante :

```
v=0
o=alice 2891092738 2891092738 IN IP4 w-land.exemple.com
s=Cool stuff
e=alice@w-land.exemple.com
t=0 0
c=IN IP4 w-land.exemple.com
a=key-mgmt:mikey
AQAFgM0XflABAAAAAAAAAAAAAAsAyONQ6gAAAAAGEEoo2pee4hp2UaDX8ZE22YwKAAAPZG9uYWxkQG
R1Y2suY29tAQAAAAAAAQAk0JKpgaVkDaawi9whVBtBt0KZ14ymNuu62+Nv3ozPLygwK/GbAV9iemnGUIZ19fWQ
UOSrzKTAv9zV
m=audio 49000 RTP/SAVP 98
a=rtpmap:98 AMR/8000
m=video 52230 RTP/SAVP 31
a=rtpmap:31 H261/90000
```

C'est-à-dire que Alice propose d'établir un flux audio et un flux vidéo fonctionnant sur SRTP (signalé par l'utilisation du profil SAVP). Elle utilise MIKEY pour établir les paramètres de sécurité pour SRTP (Section 7). Le message MIKEY contient les paramètres de sécurité, avec le matériel de chiffrement nécessaire. Noter que MIKEY va échanger la suite de chiffrement pour les deux flux, car il est placé au niveau session. Aussi, MIKEY fournit sa propre sécurité, c'est-à-dire que quand Bob traite le message MIKEY d'Alice, il va aussi trouver la signalisation des paramètres de sécurité utilisés pour

sécuriser l'échange MIKEY. Les informations d'authentification du point d'extrémité d'Alice sont aussi portées dans le message MIKEY, pour prouver que le message est authentique. Le message MIKEY ci-dessus est un exemple de message quand la méthode MIKEY pré partagée est utilisée.

À réception de l'offre, Bob vérifie la validité du message MIKEY reçu, et, en cas de vérification réussie, il accepte l'offre et renvoie une réponse à Alice (avec ces informations d'authentification, et, si nécessaire, aussi du matériel de chiffrement provenant de son côté) :

```
v=0
o=bob 2891092897 2891092897 IN IP4 foo.exemple.com
s=Cool stuff
e=bob@foo.exemple.com
t=0 0
c=IN IP4 foo.exemple.com
a=key-mgmt:mikey AQEFgM0XflABAAAAAAAAAAAAAAYAyONQ6gAAAAAJAAAQbWlja2
V5QG1vdXNlLmNvbQABn8HdGE5BMDXFIuGEga+62AgY5cc=
m=audio 49030 RTP/SAVP 98
a=rtpmap:98 AMR/8000
m=video 52230 RTP/SAVP 31
a=rtpmap:31 H261/90000
```

À réception de la réponse, Alice vérifie qu'elle est correcte. En cas de succès, à ce point Alice et Bob partagent les paramètres de sécurité et les clés nécessaire pour une communication RTP sûre.

5.2 Exemple 2 (SDP)

Cet exemple montre ce qu'Alice aurait fait si elle souhaitait protéger seulement le flux audio. Elle aurait placé la ligne MIKEY au niveau support pour le seul flux audio (spécifiant là aussi l'utilisation du profil SRTP, SAVP). La sémantique des messages MIKEY est comme dans le cas précédent, mais ne s'applique qu'au flux audio.

```
v=0
o=alice 2891092738 2891092738 IN IP4 w-land.exemple.com
s=Cool stuff
e=alice@w-land.exemple.com
t=0 0
c=IN IP4 w-land.exemple.com
m=audio 49000 RTP/SAVP 98
a=rtpmap:98 AMR/8000
a=key-mgmt:mikey AQAFgM0XflABAAAAAAAAAAAAAsAy...
m=video 52230 RTP/AVP 31
a=rtpmap:31 H261/90000
```

Bob agirait alors comme décrit dans l'exemple précédent, incluant la réponse MIKEY au niveau support pour le flux audio (comme l'a fait Alice).

Noter que même si l'attribut gestion de clés était spécifié au niveau session, la partie vidéo n'en serait pas affectée (car le profil de sécurité n'est pas utilisé ; à la place le profil RTP/AVP est signalé).

5.3 Exemple 3 (RTSP)

Un client veut établir une session en direct et demande une description de supports au serveur de flux en direct :

```
DESCRIBE rtsp://server.exemple.com/fizzle/foo RTSP/1.0
CSeq: 312
Accept: application/sdp
From: user@exemple.com
```

Le serveur renvoie un message OK incluant une description SDP, avec le message MIKEY. Le message MIKEY contient les paramètres de sécurité nécessaires que le serveur veut offrir au client, avec des informations d'authentification (pour prouver que le message est authentique) et le matériel de chiffrement. Le profil SAVP signale aussi l'utilisation de SRTP pour sécuriser les sessions de supports.

RTSP/1.0 200 OK
CSeq: 312
Date: 23 Jan 1997 15:35:06 GMT
Content-Type: application/sdp
Content-Length: 478

v=0
o=actionmovie 2891092738 2891092738 IN IP4 movie.exemple.com
s=Film d'action
e=action@movie.exemple.com
t=0 0
c=IN IP4 movie.exemple.com
a=control:rtsp://movie.exemple.com/action
a=key-mgmt:mikey AQAFgM0XflABAAAAAAAAAAAAAAsAy...
m=audio 0 RTP/SAVP 98
a=rtpmap:98 AMR/8000
a=control:rtsp://movie.exemple.com/action/audio
m=video 0 RTP/SAVP 31
a=rtpmap:31 H261/90000
a=control:rtsp://movie.exemple.com/action/video

Le client vérifie la validité du message MIKEY reçu, et, en cas de vérification réussie, il accepte le message. Le client inclut alors ses données de gestion de clés dans la demande SETUP qui revient au serveur, et les informations d'authentification du client (pour prouver que le message est authentique) et, si nécessaire, du matériel de chiffrement.

SETUP rtsp://movie.exemple.com/action/audio RTSP/1.0
CSeq: 313
Transport: RTP/SAVP/UDP;unicast;client_port=3056-3057
keymgmt: prot=mikey; uri="rtsp://movie.exemple.com/action";
data="AQEFgM0XflABAAAAAAAAAAAAAYAyONQ6g..."

Le serveur traite la demande incluant la vérification de la validité de l'en-tête de gestion de clés.

RTSP/1.0 200 OK
CSeq: 313
Session: 12345678
Transport: RTP/SAVP/UDP;unicast;client_port=3056-3057;server_port=5000-5001

Noter que dans ce cas, la ligne gestion de clés a été spécifiée au niveau session, et que les informations de gestion de clés ne sont que dans le SETUP relatif au premier flux. Le champ "uri" indique au serveur que le contexte est pour toute la session agrégée à laquelle la gestion de clés s'applique. Le client RTSP poursuit alors l'établissement du second support (vidéo) en agrégation avec l'audio. Comme les deux supports sont en agrégation et que le contexte de clés a été établi dans le premier échange, aucun autre message de gestion de clés n'est nécessaire.

5.4 Exemple 4 (RTSP)

L'utilisation du message MIKEY au niveau support changerait l'exemple précédent comme suit.
Le 200 OK contiendrait les deux attributs SDP distincts pour MIKEY au niveau support l :

RTSP/1.0 200 OK
CSeq: 312
Date: 23 Jan 1997 15:35:06 GMT
Content-Type: application/sdp
Content-Length: 561

v=0
o=actionmovie 2891092738 2891092738 IN IP4 movie.exemple.com
s=Film d'action
e=action@movie.exemple.com
t=0 0
c=IN IP4 movie.exemple.com

```
a=control:rtsp://movie.exemple.com/action
m=audio 0 RTP/SAVP 98
a=rtpmap:98 AMR/8000
a=key-mgmt:mikey AQAFgM0XflABAAAAAAAAAAAA...
a=control:rtsp://movie.exemple.com/action/audio
m=video 0 RTP/SAVP 31
a=rtpmap:31 H261/90000
a=key-mgmt:mikey AQAFgM0AdlABAAAAAAAAAAAA...
a=control:rtsp://movie.exemple.com/action/video
```

Un en-tête de gestion de clé RTSP est inséré dans le SETUP relatif à l'audio et la vidéo séparément :

```
SETUP rtsp://movie.exemple.com/action/audio RTSP/1.0
CSeq: 313
Transport: RTP/SAVP/UDP;unicast;client_port=3056-3057
keymgmt: prot=mikey; uri="rtsp://movie.exemple.com/action/audio";
        data="AQEFgM0XflABAAAAAAAAAAAA..."
```

et de même pour la session vidéo :

```
SETUP rtsp://movie.exemple.com/action/video RTSP/1.0
CSeq: 315
Transport: RTP/SAVP/UDP;unicast;client_port=3058-3059
keymgmt: prot=mikey; uri="rtsp://movie.exemple.com/action/video";
        data="AQEFgM0AdlABAAAAAAAAAAAA..."
```

Note : le paramètre "uri" pourrait être exclu des deux messages SETUP dans cet exemple.

6. Ajout d'autres protocole de gestion de clés

Ce cadre ne peut pas être utilisé avec tous les protocoles de gestion de clés. Le protocole de gestion de clés doit se conformer aux exigences décrites à la Section 4. En plus de cela, ce qui suit doit être défini :

* L'identifiant de protocole de gestion de clés à utiliser comme identifiant de protocole devrait être enregistré par l'IANA conformément à la Section 9.

* Les informations dont la gestion de clés a besoin de la part de SDP et RTSP, et vice versa, comme décrit Section 4. L'API exacte est spécifique de la mise en œuvre, mais elle DOIT au moins prendre en charge l'échange des informations spécifiées.

* Le protocole de gestion de clés à ajouter DOIT être tel que le traitement de la Section 4 (qui décrit ses interactions avec SDP et RTSP) peut être appliqué. Noter en particulier que le paragraphe 4.1.4 exige que chaque protocole de gestion de clés spécifie comment la liste des identifiants de protocoles est authentifiée dans ce protocole de gestion de clés. La gestion de clés DOIT toujours recevoir un identifiant de protocole du ou des protocoles de gestion de clés inclus dans l'offre dans l'ordre correct où ils apparaissent.

Finalement, il est évidemment crucial d'analyser les possibles implications de sécurité induites par l'introduction d'un nouveau protocole de gestion de clés dans le cadre décrit.

Aujourd'hui, le protocole MIKEY [RFC3830] a adopté les extensions de gestion de clés pour fonctionner avec SIP et RTSP (voir la Section 7). D'autres protocoles PEUVENT utiliser l'attribut et l'en-tête décrits, par exemple, Kerberos [RFC4120]; cependant, ceci fera l'objet d'une normalisation ultérieure.

7. Intégration de MIKEY

La [RFC3830] décrit un protocole de gestion de clés pour les applications en temps réel (aussi bien pour la communication d'homologue à homologue que pour la communication de groupe). MIKEY porte les paramètres de sécurité nécessaires pour établir le protocole de sécurité (par exemple, SRTP) qui protège le flux de supports. MIKEY peut être intégré avec SDP et RTSP, suivant les règles et lignes directrices décrites dans le présent document.

MIKEY satisfait aux exigences décrites à la Section 4. Le message MIKEY est formé comme défini dans la [RFC3830], puis passé de MIKEY à l'application SDP qui le code en base64, et l'encapsule dans l'attribut keymgmt-data. Les exemples de la Section 5 utilisent MIKEY, et la sémantique de l'échange est aussi brièvement expliquée.

L'identifiant de protocole de gestion de clés (KMPID) à utiliser comme identifiant de protocole DEVRA être "mikey" et est enregistré par l'IANA ; voir les détails à la Section 9.

Les informations dont la gestion de clés a besoin de la part de SDP et RTSP, et vice versa, suivent la Section 4. Pour éviter les attaques en dégradation, les directives du paragraphe 4.1.4 sont suivies. La liste des identifiants de protocoles est authentifiée dans MIKEY en plaçant la liste dans une charge utile d'extension générale (de type "SDP IDs", [RFC3830]), qui va alors être automatiquement protégée/signée pour son intégrité. Le receveur DEVRA alors confronter la liste dans la charge utile d'extension générale à la liste incluse dans SDP et DEVRAIT (selon la politique) si elles diffèrent, ou si la vérification d'intégrité/signature échoue, rejeter l'offre.

Le serveur va devoir être capable de connaître l'identité du client avant de créer et envoyer un message MIKEY. Pour signaler l'identité (MIKEY) du client au serveur dans le DESCRIBE, il est RECOMMANDÉ d'inclure le champ d'en-tête From dans RTSP. D'autres méthodes pour établir l'identité pourraient être d'utiliser l'adresse IP ou de restituer l'identité de l'authentification RTSP si elle est utilisée.

7.1 Interface MIKEY

Ce paragraphe décrit certains aspects, que les mises en œuvre DEVRAIENT prendre en considération. Si la mise en œuvre de MIKEY est séparée de SDP/SIP/RTSP, une interface de programme d'application (API, *application programming interface*) entre MIKEY et ces protocoles est nécessaire avec certaines fonctionnalités (cependant, ce à quoi cela ressemble exactement dépend de la mise en œuvre).

Le aspects suivants doivent être considérés :

* La possibilité que MIKEY reçoive des informations sur les sessions négociées. Ceci dépend dans une certaine mesure de la mise en œuvre. Mais il est RECOMMANDÉ que, dans le cas de flux SRTP, le nombre de flux SRTP soit inclus (et leur direction). Il est aussi RECOMMANDÉ de fournir les adresses et accès de destination à MIKEY. Quand on se réfère aux flux décrits dans SDP, MIKEY DEVRA allouer deux numéros consécutifs pour les indices relatifs à la session de chiffrement (car chaque flux peut être bidirectionnel). Un exemple : si SDP contient deux lignes m (spécifiant la direction du flux) et si MIKEY est au niveau session, alors MIKEY alloue, par exemple, les identifiants de session de chiffrement (CS ID, *Crypto Session Identifier*) [RFC3830]) '1' et '2' pour la première ligne m, et '3' et '4' pour la seconde ligne m.

* La possibilité que MIKEY reçoive les messages MIKEY entrants et retourne un code d'état à l'application SIP/RTSP.

* La possibilité pour l'application SIP ou RTSP de recevoir des informations de MIKEY. Cela va normalement inclure la réception de l'identifiant de bouquet de session de chiffrement (CSB ID, *Crypto Session Bundle Identifier*) [RFC3830], pour être plus tard capable d'identifier la session MIKEY active) et les SSRC et le compteur de retour à zéro (ROC, *rollover counter*) [RFC3711]) pour l'usage de SRTP. Il est aussi RECOMMANDÉ que des informations supplémentaires sur les erreurs puissent être reçues.

* La possibilité que l'application SIP ou RTSP reçoive les messages MIKEY sortants.

* La possibilité de supprimer un CSB MIKEY (par exemple, si la session SIP est close, le CSB DEVRAIT aussi être clos).

8. Considérations sur la sécurité

Le cadre pour le transfert des données de gestion de clés décrit ici est destiné à fournir les paramètres de sécurité pour la protection de bout en bout de la session de supports. Il est de plus de bonne pratique de sécuriser l'établissement de session (par exemple, SDP, SIP, RTSP, SAP). Cependant, il se peut que la sécurité de l'établissement de session ne soit pas possible de bout en bout, mais seulement bond par bond. Par exemple, SIP exige que des mandataires intermédiaires aient accès à une partie du message SIP, et parfois aussi à la description SDP (cf. [RFC4189]) bien que la confidentialité de bout en bout puisse cacher certains corps aux intermédiaires. Les considérations générales de sécurité pour l'établissement de session se trouvent dans SDP [RFC4566], SIP [RFC3261], et RTSP [RFC2326]. Le cadre défini dans le présent mémoire

est utile quand l'établissement de session n'est pas protégé de bout en bout, mais le flux de supports a besoin d'être protégé de bout en bout ; donc les paramètres de sécurité (comme les clés) ne sont pas révélés ou manipulés par les intermédiaires.

La sécurité va aussi dépendre du niveau de sécurité offert par le protocole de gestion de clés. Il s'ensuit que, dans l'hypothèse où les schémas de gestion de clés sont sûrs, le SDP peut être passé non chiffré sans affecter la gestion de clés en tant que telle, et le flux de supports va quand même être sûr même si certains attaquants obtiennent la connaissance du contenu de SDP. D'autres considérations de sécurité peuvent être trouvées pour chaque protocole de gestion de clés (pour MIKEY dans la [RFC3830]). Cependant, si les messages SDP ne sont pas envoyés protégés en intégrité entre les parties, il est possible à un attaquant actif de changer les attributs sans être détecté. Comme le protocole de gestion de clés peut (indirectement) s'appuyer sur certaines des informations de session provenant de SDP (par exemple, informations d'adresse) une attaque sur SDP peut avoir des conséquences indirectes sur la gestion de clés. Même si le protocole de gestion de clés ne s'appuie pas sur les paramètres de SDP et ne va pas être affecté par leur manipulation, différentes attaques de déni de service (DoS) visant SDP peuvent conduire à une interruption non désirée de l'établissement. Voir aussi les attaques décrites à la fin de cette section.

Le seul attribut protégé en intégrité du flux de supports est, dans le cadre proposé ici, l'ensemble des protocoles de gestion de clés. Par exemple, il est possible de (1) échanger les offres de gestion de clés à travers les messages SDP, ou (2) injecter une offre précédente de gestion de clés dans un nouveau message SDP. En faisant l'hypothèse (nécessaire) que tous les protocoles de gestion de clés impliqués sont sûrs, la seconde attaque va être détectée par les mécanismes de protection contre la répétition du ou des protocoles de gestion de clés. En faisant de plus l'hypothèse que, selon les bonne pratiques courantes normales, la production de chaque offre de gestion de clés est faite avec des choix (pseudo) aléatoires indépendants (pour les clés de session et autres paramètres) la première attaque va être détectée dans la vérification du message de réponse (maintenant incorrect) du répondant (si un tel message est utilisé) ou être une pure attaque de DoS, résultant en ce que l'initiateur et celui qui répond utilisent des clés différentes.

Il est RECOMMANDÉ pour l'identité au niveau de SPD qu'elle soit celle qui est authentifiée au niveau du protocole de gestion de clés. Cependant, cela peut toucher des aspects de confidentialité qui sortent du domaine d'application du présent cadre.

L'utilisation de plusieurs protocoles de gestion de clés dans la même offre peut ouvrir la possibilité d'une attaque en dégradation, comme spécifiée au paragraphe 4.1.4. Pour exclure une telle possibilité, l'authentification de la liste des identifiants de protocole est utilisée. Noter cependant que le niveau de sécurité de l'identifiant de protocole authentifié va être celui (ou moins) du "plus faible" protocole. Donc, l'offre NE DOIT PAS contenir de protocole de sécurité (ou de sa configuration) plus faible que permis par la politique locale de sécurité.

Noter qu'il est impossible de s'assurer de l'authenticité d'une offre déclinée, car même si elle vient du vrai répondant, le fait que la réponse décline l'offre signifie généralement qu'il ne prend pas en charge le ou les protocoles offerts, et par conséquent ne peut être supposé authentifier non plus la réponse. Cela signifie que si l'initiateur n'est pas sûr des protocoles que celui qui répond prend en charge, on RECOMMANDE que l'initiateur offre tous les protocoles acceptables dans une seule offre. Sinon, cela ouvre la possibilité qu'un infiltré affecte le résultat du protocole finalement accepté, en falsifiant des messages d'erreur non authentifiés jusqu'à ce que l'initiateur offre finalement un protocole conforme aux désirs de l'infiltré. Ceci n'est pas réellement un problème de sécurité, mais plutôt une forme douce de déni de service qui peut être évitée en suivant la recommandation ci-dessus. Noter aussi que l'offre déclinée pourrait être le résultat d'un attaquant situé sur le chemin et qui supprime toutes les offres de gestion de clés. La prévention de l'attaque en dégradation, décrite plus haut, ne fonctionnerait pas dans ce cas (car celui qui répond ne reçoit pas d'attribut de gestion de clés). Aussi, il est impossible d'assurer ici l'authenticité d'une offre déclinée, bien que la raison en soit l'attaque de "pelage". Il appartient à la politique locale de décider du comportement à adopter dans le cas où la réponse décline toute sécurité (donc, il est impossible de l'authentifier). Par exemple, si la politique locale exige une communication sûre et ne peut pas en accepter une non sécurisée, l'établissement de la session DEVRA être interrompu.

9. Considérations relatives à l'IANA

9.1 Enregistrement d'attribut SDP

L'IANA a créé un nouveau sous registre pour l'intégration de protocole de gestion de clés à SDP.

Champ d'attribut SDP ("att-field") :

Nom : key-mgmt
Forme longue : field key management protocol attribute *(champ d'attribut de protocole de gestion de clés)*
Type de nom : att-field
Type d'attribut : niveau session et niveau support

Objet : voir la RFC 4567, Section 3.
Référence : RFC 4567, paragraphe 3.1
Valeurs : voir la RFC 4567, paragraphes 3.1 et 9.3.

9.2 Enregistrement RTSP

L'IANA a créé un nouveau sous registre pour l'intégration de protocole de gestion de clés à RTSP.

Suivant les lignes directrices de la [RFC2326], l'enregistrement est défini comme suit :

Nom d'en-tête : keymgmt
Syntaxe d'en-tête : voir la RFC 4567, paragraphe 3.2
Usage prévu : voir la RFC 4567, paragraphe 3.2
Traitement de mandataire : les mandataires NE DEVRONT PAS ajouter, changer, ou supprimer l'en-tête. Le mandataire n'a pas besoin de lire cet en-tête.
Objet : voir la RFC 4567, Section 3

Le code d'état RTSP "463" (RFC 4567) avec la chaîne par défaut "Échec de gestion de clé" doit être enregistré.

9.3 Enregistrement d'identifiant de protocole

Le présent document définit un nouvel espace de noms, "identifiant de protocole de gestion de clés SDP/RTSP", associé à l'identifiant de protocole, KMPID, défini à la Section 3 pour être utilisé avec les attributs enregistrés ci-dessus dans SDP et RTSP.

L'IANA a créé un nouveau sous registre pour le paramètre KMPID, avec l'enregistrement créé initialement : "mikey".

Nom de la valeur : mikey
Nom long : Multimedia Internet KEYing
Objet : Usage de MIKEY avec l'attribut SDP key-mgmt et l'en-tête RTSP keymgmt.
Référence : Section 7 de la RFC 3830

Noter que cet enregistrement implique que l'espace de noms d'identifiant de protocole, KMPID, va être partagé entre SDP et RTSP.

D'autres valeurs peuvent être enregistrées selon la politique "Spécification exigée" définie dans la [RFC2434]. Chaque nouvel enregistrement doit indiquer le nom de paramètre, et l'enregistrer auprès de l'IANA. Noter que le nom de paramètre est sensible à la casse, et il est RECOMMANDÉ que le nom soit en minuscules. Pour chaque nouvel enregistrement, il est obligatoire qu'un document permanent, stable, et publiquement accessible existe pour spécifier la sémantique du paramètre enregistré et les détails demandés d'interaction entre le protocole de gestion de clés et SDP, comme spécifié dans la RFC 4567.

Les nouvelles valeurs DOIVENT être enregistrées par l'IANA. Les enregistrements DEVRONT inclure les informations suivantes :

* Contact : le nom et l'adresse de messagerie électronique du contact.
* Nom de la valeur : nom de la valeur enregistrée (DOIT se conformer au KMPID défini à la Section 3).
* Nom long : nom de forme longue en anglais.
* Objet : brève explication de l'objet du nom enregistré.
* Référence : référence à la spécification (par exemple, numéro de RFC) fournissant les lignes directrices d'usage conformément à la Section 6 (et aussi conforme aux exigences spécifiées).

10. Remerciements

Les auteurs tiennent à remercier Francois Audet, Rolf Blom, Johan Bilien, Magnus Brolin, Erik Eliasson, Martin Euchner, Steffen Fries, Joerg Ott, Jon Peterson, et Jon-Olov Vatn. Des remerciements particuliers à Colin Perkins et Magnus Westerlund, qui ont contribué dans ne nombreuses sections.

11. Références

11.1 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)

[RFC2326] H. Schulzrinne, A. Rao et R. Lanphier, "Protocole de flux directs en temps réel (RTSP)", avril 1998. *(Remplacée par* RFC7826)

[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)

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

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

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

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

[RFC3986] T. Berners-Lee, R. Fielding et L. Masinter, "Identifiant de ressource uniforme (URI) : Syntaxe générique", STD 66, janvier 2005. *(P.S. ; MàJ par* RFC8820)

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

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

11.2 Références pour information

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

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

[RFC4120] C. Neuman et autres, "Service Kerberos d'authentification de réseau (V5)", juillet 2005. (*MàJ par* RFC4537, 5021, 6649, 7751, 8062, 8129, 8429)

[RFC4189] K. Ono, S. Tachimoto, "Exigences de sécurité jusqu'à mi-parcours pour le protocole d'initialisation de session (SIP)", octobre 2005. *(Information)*

[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.)*

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

Adresse des auteurs

Jari Arkko
Ericsson
02420 Jorvas
Finland
téléphone : +358 40 5079256
mél : jari.arkko@ericsson.com

Elisabetta Carrara
Royal Institute of Technology
Stockholm
Sweden
mél : carrara@kth.se

Fredrik Lindholm
Ericsson
SE-16480 Stockholm
Sweden
téléphone : +46 8 58531705
mél : fredrik.lindholm@ericsson.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

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