Groupe de travail Réseau
Request for Comments : 4650
Catégorie : Sur la voie de la normalisation

M. Euchner
septembre 2006
Traduction Claude Brière de L'Isle

Diffie-Hellman authentifié par HMAC pour chiffrement multimédia Internet (MIKEY)

Statut du présent mémoire

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

Résumé

Le présent document décrit une variante de protocole léger de gestion de clés point à point pour le protocole de chiffrement multimédia Internet (MIKEY) comme défini dans la RFC 3830. En particulier, cette variante déploie le protocole classique d'accord de clé Diffie-Hellman pour l'établissement de clé fournissant le secret parfait vers l'avant en conjonction avec un code d'authentification de message haché chiffré pour réaliser l'authentification mutuelle et l'intégrité de message pour les messages de gestion de clé échangés. Ce protocole traite les contraintes de sécurité et de performances de la gestion de clés multimédia dans MIKEY.

Table des matières

1. Introduction

L'IETF travaille à développer les schémas de gestion de clés. Par exemple, IKE [RFC2409] est un schéma largement accepté d'envoi individuel pour IPsec, et le groupe de travail MSEC développe d'autres schémas, visant la communication de groupe [RFC3547], [RFC4535]. Pour des raisons qu'on discutera plus loin, il y a cependant un besoin pour un schéma à faible latence, convenant à des cas exigeants comme les données en temps réel sur des réseaux hétérogènes et de petits groupes interactifs.

Comme mentionné dans MIKEY [RFC3830], les applications multimédia sûres en temps réel demandent un schéma particulier de gestion de clé léger adéquat pour veiller à établir efficacement en toute sécurité des clés de session dynamiques dans un scénario de conversation multimédia.

En général, les scénarios de MIKEY couvrent de l'homologue à homologue, de un à plusieurs simple, et des groupes de petite taille. MIKEY en particulier décrit trois schémas de gestion de clés pour le cas d'homologue à homologue qui finissent tous leur tâche en un seul aller-retour :
- un protocole de distribution de clé symétrique (MIKEY-PS) fondé sur des clés maîtresses pré-partagées,
- un protocole de distribution de clé fondé sur le chiffrement de clé publique (MIKEY-PK et MIKEY-RSA-R en mode inverse [RFC4738]) supposant une infrastructure de clé publique avec des clés publiques/privées fondées sur RSA (Rivest, Shamir et Adleman) et des certificats numériques,
- un protocole d'accord de clé Diffie-Hellman (MIKEY-DHSIGN) déployant des signatures et certificats numériques.

Ces trois protocoles de gestion de clés sont tous conçus de façon à ce que ils achèvent leur travail en un seul aller-retour. Cela exige de dépendre d'horloges en gros synchronisées et de déployer des horodatages au sein des protocoles de gestion de clés.

Cependant, il est connu [Handbook] que chacun des trois schémas de gestion de clé a de subtiles contraintes et limitations :

- Le protocole de distribution de clés symétriques (MIKEY-PS) est simple à mettre en œuvre ; cependant, il n'a pas été destiné à s'adapter à prendre en charge d'autres configurations que d'homologue à homologue, le un à plusieurs simple, et les petits groupes interactifs, à cause du besoin de secrets maîtres pré-alloués partagés mutuellement.
  De plus, la sécurité fournie ne réalise pas la propriété de secret parfait vers l'avant ; c'est-à-dire, la compromission du secret maître partagé va rendre les clés de session passées et même futures susceptibles de compromission.
  De plus, la génération de la clé de session se fait juste chez l'initiateur. Donc, le répondant doit faire pleine confiance à l'initiateur pour choisir un bon et sûr secret de session ; le répondant n'est capable ni de participer à la génération de clé ni d'influencer ce processus. Ceci est considéré comme une limitation spécifique dans des environnements de moindre confiance.

- Le schéma de chiffrement à clé publique (MIKEY-PK et MIKEY-RSA-R [RFC4738]) dépend d'une infrastructure de clé publique qui certifie les clés privées-publiques en produisant et maintenant des certificats numériques. Bien que de tels schémas de gestion de clés fournissent une complète adaptabilité dans les configurations de grands réseaux, les infrastructures de clés publiques ne sont toujours pas largement disponibles, et, en général, les mises en œuvre sont significativement complexes.

  De plus, des allers-retours et des calculs supplémentaires peuvent être nécessaires pour que chaque système d'extrémité assure la vérification des certificats numériques. Par exemple, le fonctionnement normal dans le contexte d'une infrastructure de clé publique peut impliquer des prises de contact de communication de réseau supplémentaires avec l'infrastructure de clé publique et avec les autorités de certification et peut normalement impliquer des étapes de traitement supplémentaire dans les systèmes d'extrémité. Ces opérations vont inclure de valider les certificats numériques [RFC3029], de s'assurer de l'état de révocation des certificats numériques [RFC2560], de s'assurer des politiques de certificat, de la construction de chemins de certification [RFC4158], de demander et obtenir les certificats nécessaires [RFC2511], et de gérer les certificats pour ces besoins [RFC4210]. Ces étapes et tâches résultent toutes en délais de l'accord de clés ou de la phase d'établissement de clés parmi les systèmes d'extrémité, qui affectent négativement le temps d'établissement. Toutes les prises de contact et les traitements de PKI ne sont pas dans le domaine de MIKEY, et comme le présent document s'étend seulement sur les mécanismes de sécurité symétriques, les aspects de PKI, de certificats numériques, et les traitements qui s'y rapportent ne sont pas discutés dans le présent document.

  Finalement, comme dans le cas symétrique, le rép pré-allouésondant dépend complètement du choix par l'initiateur de clés de session bonnes et sûres.

- Le troisième protocole de gestion de clés MIKEY-DHSIGN déploie le schéma d'accord de clés Diffie-Hellman et authentifie l'échange des demies clés Diffie-Hellman dans chaque direction en utilisant une signature numérique. Cette approche a les mêmes avantages et inconvénients que décrits au paragraphe précédent en termes d'infrastructure de clé publique.

  Cependant, le protocole d'accord de clé Diffie-Hellman est connu pour ses subtiles forces de sécurité en ce qu'il est capable de fournir un secret parfait vers l'avant (PFS, *perfect forward secrecy*) complet et a de plus les deux parties activement impliquées dans la génération des clés de session. Ces propriétés de sécurité particulières (en dépit de coûts de calcul un peu plus élevés) rendent les techniques Diffie-Hellman attractives en pratique.

Afin de surmonter certaines des limitations mentionnées ci-dessus, il a été reconnu un besoin particulier d'une autre variante efficace de protocole d'accord de clé dans MIKEY. Cette variante de protocole vise à fournir la capacité de secret parfait vers l'avant au titre de l'accord de clés avec une faible latence sans dépendance à l'infrastructure de clé publique.

Le présent document décrit un quatrième schéma de gestion de clés léger pour MIKEY qui pourrait être vu un peu comme une optimisation synergique entre le schéma de distribution de clés pré-partagées et l'accord de clé Diffie-Hellman.

L'idée du protocole de ce document est d'appliquer l'accord de clés Diffie-Hellman, mais plutôt que de déployer une signature numérique pour l'authenticité du matériel de chiffrement échangé, il utilise un hachage chiffré pour les secrets partagés pré alloués symétriquement. Cette combinaison des mécanismes de sécurité est appelée l'accord de clé Diffie-Hellman (DH) authentifié par HMAC pour MIKEY (DHHMAC).

La variante DHHMAC suit étroitement la conception et la philosophie de MIKEY et réutilise les composants de charge utile du protocole MIKEY et les mécanismes de MIKEY pour son bénéfice maximum et la meilleure compatibilité.

Comme le protocole MIKEY Diffie-Hellman, DHHMAC ne s'adapte pas au delà d'une constellation point à point ; donc, les deux protocoles MIKEY Diffie-Hellman ne prennent pas en charge le chiffrement fondé sur le groupe pour tout groupe de plus de deux entités.

1.1 Définitions

Les définitions et notations de ce document sont alignées sur MIKEY ; voir la [RFC3830] paragraphes 1.3 - 1.4.

Tous les calculs de grands entiers dans le présent document devraient être compris comme étant modulo p au sein d'un groupe G fixé pour un grand nombre premier p ; voir la [RFC3830] paragraphe 3.3. Cependant, le protocole DHHMAC est aussi applicable généralement aux autres groupes cycliques finis appropriés.

Il est supposé qu'une clé pré-partagée s est connue des deux entités (initiateur et répondant). La clé d'authentification auth_key est déduite du secret pré-partagé s en utilisant la fonction pseudo aléatoire PRF ; voir les paragraphes 4.1.3 et 4.1.5 de la [RFC3830].

Dans ce texte, [X] représente un élément d'information facultatif. Généralement dans ce texte, X DEVRAIT être présent sauf si certaines circonstances PEUVENT permettre que X soit facultatif et ne soit pas présent, résultant potentiellement en une sécurité plus faible. De même, [X, Y] représente un élément d'information composé où les éléments X et Y DEVRAIENT soit être tous deux présents, soit PEUVENT être FACULTATIVEMENT tous deux absents. {X} note zéro, une ou plusieurs occurrences de X.

1.2 Abréviations

auth_key : clé d'authentification pré-partagée, déduite par PRF de la clé pré-partagée s.
DH : Diffie-Hellman
DHi : demie clé publique Diffie-Hellman g^(xi) de l'initiateur
DHr : demie clé publique Diffie-Hellman g^(xr) du répondant
DHHMAC : Diffie-Hellman authentifié par HMAC
DoS : déni de service
G : groupe Diffie-Hellman
HDR : charge utile d'en-tête commun MIKEY
HMAC : code d'authentification de message haché chiffré
HMAC-SHA1 : HMAC utilisant SHA1 comme fonction de hachage (résultat de160 bits)

IDi : identité de l'initiateur
IDr : identité du receveur
IKE : échange de clé Internet
IPsec : sécurité du protocole Internet
MIKEY *(Multimedia Internet KEYing)* chiffrement Internet multimédia
MIKEY-DHHMAC : protocole de gestion de clés Diffie-Hellman MIKEY avec HMAC
MIKEY-DHSIGN : protocole d'accord de clé Diffie-Hellman MIKEY
MIKEY-PK : protocole MIKEY de distribution de clé fondé sur le chiffrement à clé publique
MIKEY-PS : protocole MIKEY de distribution de clé pré-partagée
p : module de nombre premier Diffie-Hellman
PKI : Infrastructure de clé publique
PRF : fonction pseudo aléatoire MIKEY (voir la [RFC3830] paragraphe 4.1.3)
RSA : Rivest, Shamir, et Adleman
s : clé pré-partagée
SDP : protocole de description de session
SOI (*Son-of-IKE*) : IKEv2
SP *(Security Policy)* : charge utile de politique de sécurité MIKEY (paramètre)
T *(Timestamp)* : horodatage
TEK *(Traffic Encryption Key)* : clé de chiffrement du trafic
TGK *(TEK Generation Key)* : clé de génération de clé de chiffrement du trafic MIKEY, comme secret partagé commun Diffie-Hellman
TLS *(Transport Layer Security)* : sécurité de la couche transport
xi : clé secrète, (pseudo) aléatoire Diffie-Hellman de l'initiateur
xr : clé secrète, (pseudo) aléatoire Diffie-Hellman du répondant

1.3 Conventions utilisées dans le document

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. Scénario

Le protocole d'accord de clé Diffie-Hellman authentifié par HMAC (DHHMAC) pour MIKEY vise les mêmes scénarios et portées que les trois autres schémas de gestion de clé de MIKEY.

DHHMAC est applicable dans un groupe d'homologue à homologue où aucun accès à une infrastructure de clé publique ne peut être supposé disponible. Des secrets maîtres pré partagés sont plutôt supposés être disponibles parmi les entités dans cet environnement.

Dans un groupe de paires, il est supposé que chaque client va établir une clé de session pour ses liaisons sortantes avec son homologue en utilisant le protocole d'accord de clé DH-MAC.

Comme c'est le cas pour les trois autres protocoles de gestion de clés MIKEY, DHHMAC suppose, au moins, des horloges à synchronisation lâche parmi les entités du petit groupe.

Pour synchroniser les horloges de manière sûre, des moyens opérationnels ou de procédure sont recommandés. MIKEY-DHHMAC ne définit aucune mesure de synchronisation sûre ; cependant, les paragraphes 5.4 et 9.3 de la [RFC3830] fournissent des directives de mise en œuvre sur la synchronisation d'horloge et les horodatages.

2.1 Applicabilité

MIKEY-DHHMAC et les autres protocoles de gestion de clés MIKEY sont destinés à la gestion de clé de niveau application et sont optimisés pour les applications multimédia avec des contraintes d'établissement de session et de gestion de session en temps réel.

Comme le protocole de gestion de clés MIKEY-DHHMAC se termine sur un aller-retour, DHHMAC est applicable pour l'intégration dans une session à prise de contact en deux temps ou des protocoles de signalisation d’appel comme :

a) SIP [RFC3261] et SDP, où les messages MIKEY codés sont encapsulés et transportés dans des conteneurs SDP de la prise de contact par offre/réponse SDP (voir la [RFC3264]), comme décrit dans la [RFC4567]; et

b) H.323 (voir [H.235.7]), où les messages codés MIKEY sont transportés dans la prise de contact à signalement d'appel à démarrage rapide de H.225.0. L'Appendice A précise l'usage de MIKEY-DHHMAC dans H.235.

MIKEY-DHHMAC est offert comme option aux autres variantes de gestion de clé MIKEY (MIKEY pré-partagé, MIKEY à clé publique et MIKEY-DH-SIGN) pour tous les cas où DHHMAC a ses forces particulières (voir la Section 5).

2.2 Relation avec GKMARCH

L'architecture de gestion de clé de groupe (GKMARCH) [RFC4046] décrit une architecture générique pour les protocoles de gestion de clés de groupe de diffusion groupée. Dans le contexte de cette architecture, MIKEY-DHHMAC peut opérer comme un protocole d'enregistrement ; voir aussi au paragraphe 2.4 de la [RFC3830]. Les principales entités impliquées dans l'architecture sont un contrôleur de groupe/serveur de clé (GCKS, *group controller/key server*), le ou les receveurs, et le ou les envoyeurs. Du fait de la nature par paires du fonctionnement de Diffie-Hellman et de la contrainte d'un seul aller-retour, l'usage de MIKEY-DHHMAC exclut tout déploiement comme protocole de gestion de clés de groupe avec plus de deux entités de groupe. Seul le cas dérivé avec deux homologues est possible lorsque, par exemple, le répondant agit comme contrôleur de groupe.

Noter que MIKEY n'assure pas le changement de clés au sens de GKMARCH, mais seulement la mise à jour des clés par des messages normaux en envoi individuel.

3. Protocole de sécurité DHHMAC

La figure suivante définit le protocole de sécurité pour DHHMAC :

Figure 1 : échange fondé sur une clé Diffie-Hellman authentifiée par HMAC, où xi et xr sont choisis de façon (pseudo) aléatoire, respectivement, par l'initiateur et le répondant

L'échange de clé DHHMAC DEVRA être fait en accord avec la Figure 1. L'initiateur choisit une valeur (pseudo) aléatoire, xi, et envoie un message HMAC incluant g^(xi) et un horodatage au répondant. Il est recommandé que l'initiateur DEVRAIT toujours inclure les charges utiles d'identité IDi et IDr dans le I_message ; sauf si le receveur peut déduire l'identité de l'initiateur par d'autres moyens, IDi PEUT facultativement être omis. L'initiateur DEVRA toujours inclure l'identité du receveur.

Les paramètres de groupe (par exemple, le groupe G) sont un ensemble de paramètres choisis par l'initiateur. Noter que comme dans le protocole MIKEY, l'envoyeur et le receveur transmettent tous deux explicitement le groupe G Diffie-Hellman au sein de la charge utile Diffie-Hellman DHi ou DHr par un codage (par exemple, un numéro de groupe OAKLEY ; voir la [RFC3830] paragraphe 6.4). Les paramètres de groupe g et p réels ne sont cependant pas explicitement transmis mais peuvent être déduits du groupe Diffie-Hellman G. Le répondant choisit un entier positif (pseudo) aléatoire, xr, et envoie un message HMAC incluant g^(xr) et l'horodatage à l'initiateur. Le répondant DEVRA toujours inclure l'identité de l'initiateur IDi sans considération de si le I_message portait un IDi. Il est RECOMMANDÉ que le répondant DEVRAIT toujours inclure la charge utile d'identité IDr dans le R_message ; si l'initiateur peut déduire l'identité du répondant par d'autres moyens, IDr PEUT facultativement être omis.

Les deux parties calculent alors la TGK comme g^(xi * xr).

L'authentification HMAC assure l'authentification des demies clés DH et est nécessaire pour éviter des attaques par interposition.

Cette approche est moins coûteuse que la signature numérique Diffie-Hellman en ce que les deux côtés calculent une

exponentiation et un HMAC d'abord, puis une vérification HMAC, et finalement une autre exponentiation Diffie-Hellman.

Avec le pré calcul hors ligne, la demie clé initiale Diffie-Hellman PEUT être calculée avant la transaction de gestion de clé et par là PEUT encore réduire le délai global d'aller-retour, ainsi que le risque d'attaque de déni de service.

Le traitement de la TGK DEVRA être accompli comme décrit dans MIKEY [RFC3830] Section 4.

Le résultat calculé du HMAC DEVRA être porté dans le champ de charge utile KEMAC où les champs MAC contiennent le résultat du HMAC. Le HMAC DEVRA être calculé sur le message entier, excluant le champ MAC en utilisant auth_key ; voir aussi le paragraphe 4.2.

3.1 Changement de clé TGK

Le changement de TGK pour DHHMAC se fait généralement comme décrit au paragraphe 4.5 de la [RFC3830]. Précisément, la Figure 2 montre l'échange de messages pour le message de mise à jour DHHMAC.

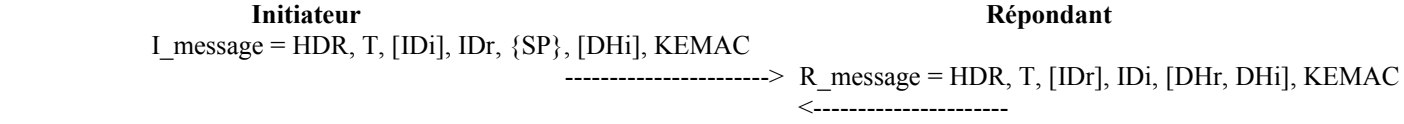

Figure 2 : Message de mise à jour DHHMAC

Le changement de TGK supporte deux procédures :

a) Un vrai changement de clé en échangeant des demies clés Diffie-Hellman nouvelles et fraîches. Pour cela, l'initiateur DEVRA fournir une nouvelle Dhi fraîche, et le répondant DEVRA répondre avec une nouvelle DHr fraîche et la Dhi reçue.

b) Une mise à jour d'informations sans relation avec la clé sans inclure de demies clés Diffie-Hellman dans l'échange. Une telle transaction ne change pas la TGK actuelle mais met à jour les autres informations comme les paramètres de politique de sécurité. Pour mettre à jour seulement les informations non relatives à la clé, [DHi] et [DHr, DHi] DEVRONT être omises.

4. Formats de charge utile DHHMAC

Cette section spécifie les formats de charge utile et les valeurs de type de données pour DHHMAC ; voir aussi la [RFC3830] section 6, pour une définition des charges utiles MIKEY.

Le présent document ne définit pas de nouveau format de charge utile mais réutilise les charges utiles de MIKEY pour DHHMAC comme suit :

* Charge utile d'en-tête commun (HDR) ; voir le paragraphe 4.1 et la [RFC3830] paragraphe 6.1.
* Sous charge utile Identifiant SRTP ; voir la [RFC3830] paragraphe 6.1.1.
* Charge utile Transport de données de clé (KEMAC) ; voir le paragraphe 4.2 et la [RFC3830] paragraphe 6.2.
* Charge utile Données DH ; voir la [RFC3830] paragraphe 6.4.
* Charge utile Horodatage ; voir la [RFC3830] paragraphe 6.6.
* Charge utile Identifiant ; [RFC3830] paragraphe 6.7.
* Charge utile Politique de sécurité (SP) ; voir la [RFC3830] paragraphe 6.10.
* Charge utile RAND (RAND) ; voir la [RFC3830] paragraphe 6.11.
* Charge utile Erreur (ERR) ; voir la [RFC3830] paragraphe 6.12.
* Charge utile Extension générale ; voir la [RFC3830] paragraphe 6.15.

4.1 Charge utile d'en-tête commun

En se référant à la [RFC3830] paragraphe 6.1, les types de données suivants DEVRONT être utilisés pour DHHMAC :

Type de données	Valeur	Commentaire
DHHMAC init	7	message d'échange DHHMAC de l'initiateur
DHHMAC resp	8	message d'échange DHHMAC du répondant
Erreur	6	message d'erreur ; voir la [RFC3830] paragraphe 6.12

Tableau 4.1.a

Note : un répondant est capable de reconnaître le protocole MIKEY DHHMAC en évaluant le champ Type de données comme 7 ou 8. C'est la façon dont le répondant peut différencier MIKEY et MIKEY DHHMAC.

Le champ Prochaine charge utile DEVRA être d'une des valeurs suivantes :

Prochaine charge utile	Valeur	Paragraphe
Dernière charge utile	0	-
KEMAC	1	paragraphe 4.2 et [RFC3830] paragraphe 6.2
DH	3	[RFC3830] paragraphe 6.4
T	5	[RFC3830] paragraphe 6.6
ID	6	[RFC3830] paragraphe 6.7
SP	10	[RFC3830] paragraphe 6.10
RAND	11	[RFC3830] paragraphe 6.11
ERR	12	[RFC3830] paragraphe 6.12
General Ext.	21	[RFC3830] paragraphe 6.15

Tableau 4.1.b

Les autres valeurs de prochaine charge utile définies dans la [RFC3830] NE DEVRONT PAS être appliquées à DHHMAC.

En cas d'erreur de décodage ou d'échec de vérification d'authentification HMAC, le répondant DEVRA appliquer le type de données de charge utile Erreur (12).

4.2 Charge utile de transport de données de clé (KEMAC)

DHHMAC DEVRA appliquer cette charge utile pour porter le résultat HMAC avec algorithme d'authentification indiqué. Quand il est utilisé en conjonction avec DHHMAC, KEMAC NE DEVRA PAS porter de données chiffrées ; donc, Encr alg DEVRA être réglé à 2 (NULL), Encr data len DEVRA être réglé à 0, et Encr data DEVRA être laissé vide. La méthode AES key wrap (voir la [RFC3394]) NE DEVRA PAS être appliquée pour DHHMAC.

Pour DHHMAC, cette charge utile de transport de données de clés DEVRA être la dernière charge utile dans le message. Noter que le champ Prochaine charge utile DEVRA être réglé à Dernière charge utile (0). Le HMAC est ensuite calculé sur le message MIKEY entier, à l'exclusion du champ MAC, en utilisant auth_key comme décrit au paragraphe 5.2 de la [RFC3830], et ensuite mémorisé dans le champ MAC.

Algorithme MAC	Valeur	Commentaire
HMAC-SHA-1	0	Obligatoire, par défaut (voir [FIBS180-2])
NULL	1	Usage très restreint ; voir au paragraphe 4.2.4 de la [RFC3830]

Tableau 4.2.a

HMAC-SHA-1 est la fonction de hachage par défaut qui DOIT être mise en œuvre au titre de DHHMAC. La longueur du résultat HMAC-SHA-1 est de 160 bits.

4.3 Charge utile ID (ID)

Pour DHHMAC, cette charge utile DEVRA seulement contenir une identité non fondée sur un certificat.

4.4 Charge utile d'extension générale

Pour DHHMAC, pour éviter des attaques en dégradation, cette charge utile DEVRA faire la liste de tous les identifiants de protocole de gestion de clés d'un protocole d'encapsulation englobant, comme SDP [RFC4567]. La charge utile Extension

générale DEVRA être protégée en intégrité avec le HMAC en utilisant le secret partagé.

Type	Valeur	Commentaire
ID SDP	1	Liste des identifiants de gestion de clé SDP (alloués dans la [RFC4567]) ; voir aussi au paragraphe 6.15 de la [RFC3830].

Tableau 4.4.a

5. Considérations sur la sécurité

Le présent document traite entièrement des questions de sécurité de la gestion de clé. Pour une explication complète des considérations de sécurité de MIKEY, prière de se reporter à la section 9 de MIKEY [RFC3830].

De plus, ce document traite des questions de sécurité de la [RFC3552], où les considérations de sécurité suivantes s'appliquent en particulier au présent document:

5.1 Environnement de sécurité

Le protocole de sécurité DHHMAC décrit dans ce document se concentre principalement sur la sécurité de communication ; c'est-à-dire, les questions de sécurité qui concernent le protocole MIKEY DHHMAC. Néanmoins, certaines questions de sécurité de système qui sont aussi intéressantes ne sont pas explicitement définies par le protocole DHHMAC, mais elles devraient être en pratique traitées localement.

Le système qui fait fonctionner l'entité de protocole DHHMAC DEVRA avoir la capacité de générer des nombres (pseudo) aléatoires en entrée de l'opération Diffie-Hellman (voir la [RFC1750]). De plus, le système DEVRA être capable de mémoriser les données (pseudo) aléatoires générées, les données secrètes, les clés, et autres paramètres de sécurité secrets de façon sûre (c'est-à-dire, confidentielle et à l'abri d'une altération non autorisée).

5.2 Modèle de menace

Le modèle de menace, auquel le présent document adhère, couvre les questions de sécurité de bout en bout dans l'Internet général, sans exclure la possibilité que MIKEY DHHMAC puisse être déployé dans un environnement IP clos d'entreprise. Cela inclut aussi la possibilité que MIKEY DHHMAC puisse être déployé bond par bond avec l'implication de mandataires intermédiaires "MIKEY DHHMAC" de confiance.

Comme DHHMAC est un protocole de gestion de clés, les menaces sur la sécurité suivantes sont à prévoir :

* Interception non autorisée des TGK : pour DHHMAC, cette menace n'existe pas car la TGK n'est pas réellement transmise sur le réseau (même pas de façon chiffrée).

* Espionnage des autres informations de chiffrement transmises : le protocole DHHMAC ne transmet pas explicitement du tout la TGK. Il utilise à la place l'opération de "chiffrement" Diffie-Hellman, qui dissimule les valeurs secrètes (pseudo) aléatoires, et seules sont transmises des informations partielles (c'est-à-dire, la demie clé DH) pour la construction de la TGK. Il est fondamentalement supposé que la disponibilité de ces demies clés Diffie-Hellman pour un espion ne résulte en aucun risque de sécurité substantiel ; voir au paragraphe 5.4. De plus, DHHMAC porte d'autres données telles que des horodatages, des valeurs (pseudo) aléatoires, des informations d'identification ou des paramètres de politique de sécurité ; l'espionnage de ces données n'est pas considéré comme donnant un risque de sécurité significatif.

* Déguisement en l'une ou l'autre entité : cette menace pour la sécurité doit être évitée, et si une attaque par déguisement devait être tentée, des moyens de détection appropriés doivent être en place. DHHMAC traite cette menace en fournissant l'authentification mutuelle des entités homologues.

* Attaques par interposition : de telles attaques menacent la sécurité des messages non authentifiés échangés. Les attaques par interposition viennent généralement avec un déguisement et/ou la perte de l'intégrité de message (voir ci-dessous). Les attaques par interposition doivent être évitées et, si il s'en présente ou si il en est tenté, doivent être détectées de façon appropriée. DHHMAC traite cette menace en fournissant l'authentification mutuelle des entités homologues et la protection de l'intégrité du message.

* Perte d'intégrité : cette menace pour la sécurité se rapporte à la répétition, suppression, insertion, et manipulation non autorisée des messages. Bien que de telles attaques ne puissent pas être évitées, elles doivent au moins être détectées. DHHMAC traite cette menace en assurant l'intégrité du message.

* Attaques en dégradation : quand plusieurs protocoles de gestion de clés, ayant chacun un niveau de sécurité distinct, sont offerts (comme ceux rendus possibles par SDP [RFC4567]), éviter les attaques en dégradation est un souci. DHHMAC traite cette menace en réutilisant le mécanisme de charge utile Extension générale de MIKEY, où la liste de tous les identifiants de protocole de gestion de clés figure dans la charge utile Extension générale de MIKEY.

Certaines menaces potentielles ne sont pas couvertes par ce modèle de menaces :

* Cryptanalyse passive et hors ligne de l'algorithme Diffie-Hellman : dans certaines hypothèses raisonnables (voir au paragraphe 5.4, ci-dessous) il est largement estimé que DHHMAC est suffisamment sûr et que de telles attaques sont infaisables, bien que la possibilité d'une attaque réussie ne puisse pas être exclue.

* Non répudiation de la réception ou de l'origine du message : ce ne sont pas des exigences dans le contexte de DHHMAC dans cet environnement, et donc des contre mesures à ce sujet ne sont pas fournies du tout.

* Attaques de déni de service ou de déni de service réparties : une certaine considération doit être accordée à ces attaques, mais DHHMAC ne prétend pas fournir de contre mesure contre ces attaques. Par exemple, affaiblir la disponibilité des entités n'est pas contrecarré au moyen du protocole de gestion de clés ; d'autres contre mesures locales devraient être appliquées. De plus, certaines attaques de DoS ne sont pas contrées, comme l'interception d'une demande DH valide et sa duplication massive instantanée. De telles attaques pourraient au moins être contrées partiellement par des moyens locaux qui sortent du domaine d'application du présent document.

* Protection d'identité : comme dans MIKEY, la protection de l'identité n'est pas une exigence majeure de la conception de MIKEY-DHHMAC, voir la [RFC3830]. Aucun protocole de sécurité connu n'est capable de fournir les objectifs de DHHMAC comme déclarés au paragraphe 5.3, incluant la protection de l'identité au sein d'un seul aller-retour. MIKEY-DHHMAC échange la protection de l'identité contre une meilleure sécurité du matériel de chiffrement et un temps plus court d'aller retour. Donc, MIKEY-DHHMAC ne fournit pas de protection d'identité par lui-même mais peut hériter d'une telle propriété d'un protocole de sécurité sous lequel cette protection d'identité est réellement fournie.

Le protocole de sécurité DHHMAC (voir la Section 3) et le protocole de sécurité du changement de clé de TGK (voir le paragraphe 3.1) fournissent l'option de ne pas donner d'informations d'identité. Cette option est seulement applicable si d'autres moyens sont disponibles pour fournir des informations d'identité dignes de confiance ; par exemple, en s'appuyant sur des liaisons sécurisées en dessous de MIKEY qui fournissent des informations d'identité dignes de confiance par d'autre moyens. Cependant, on comprend que sans informations d'identité, les protocoles de gestion de clé MIKEY seraient l'objet de faiblesses de sécurité comme les attaques par déguisement, usurpation d'identité, et réflexion, en particulier dans des scénarios de bout en bout où aucun autre moyen sûr d'assurer les informations d'identité n'est fourni.

Laisser facultatifs les champs d'identité (si il est possible de le faire) ne devrait donc pas être vu comme une méthode confidentielle, mais plutôt comme une caractéristique d'optimisation de protocole.

5.3 Caractéristiques et propriétés de sécurité

En pensant aux menaces sur la sécurité, le présent document fournit les caractéristiques de sécurité et propriétés suivantes :

* Un accord de clé sûr avec l'établissement d'une TGK chez les deux homologues : ceci est fait en utilisant un protocole de gestion de clés Diffie-Hellman authentifié.

* Authentification (mutuelle) des entités homologues : cette authentification corrobore que l'hôte/usager est authentique en ce que la possession d'une clé secrète pré-allouée est prouvée en utilisant le HMAC chiffré. L'authentification se fait sur le message de demande et sur le message de réponse ; donc l'authentification est mutuelle.
  Le calcul de HMAC corrobore l'authentification et l'intégrité du message des demies clés Diffie-Hellman échangées et des messages associés. L'authentification est absolument nécessaire afin d'éviter des attaques par interposition sur les messages échangés en transit et, en particulier, sur les demies clés Diffie-Hellman échangées qui ne sont pas autrement authentifiées.

Note : ce document ne traite pas les problèmes qui concernent l'autorisation ; ce trait n'est pas explicitement fourni.

Cependant, l'authentification DHHMAC signifie la prise en charge et facilite la réalisation des moyens d'autorisation (question locale).

* Vérification d'intégrité cryptographique : elle est réalisée en utilisant un résumé de message (HMAC chiffré). Elle inclut les demies clés Diffie-Hellman échangées mais couvre aussi les autres parties du message échangé. L'authentification mutuelle des entités homologues et l'intégrité de message fournissent toutes deux des contre mesures efficaces contre les attaques par interposition.
  L'initiateur peut déployer un temporisateur local qui expire si le message de réponse attendu n'arrive pas à temps. C'est destiné à détecter la suppression de messages entiers.

* La protection contre la répétition des messages est réalisée en utilisant des horodatages incorporés : afin de détecter les messages répétés, il est essentiel que les horloges sont en gros synchronisées entre l'initiateur et l'envoyeur. Le lecteur se reportera au paragraphe 5.4 de la [RFC3830] et au paragraphe 9.3 de la [RFC3830] qui discutent plus à fond et donnent des directives sur la synchronisation d'horloge et l'usage de l'horodatage. Si la synchronisation d'horloge devait être perdue, les systèmes d'extrémité ne pourraient pas détecter les messages répétés, et le système d'extrémité ne pourrait pas établir en toute sécurité le matériel de chiffrement. Il peut en résulter un déni de service ; voir au paragraphe 9.5 de la [RFC3830].

* Protection limitée contre le DoS : une vérification rapide du résumé de message permet de vérifier l'authenticité et l'intégrité d'un message avant de lancer les opérations Diffie-Hellman grosses consommatrices de CPU ou de commencer d'autres tâches consommatrices de ressources. Cela protège contre certaines attaques de déni de service : la modification malveillante des messages et les attaques de pourriels avec des messages répétés ou falsifiés. DHHMAC ne contre probablement pas explicitement les attaques réparties de déni de service sophistiquées à grande échelle qui compromettent la disponibilité du système, par exemple. Une certaine protection contre le DoS est fournie par l'inclusion de la charge utile Identité de l'initiateur dans le I_message. Cela permet au receveur de filtrer les I_messages (répétés) qui ne lui sont pas destinés et d'éviter de créer des sessions MIKEY inutiles.

* Secret parfait vers l'avant (PFS) : à la différence des protocoles MIKEY de distribution de clé fondée sur des clés pré-partagées et des clés publiques, le protocole d'accord de clé Diffie-Hellman affiche une propriété de sécurité appelée secret parfait vers l'avant. C'est-à-dire que même si la clé pré partagée à long terme est compromise à un moment, cela ne compromet pas les clés de session passées ou futures.

  Ni la variante de protocole MIKEY à clé pré-partagée ni celle à clé publique ne sont capables de fournir la propriété de sécurité du secret parfait vers l'avant. Donc, aucun des autres protocoles MIKEY n'est capable de se substituer à la propriété de PFS de Diffie-Hellman.

  À ce titre, DHHMAC et DH à signature numérique fournissent un niveau de sécurité très supérieur à cet égard à celui du protocole de distribution de clé à clé pré-partagée ou à clé publique.

* Contribution mutuelle équitable de clé : le protocole de gestion de clés Diffie-Hellman n'est pas en soi un strict protocole de distribution de clés, dans lequel l'initiateur distribue une clé à son homologue. En fait, les deux parties impliquées dans l'échange de protocole sont capables de contribuer également au trafic Diffie-Hellman commun de clé de génération de trafic de TEK. Cela réduit le risque que l'une des parties triche ou génère par inadvertance une clé de session faible. Cela fait de DHHMAC un protocole d'accord de clé équitable. On peut voir cette propriété comme une mesure de sécurité répartie supplémentaire qui augmente la robustesse de la sécurité par rapport au cas où la sécurité ne dépend que de la bonne mise en œuvre chez une seule entité.

  Pour que l'accord de clé Diffie-Hellman soit sûr, chaque partie DEVRA générer ses valeurs de xi ou xr en utilisant un générateur de nombres pseudo-aléatoires fort et imprévisible si une source de vrai aléa n'est pas disponible. De plus, ces valeurs xi ou xr DEVRONT rester confidentielles. Il est RECOMMANDÉ que ces valeurs secrètes soient détruites une fois établie la clé secrète Diffie-Hellman partagée.

* Efficacité et performances : comme le protocole MIKEY à clé publique, le protocole d'accord de clé MIKEY DHHMAC établit en toute sécurité une TGK en juste un seul aller-retour. Les autres techniques existantes de gestion de clé, comme IPsec-IKE [RFC2409], IPsec-IKEv2 [RFC4306], TLS [RFC4346], et autres schémas, ne sont pas réputés adéquats pour traiter suffisamment ces exigences de temps réel et de sécurité ; ils utilisent tous plus d'un aller-retour. Tous les protocoles de gestion de clés MIKEY sont capables d'achever leur tâche de négociation de paramètre de politique de sécurité, incluant l'accord de clé ou la distribution de clé, en un aller-retour. Cependant, les protocoles MIKEY à clé pré-partagée et MIKEY à clé publique sont tous deux capables d'achever leur tâche même en un demi aller-retour quand les messages de confirmation sont omis.

Utiliser HMAC en conjonction avec une forte fonction de hachage unidirectionnelle (comme SHA1) peut être réalisé plus efficacement dans le logiciel que dans de coûteuses opérations de clé publique. Cela donne un avantage de performances particulier à DHHMAC sur DH signé ou sur le protocole de chiffrement à clé publique.

Si un niveau de sécurité très élevé est désiré pour le secret à long terme du secret partagé négocié par Diffie-Hellman, des valeurs plus longues de hachage peuvent être déployées, comme SHA256, SHA384, ou SHA512, éventuellement en conjonction avec des groupes Diffie-Hellman plus forts. Ceci fera l'objet d'études ultérieures.

Dans le souci d'améliorer les performances et réduire le délai d'aller-retour, l'une et l'autre partie peut pré-calculer sa demie clé publique Diffie-Hellman hors ligne.

Par ailleurs, et sous des conditions raisonnables, DHHMAC consomme plus de cycles de CPU que le protocole de distribution de clé pré partagée MIKEY. Cela peut être vrai probablement aussi pour le protocole de distribution de clé MIKEY à clé publique (selon le choix des longueurs de clé privée et publique). À ce titre, on peut dire que DHHMAC a des performances meilleures, comparées aux autres variantes du protocole MIKEY.

L'utilisation d'informations d'identité facultatives (avec les contraintes déclarées au paragraphe 5.2) et les champs facultatifs de demie clé Diffie-Hellman fournit un moyen pour augmenter les performances et diminuer la consommation de bande passante du réseau.

* Infrastructure de sécurité : le présent document décrit le protocole d'accord de clé Diffie-Hellman authentifié par HMAC, qui évite complètement les signatures numériques et l'infrastructure de clé publique associée, comme cela serait nécessaire pour le protocole de distribution de clé fondée sur une clé publique RSA X.509 ou le protocole d'accord de clé Diffie-Hellman à signature numérique décrit dans MIKEY. Des infrastructures de clé publique ne peuvent pas toujours être disponibles dans certains environnements, ni être réputées adéquates pour des applications multimédia en temps réel quand des étapes supplémentaires sont nécessaires pour la validation de certificat et des méthodes de révocation de certificat avec des aller-retours supplémentaires à prendre en compte.

  DHHMAC ne dépend pas de PKI, et les mises en œuvre n'ont pas besoin de normes de PKI. Donc, on estime qu'il est beaucoup plus simple que les complexes facilités de PKI.

  DHHMAC est particulièrement attrayant dans les environnements où le provisionnement d'une clé pré-partagée a déjà été effectué.

* Pas d'opposition aux NAT : DHHMAC est capable d'opérer en douceur à travers les appareils de pare-feu/NAT pour autant que les informations d'identité protégées de l'entité d'extrémité ne soient pas une adresse IP/transport.

* Adaptabilité : comme le protocole MIKE en temps réelY Diffie-Hellman signé, DHHMAC ne s'adapte pas bien à des configurations plus grandes que des groupes d'homologue à homologue.

5.4 Hypothèses

Le présent document déclare quelques hypothèses qui ont une influence significative sur la sécurité de DHHMAC. Les conditions suivantes sont supposées :

* Les paramètres xi, xr, s, et auth_key sont à garder secrets.
* La clé pré-partagée s a suffisamment d'entropie et ne peut pas être effectivement devinée.
* La fonction pseudo aléatoire (PRF) est sûre, donne la propriété pseudo-aléatoire et maintient l'entropie.
* Un groupe Diffie-Hellman suffisamment grand et sûr est appliqué.
* L'hypothèse Diffie-Hellman tient en disant que fondamentalement même si il y a connaissance des demies clés Diffie-Hellman échangées et du groupe Diffie-Hellman, il est infaisable de calculer la TGK ou de déduire les paramètres secrets xi ou xr. Cette dernière est aussi appelée l'hypothèse de logarithme discret. Se reporter à [Handbook], [D-H], ou [DL.D-H] pour plus d'informations concernant le problème Diffie-Hellman et ses hypothèses de complexité de calcul.
* La fonction de hachage (SHA1) est sûre ; c'est-à-dire, il est infaisable de trouver par le calcul un message qui corresponde à un résumé de message donné, ou de trouver deux messages différents qui produisent le même résumé de message.
* L'algorithme HMAC est sûr et ne révèle pas la auth_key. En particulier, la sécurité dépend de la propriété d'authentification de message de la fonction de compression de la fonction de hachage H quand elle est appliquée à un seul bloc (voir la [RFC2104]).
* Une source capable de produire suffisamment de bits de (pseudo) aléa est disponible.
* Le système sur lequel fonctionne DHHMAC est suffisamment sûr.

5.5 Risque résiduel

Bien que ces hypothèses détaillées ne soient pas négligeables, les experts en sécurité estiment généralement que toutes ces hypothèses sont raisonnables et que les hypothèses faites peuvent être satisfaites en pratique à peu de frais.

Les hypothèses mathématiques et cryptographiques des propriétés de la PRF, de l'algorithme Diffie-Hellman (hypothèse de logarithme discret), l'algorithme HMAC, et les algorithmes SHA1 n'ont pas été prouvées ni infirmées pour l'instant.

Donc, un certain risque résiduel reste, qui pourrait menacer la sécurité globale à un moment imprévisible à l'avenir.

Le DHHMAC pourrait être compromis aussitôt qu'une des hypothèses mentionnées ne tiendrait plus.

Le mécanisme Diffie-Hellman est une technique de sécurité générique qui n'est pas seulement applicable aux groupes de premier ordre ou de caractéristique deux. C'est parce que l'hypothèse mathématique fondamentale est que le problème du logarithme discret est aussi très difficile dans les groupes généraux. Cela permet que Diffie-Hellman soit déployé aussi pour GF(p)*, pour des sous groupes de taille suffisante, et pour des groupes sur des courbes elliptiques. RSA ne permet pas une telle généralisation, car le cœur du problème mathématique est différent (factorisation de grands entiers).

Les clés RSA asymétriques tendent à devenir de plus en plus longues (1536 bits et plus) et donc très gourmandes en calcul. Néanmoins, la courbe elliptique Diffie-Hellman (ECDH, *Elliptic Curve Diffie-Hellman*) permet que des clés longues soient substantiellement réduites (disons à 170 bits ou moins) tout en conservant au moins le niveau de sécurité et en fournissant en pratique des avantages de performances même plus significatifs. De plus, on estime que les techniques de courbe elliptique fournissent une bien meilleure protection contre les attaques de canal latéral due à la redondance inhérente des coordonnées projectives. Pour toutes ces raisons, on peut voir le Diffie-Hellman fondé sur la courbe elliptique comme plus "à l'épreuve du futur" et robuste contre les menaces potentielles que ne l'est RSA. Noter que des variantes de MIKEY de courbe elliptique Diffie-Hellman sont définies dans [ECC].

HMAC-SHA1 est un mécanisme de sécurité clé dans DHHMAC duquel dépend la sécurité globale de MIKEY DHHMAC. MIKEY DHHMAC utilise HMAC-SHA1 en combinaison avec le schéma classique d'accord de clé Diffie-Hellman. HMAC-SHA1 est une fonction de hachage unidirectionnelle chiffrée qui implique un secret dans son calcul. DHHMAC applique HMAC-SHA1 pour la protection de la charge utile MIKEY. De même, la fonction pseudo aléatoire PRF au sein de MIKEY [RFC3830] utilise le mécanisme HMAC-SHA1 comme fonction de déduction de clé. Bien que certaines attaques aient été rapportées contre SHA1 et MD5 (voir la [RFC4270]), avec les connaissances actuelles (voir la [RFC4270], et [Hash]) aucune attaque n'a été rapportée contre le mécanisme de sécurité de HMAC-SHA1. En fait, [NMAC] prouve que HMAC possède la propriété d'une fonction pseudo aléatoire PRF en supposant seulement que la fonction de hachage (SHA1) est une fonction pseudo aléatoire. [NMAC] fournit aussi la preuve que HMAC est robuste contre les attaques de collision sur la fonction de hachage sous-jacente. On estime que MIKEY DHHMAC devrait être considéré comme assez sûr pour l'instant. Donc, il n'est pas besoin de changer le mécanisme de sécurité sous-jacent dans le protocole MIKEY DHHMAC.

Il n'est pas recommandé de déployer DHHMAC pour un autre usage que celui décrit à la Section 2. Toute application inappropriée peut conduire à des propriétés inconnues ou imprévues.

5.6 Autorisation et modèle de confiance

Fondamentalement, des remarques similaires à celles déclarées au paragraphe 4.3.2 de la [RFC3830] sur l'autorisation tiennent aussi pour DHHMAC. Cependant, comme noté précédemment, ce protocole de gestion de clés ne sert pas pour des groupes complets.

On peut voir le secret partagé pré établi comme donnant une relation de confiance pré établie entre l'initiateur et le répondant. Il en résulte un modèle de confiance beaucoup plus simple pour DHHMAC que ce ne serait le cas pour des protocoles de gestion de clés de groupe générique et de potentielles entités de groupe sans aucune relation de confiance pré définie. En conjonction avec l'hypothèse d'une clé partagée, le contrôleur de groupe commun simplifie l'établissement de communication des entités.

On peut voir la relation de confiance pré établie à travers le secret pré-partagé comme un moyen d'autorisation pré accordée, implicite. Le présent document ne définit aucun moyen particulier d'autorisation mais laisse ce sujet à l'application.

6. Remerciements

Ce document incorpore les retours précieux de relecture de Steffen Fries, Hannes Tschofenig, Fredrick Lindholm, Mary Barnes, et Russell Housley et les retours généraux du groupe de travail MSEC.

7. Considérations relatives à l'IANA

Le présent document ne définit pas son propre nouvel espace de noms pour DHHMAC, au delà de l'espace de noms de l'IANA qui a été alloué pour MIKEY ; voir la Section 10 et le paragraphe 10.1 de la [RFC3830] et l'espace de noms de charge utile MIKEY [MIKEY] de l'IANA.

Afin d'aligner le Tableau 4.1.a avec le Tableau 6.1.a de la [RFC3830], il est demandé à l'IANA d'ajouter les entrées suivantes à l'espace de noms de charge utile MIKEY :

Type de données	Valeur	Référence
DHHMAC init	7	RFC 4650
DHHMAC resp	8	RFC 4650

8. Références

8.1 Références normatives

[FIBS180-2] NIST, FIBS-PUB 180-2, "Secure Hash Standard", avril 1995, http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf.

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

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

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

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

8.2 Références pour information

[D-H] Ueli M. Maurer, S. Wolf, "The Diffie-Hellman Protocol", Designs, Codes, and Cryptography, Special Issue Public Key Cryptography, Kluwer Academic Publishers, vol. 19, pp. 147-171, 2000. ftp://ftp.inf.ethz.ch/pub/crypto/publications/MauWol00c.ps.

[DL.D-H] "Discrete Logarithms and the Diffie-Hellman Protocol", http://www.crypto.ethz.ch/research/ntc/dldh/ .

[ECC] Milne, A., Blaser, M., Brown, D., and L. Dondetti, "ECC Algorithms For MIKEY", *Travail en cours*, juin 2005.

[H.235.0] Recommandation UIT-T H.235.0, "Cadre de sécurité pour H.323 : cadre de sécurité pour les systèmes multimédia pour la série H (H.323 et autres fondées sur H.245)", (09/2005).

[H.235.7] Recommandation UIT-T H.235.7, "Cadre de sécurité pour H.323 : Usage du protocole de gestion de clé MIKEY pour le protocole de transport sûr en temps réel (SRTP) au sein de H.235", septembre 2005.

[Handbook] .J. Menezes, P. van Oorschot, S. A. Vanstone: "Handbook of Applied Cryptography", CRC Press 1996.

[Hash] Bellovin, S.M. and E.K. Rescorla: "Deploying a New Hash Algorithm", octobre 2005, http://www.cs.columbia.edu/~smb/papers/new-hash.pdf .

[MIKEY] IANA "MIKEY Payload Name Spaces per RFC 3830", voir http://www.iana.org/assignments/mikey-payloads .

[NMAC] Bellare, M.: "New Proofs for NMAC and HMAC: Security Without Collision-Resistance", http://eprint.iacr.org/2006/043.pdf , novembre 2005.

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

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

[RFC2560] M. Myers, R. Ankney, A. Malpani, S. Galperin et C. Adams, "Protocole d'état de certificat en ligne d'infrastructure de clé publique X.509 pour l'Internet - OCSP", juin 1999. *(P.S.) (Remplacée par* RFC6960)

[RFC3029] C. Adams et autres, "Protocoles de serveur de validation et de certification de données d'infrastructure de clé publique X.509 sur Internet", février 2001. *(Expérimentale)*

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

[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)

[RFC3394] J. Schaad, R. Housley, "Algorithme d'enveloppe de clés pour la norme de chiffrement évoluée (AES)", septembre 2002. *(Information)*

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

[RFC3552] E. Rescorla, B. Korver, "Lignes directrices pour la rédaction d'une section de considérations sur la sécurité dans les RFC", juillet 2003. (BCP0072)

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

[RFC4046] M. Baugher et autres, "Architecture de gestion de clé de groupe de diffusion groupée sécurisée (MSEC)", avril 2005. *(Info.)*

[RFC4158] M. Cooper et autres, "Infrastructure de clés publiques X.509 pour l'Internet : construction du chemin de certification", septembre 2005. *(Information)*

[RFC4210] C. Adams et autres, "Protocole de gestion de certificat (CMP) d'infrastructure de clé publique X.509 pour l'Internet", septembre 2005. (*MàJ par la* RFC6712) *(P.S.)*

[RFC4211] J. Schaad, "Format de message de demande de certificat d'infrastructure de clé publique X.509 pour l'Internet", septembre 2005. (*Remplace* RFC2511) *(P.S.)*

[RFC4270] P. Hoffman, B. Schneier, "Attaques contre les hachages cryptographiques dans les protocoles Internet", nov. 2005. *(Info.)*

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

[RFC4346] T. Dierks et E. Rescorla, "Protocole de sécurité de la couche Transport (TLS) version 1.1", avril 2006. *(Remplace* RFC2246 *; Remplacée par* RFC5246 *; MàJ par* RFC4366, 4680, 4681, 5746, 6176, 7465, 7507, 7919)

[RFC4535] H. Harney et autres, "GSAKMP : protocole de gestion de clés d'association de groupe sécurisé", juin 2006. *(P.S.)*

[RFC4738] D. Ignjatic et autres, "MIKEY-RSA-R : un mode supplémentaire de distribution de clés dans le chiffrement Internet multimédia (MIKEY)", novembre 2006. (*MàJ* RFC3830) *(P.S.)*

Appendice A. Usage de MIKEY-DHHMAC dans H.235

Cet appendice donne des informations générales sur la façon dont MIKEY-DHHMAC peut être appliqué dans certains environnements multimédia fondés sur H.323. Généralement, MIKEY est applicable aux applications multimédia incluant la téléphonie IP. [H.235.7] décrit divers cas d'utilisation des protocoles de gestion de clés MIKEY (MIKEY-PS, MIKEY-PK, MIKEY-DHSIGN et MIKEY-DHHMAC) dans le but d'établir du matériel de chiffrement de TGK dans les points d'extrémité H.323. Les TGK sont ensuite utilisées pour le chiffrement du support en appliquant SRTP [RFC3711]. Les scénarios visés incluent le point à point avec un ou plusieurs portiers intermédiaires (de confiance ou partiellement de confiance) entre.

Un cas d'usage particulier vise celui où MIKEY-DHHMAC établit une connexion de supports d'un point d'extrémité B appelant (à travers un portier) à un autre point d'extrémité A qui est situé dans la même zone que le portier. Alors que les points d'extrémité A et B ne partagent normalement aucune auth_key a priori, des moyens d'échange de protocole séparés sont réalisés en dehors de la procédure d'établissement d'appel réelle pour établir une auth_key pour le temps où les points d'extrémité s'enregistrent auprès du portier ; de tels protocoles existent [H.235.7] mais ne sont pas montrés dans le présent document. La auth_key entre les points d'extrémité est utilisée pour authentifier et protéger l'intégrité des messages MIKEY-DHHMAC.

Pour établir un appel, on suppose que le point d'extrémité B (EP-B) a obtenu la permission du portier (non montré). Le point B a l'appelant qui construit le I_message MIKEY-DHHMAC (voir la section 3) et il envoie le I_message encapsulé dans le H.323-SETUP au point d'extrémité A (EP-A). Un portier d'acheminement (GK) va transmettre ce message au point d'extrémité B ; dans le cas d'un portier non acheminant, le point d'extrémité B envoie le SETUP directement au point d'extrémité A. Dans l'un et l'autre cas, les mécanismes de sécurité inhérents à H.323 [H.235.0] sont appliqués pour protéger le message (encapsulation) durant le transfert. Ceci n'est pas décrit ici. Le point d'extrémité A receveur est capable de vérifier le I_message convoyé et peut calculer une TGK. En supposant que le point d'extrémité A va accepter l'appel, EP-A construit alors le R_message MIKEY-DHHMAC et renvoie la réponse au titre du message "CallProceeding-to-Connect" au point d'extrémité appelant B (éventuellement à travers un portier d'acheminement). EP-B traite le R_message convoyé pour calculer la même TGK que le point d'extrémité appelé A.

1.) EP-B -> (GK) -> EP-A: SETUP(I_fwd_message [, I_rev_message])
2.) EP-A -> (GK) -> EP-B: CallProceeding-to-CONNECT(R_fwd_message [, R_rev_message])

Note : si il est nécessaire d'établir des TGK directionnelles pour des liaisons bidirectionnelles dans les deux directions B->A et A->B, alors le point d'extrémité appelant B instancie deux fois le protocole DHHMAC : une fois dans la direction B->A en utilisant I_fwd_message et une autre fois en parallèle dans la direction A->B en utilisant I_rev_message. Dans ce cas, deux I_messages MIKEY-DHHMAC sont encapsulés dans SETUP (I_fwd_message et I_rev_message) et deux R_messages MIKEY-DHHMAC (R_fwd_message et R_rev_message) sont encapsulés dans CallProceeding-to-CONNECT. Le I_rev_message correspond au I_fwd_message. Autrement, le point d'extrémité appelé A peut instancier le protocole DHHMAC dans un cours distinct avec le point d'extrémité B (non montré) ; cependant, cela exige qu'une troisième prise de contact ait lieu.

Pour les détails sur la façon dont les protocoles MIKEY peuvent être déployés avec H.235, voir [H.235.7].

Adresse de l'auteur

Martin Euchner
Hofmannstr. 51
81359 Munich,
Germany

téléphone : +49 89 722 55790
Fax: +49 89 722 62366
mél : martin_euchner@hotmail.com

Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est fourni par l'activité de soutien administatif (IASA) de l'IETF.