Groupe de travail Réseau
Request for Comments : 4868
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

S. Kelly, Aruba Networks
S. Frankel, NIST
mai 2007

Utilisation de HMAC-SHA-256, HMAC-SHA-384, et HMAC-SHA-512 avec IPsec

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.
(La présente traduction incorpore les errata 1785 et 5507)

Résumé
La présente spécification décrit l'utilisation du code d'authentification de message par hachage de clé (HMAC, *Hashed Message Authentication Code*) en conjonction avec les algorithmes SHA-256, SHA-384, et SHA-512 dans IPsec. Ces algorithmes peuvent être utilisés comme base des mécanismes d'authentification de l'origine des données et de vérification d'intégrité pour les protocoles d'en-tête d'authentification (AH, *Authentication Header*) d'encapsulation de charge utile de sécurité (ESP, *Encapsulating Security Payload*) d'échange de clé Internet (IKE, *Internet Key Exchange*) et IKEv2, et aussi comme fonctions pseudo aléatoires (PRF, *Pseudo-Random Function*) pour IKE et IKEv2. Des longueurs de résultat tronquées sont spécifiées pour les variantes en rapport avec l'authentification, avec les algorithmes correspondants désignés comme HMAC-SHA-256-128, HMAC-SHA-384-192, et HMAC-SHA-512-256. Les variantes de PRF ne sont pas tronquées, et sont appelées PRF-HMAC-SHA-256, PRF-HMAC-SHA-384, et PRF-HMAC-SHA-512.

Table des matières

1. Introduction

Le présent document spécifie l'utilisation de SHA-256, SHA-384, et SHA-512 [SHA2-1] combinée avec HMAC [RFC2104] comme mécanismes d'authentification de l'origine des données et de vérification de l'intégrité pour les protocoles IPsec AH [RFC4302], ESP [RFC4303], IKE [RFC2409], et IKEv2 [RFC4306]. La troncature du résultat est spécifiée pour ces variantes, avec les algorithmes correspondants désignés comme HMAC-SHA-256-128, HMAC-SHA-

384-192, et HMAC-SHA-512-256. Ces longueurs de troncature sont choisies en accord avec les limites d'attaque d'anniversaire pour chaque algorithme.

La présente spécification décrit aussi des variantes non tronquées de ces algorithmes comme des fonctions pseudo aléatoires (PRF) à utiliser avec IKE et IKEv2, et ces algorithmes sont appelés PRF-HMAC-SHA-256, PRF-HMAC-SHA-384, et PRF-HMAC-SHA-512. Pour faciliter les références, ces algorithmes de PRF et variantes d'authentification décrits ci-dessus sont collectivement appelés ci-dessous les algorithmes "HMAC-SHA-256+".

Le but des variantes de PRF est de fournir des fonctions pseudo aléatoires sûres convenables pour générer du matériel de chiffrement et autres quantités numériques spécifiques d'un protocole, tandis que le but des variantes d'authentification est d'assurer que les paquets sont authentiques et ne peuvent pas être modifiés dans le transit. La sécurité relative de HMAC-SHA-256+ quand utilisé dans l'un ou l'autre cas dépend de la portée de la distribution et de l'imprévisibilité de la clé secrète associée. Si la clé est imprévisible et connue seulement de l'envoyeur et du receveur, ces algorithmes assurent que seules les parties qui détiennent une clé identique peuvent déduire les valeurs associées.

2. Algorithmes HMAC-SHA-256+

[SHA2-1] et [SHA2-2] décrivent les algorithmes SHA-256, SHA-384, et SHA-512 sous-jacents, tandis que la [RFC2104] décrit l'algorithme HMAC. L'algorithme HMAC donne un cadre pour insérer divers algorithmes de hachage comme SHA-256, et la [RFC4634] décrit l'usage combiné de ces algorithmes. Les paragraphes qui suivent décrivent les diverses caractéristiques et exigences des algorithmes HMAC-SHA-256+ quand ils sont utilisés avec IPsec.

2.1 Matériel de chiffrement

Les exigences sur le matériel de chiffrement varient selon que l'algorithme fonctionne comme PRF ou comme mécanisme d'authentification/intégrité. Dans le cas de l'authentification/intégrité, les longueurs de clés sont fixées conformément à la longueur de sortie de l'algorithme utilisé. Dans le cas des PRF, les longueurs de clé sont variables, mais des lignes directrices sont données pour assurer l'interopérabilité. Ces distinctions sont décrites plus loin.

Avant de décrire les exigences de clé pour chaque usage, il est important de préciser quelques termes :

Taille de bloc : taille du bloc de données sur lequel opère l'algorithme de hachage sous-jacent. Pour SHA-256, c'est 512 bits, pour SHA-384 et SHA-512, c'est 1024 bits.

Longueur de résultat : taille de la valeur du hachage produit par l'algorithme de hachage sous-jacent. Pour SHA-256, c'est 256 bits, pour SHA-384 c'est 384 bits, et pour SHA-512, c'est 512 bits.

Longueur d'authentifiant : taille de "l'authentifiant" en bits. Cela s'applique seulement aux algorithmes relatifs à l'authentification/intégrité, et se réfère à la longueur de bits restante après troncature. Dans la présente spécification, c'est toujours la moitié de la longueur de résultat de l'algorithme de hachage sous-jacent.

2.1.1 Usage de l'authentification de l'origine des données et de la vérification de l'intégrité

Les algorithmes HMAC-SHA-256+ sont à clé secrète. Bien qu'aucune longueur de clé fixe ne soit spécifiée dans la [RFC2104], la présente spécification exige que quand elle est utilisée comme algorithme d'intégrité/authentification, une longueur de clé fixe égale à la longueur de résultat des fonctions de hachage DOIT être prise en charge, et des longueurs de clé autres que la longueur de résultat de la fonction de hachage associée NE DOIVENT PAS être acceptées.

Ces restrictions sur la longueur de clé sont fondées en partie sur les recommandations de la [RFC2104] (les longueurs de clé inféieures à la longueur du résultat diminuent la force de la sécurité, et les clés plus longues que la longueur du résultat n'augmentent pas de façon significative la force de la sécurité) et en partie parce que permettre des longueurs de clé variables pour les fonctions d'authentifiant IPsec créerait des problèmes d'interopérabilité.

2.1.2 Usage de la fonction pseudo aléatoire (PRF)

IKE et IKEv2 utilisent des PRF pour générer du matériel de chiffrement et pour l'authentification de l'association de sécurité IKE. La spécification IKEv2 différencie les PRF avec tailles de clé fixes et celles avec tailles de clé variables, et

donc on donne des directives particulières pour cela.

Quand une PRF décrite dans le présent document est utilisée avec IKE ou IKEv2, elle est considérée avoir une longueur de clé variable, et les clés sont déduites de la façon suivante (noter qu'on répète simplement ce qui est spécifié dans la [RFC2104]) :

- Si la longueur de la clé est exactement la taille de bloc de l'algorithme, on l'utilise telle qu'elle est.

- Si la clé est plus courte que la taille de bloc, on la rallonge à exactement la taille de bloc en la bourrant sur la droite avec des bits à zéro. Cependant, noter que la [RFC2104] déconseille fortement une longueur de clé inféieure à la longueur de résultat. Néanmoins, on décrit ici le traitement de longueurs plus courtes eu égard à l'existence de longueurs plus courtes normalement choisies pour les clés pré partagées IKE ou IKEv2.

- Si la clé est plus longue que la taille de bloc, on la raccourcit en calculant le résultat d'algorithme de hachage correspondant sur la valeur de clé entière, et en traitant la valeur de sortie résultante comme clé HMAC. Noter que cela va toujours résulter en une clé qui fait moins que la taille de bloc, et cette valeur de clé va donc exiger un bourrage de zéros (comme décrit ci-dessus) avant usage.

2.1.3 Aléa et force de clé

La [RFC2104] discute des exigences pour le matériel de chiffrement, incluant une exigence de force d'aléa. Donc, une fonction pseudo aléatoire forte DOIT être utilisée pour générer la clé requise à utiliser avec HMAC-SHA-256+. Au moment de la rédaction du présent document, il n'y a pas de clés faibles publiées à utiliser avec des algorithmes HMAC-SHA-256+.

2.1.4 Distribution des clés

La [RFC4301] décrit le mécanisme général pour obtenir le matériel de chiffrement quand plusieurs clés sont requises pour une seule SA (par exemple, quand une SA ESP exige une clé pour la confidentialité et une clé pour l'authentification). Afin de fournir l'authentification de l'origine des données et la vérification de l'intégrité, le mécanisme de distribution de la clé doit assurer que des clés uniques sont allouées et qu'elles sont distribuées seulement aux parties qui participent à la communication.

2.1.5 Rafraîchissement des clés

Actuellement, il n'y a pas d'attaque pratique contre les algorithmes recommandés ici, et en particulier contre les tailles de clé recommandées ici. Cependant, comme noté dans la [RFC2104] "un rafraîchissement périodique de la clé est une pratique de sécurité fondamentale qui aide contre de potentielles faiblesses de la fonction et des clés, et limite les dommages d'une clé exposée".

Dans cette perspective, la présente spécification exige des clés de 256, 384, ou 512 bits produites par une forte PRF à utiliser comme un MAC. Une attaque en force brute contre de telles clés prendrait plus de temps à monter que l'univers n'a eu d'existence. Par ailleurs, des clés faibles (par exemple, des mots du dictionnaire) seraient dramatiquement moins résistantes à l'attaque. Il est important de noter ces points, avec le modèle spécifique de menaces pour une application particulière et l'état actuel l'art à l'égard des attaques contre SHA-256, SHA-384, et SHA-512 quand on détermine une limite supérieure appropriée pour les durées de vie de clés HMAC.

2.2 Bourrage

Les algorithmes HMAC-SHA-256 opèrent sur des blocs de données de 512 bits, tandis que les algorithmes HMAC-SHA-384 et HMAC-SHA-512 opèrent sur des blocs de données de 1024 bits. Les exigences de bourrage sont spécifiées dans [SHA2-1] au titre des algorithmes SHA-256, SHA-384, et SHA-512 sous-jacent, de sorte que si on met en œuvre conformément à [SHA2-1], on n'a pas besoin d'ajouter de bourrage supplémentaire pour autant que les algorithmes HMAC-SHA-256+ spécifiés ici sont concernés. À l'égard du "bourrage implicite de paquet" défini dans la [RFC4302], aucun bourrage implicite de paquet n'est requis.

2.3 Troncature

Les algorithmes HMAC-SHA-256+ produisent chacun une valeur de nnn bits, où nnn correspond à la longueur en bits de résultat de l'algorithme, par exemple, HMAC-SHA-nnn. Pour l'utiliser comme un authentifiant, cette valeur de nnn bits peut être tronquée comme décrit dans la [RFC2104]. Quand elle est utilisée comme algorithme d'authentification d'origine des données et de vérification d'intégrité dans ESP, AH, IKE, ou IKEv2, une valeur tronquée en utilisant les nnn/2 premiers bits – exactement la moitié de la taille du résultat de l'algorithme – DOIT être supportée. Aucune autre longueur de valeur d'authentifiant n'est acceptée par la présente spécification.

À l'envoi, la valeur tronquée est mémorisée au sein du champ Authentifiant. À réception, la valeur de nnn bits entière est calculée et les nnn/2 premiers bits sont comparés à la valeur mémorisée dans le champ Authentifiant, avec la valeur de 'nnn' qui dépend de l'algorithme négocié.

La [RFC2104] discute les avantages potentiels pour la sécurité résultant de la troncature de la valeur du MAC de résultat, et en général, encourage les utilisateurs de HMAC à effectuer la troncature de MAC. Dans le contexte de IPsec, une longueur de troncature de nnn/2 bits est choisie parce que elle correspond aux limites de l'attaque de l'anniversaire pour chaque algorithme HMAC-SHA-256+, et elle sert simultanément à minimiser les bits supplémentaires sur le réseau résultant de l'utilisation de cette facilité.

2.4 Utilisation de HMAC-SHA-256+ comme PRF dans IKE et IKEv2

L'algorithme PRF-HMAC-SHA-256 est identique à HMAC-SHA-256-128, excepté que les clés de longueur variable sont permises, et que l'étape de troncature N'est PAS effectuée. De même, les mises en œuvre de PRF-HMAC-SHA-384 et PRF-HMAC-SHA-512 sont identiques à celles de HMAC-SHA-384-192 et HMAC-SHA-512-256 respectivement, excepté que là encore, les clés de longueur variable sont permises, et que la troncature N'est PAS effectuée.

2.5 Interactions avec les mécanismes de chiffrement ESP, IKE, ou IKEv2

Au moment de cette rédaction, il n'y a pas de problème connu qui empêche l'utilisation des algorithmes HMAC-SHA-256+ avec aucun algorithme de chiffrement spécifique.

2.6. Résumé des paramètres de HMAC-SHA-256+

Le tableau suivant sert à résumer les diverses quantités associées aux algorithmes HMAC-SHA-256+.

Identifiant d'algorithme	Taille de bloc	Long. du résultat	Long. tronquée	Long. de clé	Type d'algorithme
HMAC-SHA-256-128	512	256	128	256	auth/integ
HMAC-SHA-384-192	1024	384	192	384	auth/integ
HMAC-SHA-512-256	1024	512	256	512	auth/integ
PRF-HMAC-SHA-256	512	256	(aucune)	variable	PRF
PRF-HMAC-SHA-384	1024	384	(aucune)	variable	PRF
PRF-HMAC-SHA-512	1024	512	(aucune)	variable	PRF

2.7 Vecteurs d'essai

Les cas d'essai suivants incluent la clé, les données, et l'authentifiant résultant, et/ou les valeurs de PRF pour chaque algorithme. Les valeurs des clés et des données sont soit des chaînes de caractères ASCII (entourées de guillemets) soit des nombres hexadécimaux. Si la valeur est une chaîne de caractères ASCII, alors le calcul de HMAC pour le cas d'essai correspondant N'inclut PAS de caractère nul ('\0') à la fin de la chaîne. Les valeurs de HMAC calculées sont toutes des nombres hexadécimaux.

2.7.1 Vecteurs d'essai de PRF

Ces cas d'essai sont empruntés à la [RFC4231]. Pour la référence des mises en œuvre des algorithmes de hachage sous-jacents, voir la [RFC4634]. Noter que pour les besoins des essais, le résultat de la PRF est considéré comme étant simplement le résultat non tronqué de l'algorithme.

Cas d'essai de PRF-1 :

Clé = 0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b (20 octets)
Données = 4869205468657265 ("Hi There")
PRF-HMAC-SHA-256 = b0344c61d8db38535ca8afceaf0bf12b881dc200c9833da726e9376c2e32cff7
PRF-HMAC-SHA-384 = afd03944d84895626b0825f4ab46907f15f9dadbe4101ec682aa034c7cebc59c
faea9ea9076ede7f4af152e8b2fa9cb6
PRF-HMAC-SHA-512 = 87aa7cdea5ef619d4ff0b4241a1d6cb02379f4e2ce4ec2787ad0b30545e17cde
daa833b7d6b8a702038b274eaea3f4e4be9d914eeb61f1702e696c203a126854

Cas d'essai de PRF-2 :
Clé = 4a656665 ("Jefe")
Données = 7768617420646f2079612077616e7420 ("what do ya want ")
666f72206e6f7468696e673f ("for nothing?")
PRF-HMAC-SHA-256 = 5bdcc146bf60754e6a042426089575c75a003f089d2739839dec58b964ec3843
PRF-HMAC-SHA-384 = af45d2e376484031617f78d2b58a6b1b9c7ef464f5a01b47e42ec3736322445e
8e2240ca5e69e2c78b3239ecfab21649
PRF-HMAC-SHA-512 = 164b7a7bfcf819e2e395fbe73b56e0a387bd64222e831fd610270cd7ea250554
9758bf75c05a994a6d034f65f8f0e6fdcaeab1a34d4a6b4b636e070a38bce737

Cas d'essai de PRF-3 :
Clé = aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaa (20 octets)
Données = dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddddddddddd (50 octets)
PRF-HMAC-SHA-256 = 773ea91e36800e46854db8ebd09181a72959098b3ef8c122d9635514ced565fe
PRF-HMAC-SHA-384 = 88062608d3e6ad8a0aa2ace014c8a86f0aa635d947ac9febe83ef4e55966144b
2a5ab39dc13814b94e3ab6e101a34f27
PRF-HMAC-SHA-512 = fa73b0089d56a284efb0f0756c890be9b1b5dbdd8ee81a3655f83e33b2279d39
bf3e848279a722c806b485a47e67c807b946a337bee8942674278859e13292fb

Cas d'essai de PRF-4 :
Clé = 0102030405060708090a0b0c0d0e0f10111213141516171819 (25 octets)
Données = cdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcd
cdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcd (50 octets)
PRF-HMAC-SHA-256 = 82558a389a443c0ea4cc819899f2083a85f0faa3e578f8077a2e3ff46729665b
PRF-HMAC-SHA-384 = 3e8a69b7783c25851933ab6290af6ca77a9981480850009cc5577c6e1f573b4e
6801dd23c4a7d679ccf8a386c674cffb
PRF-HMAC-SHA-512 = b0ba465637458c6990e5a8c5f61d4af7e576d97ff94b872de76f8050361ee3db
a91ca5c11aa25eb4d679275cc5788063a5f19741120c4f2de2adebeb10a298dd

Cas d'essai de PRF-5 :
Clé = aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa (131 octets)
Données = 54657374205573696e67204c61726765 ("Test Using Large")
72205468616e20426c6f636b2d53697a ("r Than Block-Siz")
65204b6579202d2048617368204b6579 ("e Key - Hash Key")
204669727374 (" First")
PRF-HMAC-SHA-256 = 60e431591ee0b67f0d8a26aacbf5b77f 8e0bc6213728c5140546040f0ee37f54
PRF-HMAC-SHA-384 = 4ece084485813e9088d2c63a041bc5b44f9ef1012a2b588f3cd11f05033ac4c6
0c2ef6ab4030fe8296248df163f44952
PRF-HMAC-SHA-512 = 80b24263c7c1a3ebb71493c1dd7be8b49b46d1f41b4aeec1121b013783f8f352
6b56d037e05f2598bd0fd2215d6a1e5295e64f73f63f0aec8b915a985d786598

Cas d'essai de PRF-6 :
Clé = aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa (131 octets)
Données = 546869732069732061207465737420 75 ("This est a test u")
73696e672061206c6172676572207468 ("sing a larger th")
616e20626c6f636b2d73697a65206b65 ("an block-taille ke")

7920616e642061206c61726765722074 ("y et a larger t")
68616e20626c6f636b2d73697a652064 ("han block-taille d")
6174612e20546865206b6579206e6565 ("ata. The key nee")
647320746f20626520686173686564 20 ("ds to être hashed ")
6265666f72652065696e6720757365 ("before being use")
642062792074686520484d414320616c ("d by the HMAC al")
676f726974686d2e ("gorithm.")

PRF-HMAC-SHA-256 = 9b09ffa71b942fcb27635fbcd5b0e944bfdc63644f0713938a7f51535c3a35e2
PRF-HMAC-SHA-384 = 6617178e941f020d351e2f254e8fd32c602420feb0b8fb9adccebb82461e99c5
a678cc31e799176d3860e6110c46523e
PRF-HMAC-SHA-512 = e37b6a775dc87dbaa4dfa9f96e5e3ffddebd71f8867289865df5a32d20cdc944
b6022cac3c4982b10d5eeb55c3e4de15134676fb6de0446065c97440fa8c6a58

2.7.2 Vecteurs d'essai d'authentifiant

Les paragraphes suivants sont des cas d'essai pour HMAC-SHA256-128, HMAC-SHA384-192, et HMAC-SHA512-256. Les résultats de PRF sont aussi inclus dans un but pratique. Ces cas d'essai ont été générés en utilisant le code de référence SHA256+ fourni dans la [RFC4634].

2.7.2.1 Vecteurs d'essai d'authentification SHA256

Cas d'essai de AUTH256-1 :
Clé = 0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b (32 octets)
Données = 4869205468657265 ("Hi There")
PRF-HMAC-SHA-256 = 198a607eb44bfbc69903a0f1cf2bbdc5ba0aa3f3d9ae3c1c7a3b1696a0b68cf7
HMAC-SHA-256-128 = 198a607eb44bfbc69903a0f1cf2bbdc5

Cas d'essai de AUTH256-2 :
Clé = 4a6566654a6566654a6566654a656665 ("JefeJefeJefeJefe")
4a6566654a6566654a6566654a656665 ("JefeJefeJefeJefe")
Données = 7768617420646f2079612077616e7420 ("what do ya want ")
666f72206e6f7468696e673f ("for nothing?")
PRF-HMAC-SHA-256 = 167f928588c5cc2eef8e3093caa0e87c9ff566a14794aa61648d81621a2a40c6
HMAC-SHA-256-128 = 167f928588c5cc2eef8e3093caa0e87c

Cas d'essai de AUTH256-3 :
Clé = aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa (32 octets)
Data = ddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddddddddddd (50 octets)
PRF-HMAC-SHA-256 = cdcb1220d1ecccea91e53aba3092f962e549fe6ce9ed7fdc43191fbde45c30b0
HMAC-SHA-256-128 = cdcb1220d1ecccea91e53aba3092f962

Cas d'essai de AUTH256-4 :
Key = 0102030405060708090a0b0c0d0e0f10111213141516171819 1a1b1c1d1e1f20 (32 octets)
Données = cdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcd
cdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcd (50 octets)
PRF-HMAC-SHA-256 = 372efcf9b40b35c2115b1346903d2ef42fced46f0846e7257bb156d3d7b30d3f
HMAC-SHA-256-128 = 372efcf9b40b35c2115b1346903d2ef4

2.7.2.2 Vecteurs d'essai d'authentification SHA384

Cas d'essai de AUTH384-1 :
Clé = 0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b
0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b (48 octets)
Données = 4869205468657265 ("Hi There")
PRF-HMAC-SHA-384 = b6a8d5636f5c6a7224f9977dcf7ee6c7fb6d0c48cbdee9737a959796489bddbc
4c5df61d5b3297b4fb68dab9f1b582c2
HMAC-SHA-384-128 = b6a8d5636f5c6a7224f9977dcf7ee6c7fb6d0c48cbdee973

Cas d'essai de AUTH384-2 :
Clé = 4a6566654a6566654a6566654a656665 ("JefeJefeJefeJefe")
4a6566654a6566654a6566654a656665 ("JefeJefeJefeJefe")
4a6566654a6566654a6566654a656665 ("JefeJefeJefeJefe")
Données = 7768617420646f2079612077616e7420 ("what do ya want ")
666f72206e6f7468696e673f ("for nothing?")
PRF-HMAC-SHA-384 = 2c7353974f1842fd66d53c452ca42122b28c0b594cfb184da86a368e9b8e16f5
349524ca4e82400cbde0686d403371c9
HMAC-SHA-384-192 = 2c7353974f1842fd66d53c452ca42122b28c0b594cfb184d

Cas d'essai de AUTH384-3 :
Clé = aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa (48 octets)
Données = dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddddddddd (50 octets)
PRF-HMAC-SHA-384 = 809f439be00274321d4a538652164b53554a508184a0c3160353e3428597003d
35914a18770f9443987054944b7c4b4a
HMAC-SHA-384-192 = 809f439be00274321d4a538652164b53554a508184a0c316

Cas d'essai de AUTH384-4 :
Clé = 0102030405060708090a0b0c0d0e0f10111213141516171819 1a1b1c1d1e1f20
2122232425262728292a2b2c2d2e2f30 (48 octets)
Données = cdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcd
cdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcd (50 octets)
PRF-HMAC-SHA-384 = 5b540085c6e6358096532b2493609ed1cb298f774f87bb5c2ebf182c83cc7428
707fb92eab2536a5812258228bc96687
HMAC-SHA-384-192 = 5b540085c6e6358096532b2493609ed1cb298f774f87bb5c

2.7.2.3 Vecteurs d'essai d'authentification SHA512

Cas d'essai de AUTH512-1 :
Clé = 0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b
0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b0b (64 octets)
Données = 4869205468657265 ("Hi There")
PRF-HMAC-SHA-512 = 637edc6e01dce7e6742a99451aae82df23da3e92439e590e43e761b33e910fb8
ac2878ebd5803f6f0b61dbce5e251ff8789a4722c1be65aea45fd464e89f8f5b
HMAC-SHA-512-256 = 637edc6e01dce7e6742a99451aae82df23da3e92439e590e43e761b33e910fb8

Cas d'essai de AUTH512-2 :
Clé = 4a6566654a6566654a6566654a656665 ("JefeJefeJefeJefe")
4a6566654a6566654a6566654a656665 ("JefeJefeJefeJefe")
4a6566654a6566654a6566654a656665 ("JefeJefeJefeJefe")
4a6566654a6566654a6566654a656665 ("JefeJefeJefeJefe")
Données = 7768617420646f2079612077616e7420 ("what do ya want ")
666f72206e6f7468696e673f ("for nothing?")
PRF-HMAC-SHA-512 = cb370917ae8a7ce28cfd1d8f4705d6141c173b2a9362c15df235dfb251b15454
6aa334ae9fb9afc2184932d8695e397bfa0ffb93466cfcceaae38c833b7dba38
HMAC-SHA-512-256 = cb370917ae8a7ce28cfd1d8f4705d6141c173b2a9362c15df235dfb251b15454

Cas d'essai de AUTH512-3 :
Clé = aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa (64 octets)
Données = dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddddddddd (50 octets)
PRF-HMAC-SHA-512 = 2ee7acd783624ca9398710f3ee05ae41b9f9b0510c87e49e586cc9bf961733d8
623c7b55cebefccf02d5581acc1c9d5fb1ff68a1de45509fbe4da9a433922655
HMAC-SHA-512-256 = 2ee7acd783624ca9398710f3ee05ae41b9f9b0510c87e49e586cc9bf961733d8

Cas d'essai de AUTH512-4 :
Clé = 0102030405060708090a0b0c0d0e0f10111213141516171819
1a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132

333435363738393a3b3c3d3e3f40 (64 octets)
Données = cdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcd
cdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcdcd (50 octets)
PRF-HMAC-SHA-512 = 5e6688e5a3daec826ca32eaea224eff5e700628947470e13ad01302561bab108
b8c48cbc6b807dcfbd850521a685babc7eae4a2a2e660dc0e86b931d65503fd2
HMAC-SHA-512-256 = 5e6688e5a3daec826ca32eaea224eff5e700628947470e13ad01302561bab108

3. Considérations sur la sécurité

Dans un sens général, la sécurité fournie par les algorithmes HMAC-SHA-256+ se fonde sur la force de l'algorithme de hachage sous-jacent, et sur la force supplémentaire déduite de la construction HMAC. Au moment de cette rédaction, il n'y a pas d'attaque cryptographique pratique contre SHA-256, SHA-384, SHA-512, ou HMAC. Cependant, comme avec tout algorithme cryptographique, un composant important de la force de ces algorithmes réside dans la correction de la mise en œuvre de l'algorithme, de la sécurité du mécanisme de gestion de la clé, de la force de la clé secrète associée, et de la correction de la mise en œuvre dans tous les systèmes participants. La présente spécification contient des vecteurs d'essai pour aider à vérifier la correction de la mise en œuvre de l'algorithme, mais ils ne vérifient en aucune façon la correction (ou la sécurité) de l'infrastructure de sécurité environnante.

3.1 Longueur de clé HMAC contre longueur de troncation

Il y a des différences importantes entre les niveaux de sécurité offerts par HMAC-SHA1-96 [RFC2404] et les algorithmes HMAC-SHA-256+, mais il y a aussi des considérations qui sont un peu contre intuitives. Il y a deux axes différents le long desquels on jauge la sécurité de ces algorithmes : la longueur du résultat de HMAC et la longueur de la clé HMAC. Si on suppose que la clé HMAC est un secret bien gardé qui peut seulement être déterminé par des attaques hors ligne sur des valeurs observées, et si sa longueur est inférieure ou égale à la longueur du résultat de l'algorithme de hachage sous-jacent, alors la force de la clé est directement proportionnelle à sa longueur. Et si on suppose qu'un adversaire n'a pas connaissance de la clé HMAC, alors la probabilité de deviner une valeur correcte de MAC pour un paquet donné est directement proportionnelle à la longueur du résultat du HMAC.

La présente spécification définit une troncature pour sortir des longueurs de 128, 192, ou 256 bits. Il est important de noter que pour l'instant, il n'est pas clair que HMAC-SHA-256 avec une longueur de troncature de 128 bits est plus sûr que HMAC-SHA1 avec la même longueur de troncature, en supposant que l'adversaire n'a pas connaissance de la clé HMAC. C'est parce que dans de tels cas, l'adversaire doit prédire seulement les bits qui restent après la troncature. Comme dans les deux cas cette longueur de résultat est la même (128 bits) les chances que l'adversaire devine correctement la valeur sont aussi les mêmes dans l'un et l'autre cas : 1 sur 2^128. Là encore, si on suppose que la clé HMAC reste inconnue de l'attaquant, seul un biais dans un des algorithmes va distinguer l'un de l'autre. Actuellement, l'existence d'aucun biais de cette sorte n'est connue, ni dans HMAC-SHA1, ni dans HMAC-SHA-256+.

Si, par ailleurs, l'attaquant est concentré sur deviner la clé HMAC, et si on suppose que les algorithmes de hachage sont indistinguables quand ils sont vus comme des PRF, alors la longueur de la clé HMAC donne une mesure directe de la sécurité sous jacente : plus longue est la clé, plus elle est difficile à deviner. Cela signifie que à l'égard des attaques passives sur la clé HMAC, la taille importe - et les algorithmes HMAC-SHA-256+ fournissent plus de sécurité à cet égard que HMAC-SHA1-96.

4. Considérations relatives à l'IANA

Le présent document ne spécifie pas les conventions pour utiliser SHA256+ pour les négociations de IKE phase 1, sauf à noter que l'IANA a fait les allocations d'attribut d'algorithme de hachage IKE suivantes :
SHA2-256 : 4
SHA2-384 : 5
SHA2-512 : 6

Pour les négociations de IKE phase 2, l'IANA a alloué les identifiants d'algorithme d'authentification suivants :
HMAC-SHA2-256 : 5
HMAC-SHA2-384 : 6
HMAC-SHA2-512 : 7

Pour l'utilisation de HMAC-SHA-256+ comme PRF dans IKEv2, l'IANA a alloué les identifiants de transformation de fonction pseudo aléatoire (type 2) IKEv2 :
PRF_HMAC_SHA2_256 : 5
PRF_HMAC_SHA2_384 : 6
PRF_HMAC_SHA2_512 : 7

Pour l'utilisation des algorithmes HMAC-SHA-256+ pour l'authentification de l'origine des données et la vérification d'intégrité dans IKEv2, ESP, ou AH, l'IANA a alloué les identifiants de transformation d'intégrité (type 3) IKEv2 :
AUTH_HMAC_SHA2_256_128 : 12
AUTH_HMAC_SHA2_384_192 : 13
AUTH_HMAC_SHA2_512_256 : 14

5. Remerciements

Des portions de ce texte ont été empruntés des [RFC2404] et [RFC4231]. Merci à Hugo Krawczyk de ses commentaires et recommandations sur les premières révisions du présent document, et merci aussi à Russ Housley et Steve Bellovin pour divers commentaires et recommandations relatifs à la sécurité.

6. Références

6.1 Références normatives

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

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

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

[RFC4231] M. Nystrom, "Identifiants et vecteurs d'essai pour HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, et HMAC-SHA-512", décembre 2005. *(P.S.)*

[RFC4301] S. Kent et K. Seo, "Architecture de sécurité pour le protocole Internet", décembre 2005. *(P.S.) (Remplace la* RFC2401)

[RFC4302] S. Kent, "En-tête d'authentification IP", décembre 2005. *(P.S.)*

[RFC4303] S. Kent, "Encapsulation de charge utile de sécurité dans IP (ESP)", décembre 2005. (*Remplace* RFC2406) *(P.S.)*

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

[RFC4634] D. Eastlake 3rd, T. Hansen, "Algorithmes de hachage sécurisé aux USA (SHA et HMAC-SHA)", juillet 2006. *(Info.)*

[SHA2-1] NIST, "FIPS PUB 180-2 'Specifications for the Secure Hash Standard'", février 2004, <http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf >.

6.2 Références pour information

[SHA2-2] NIST, "Descriptions of SHA-256, SHA-384, and SHA-512", mai 2001, <http://csrc.nist.gov/cryptval/shs/sha256-384-512.pdf >.

Adresse des auteurs

Scott G. Kelly
Aruba Networks
1322 Crossman Ave
Sunnyvale, CA 94089
US
mél : scott@hyperthought.com

Sheila Frankel
NIST
Bldg. 222 Room B264
Gaithersburg, MD 20899
US
mél : sheila.frankel@nist.gov

Déclaration complète de droits de reproduction

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