Groupe de travail Réseau
Request for Comments : 3537
Catégorie : En cours de normamisation
Traduction Claude Brière de L'Isle

J. Schaad, Soaring Hawk Consulting
R. Housley, Vigil Security

mai 2003

Enveloppement d'une clé de code d'authentification de message haché (HMAC) dans une clé de la norme de triple chiffrement de données (3DES) ou de la norme de chiffrement évolué (AES)

Résumé
Le présent document définit deux méthodes pour envelopper une clé de code d'authentification de message haché (HMAC, *Hashed Message Authentication Code*). La première méthode définie utilise une clé Triple DES *(Data Encryption Standard)* pour chiffrer la clé HMAC. La seconde méthode définie utilise une clé à la norme de chiffrement évolué (AES, *Advanced Encryption Standard*) pour chiffrer la clé HMAC. Cet algorithme est utilisé pour le type de données authentifiées dans la syntaxe de message de chiffrement (CMS, *Cryptographic Message Syntax*).

1. Introduction

Il existe des méthodes standard pour chiffrer une clé de chiffrement de contenu (CEK, *content-encryption key*) Triple-DES (3DES) avec une clé de chiffrement de clé (KEK, *key-encryption key*) 3DES [RFC3217], et pour chiffrer une CEK AES avec une KEK AES [RFC3394]. L'enveloppement de la clé Triple-DES impose des restrictions de parité, et dans les deux instances il y a des restrictions sur la taille de la clé enveloppée qui rendent difficile le chiffrement du matériel de chiffrement HMAC [RFC2104].

Le présent document spécifie un mécanisme pour le chiffrement d'une clé HMAC de longueur arbitraire par une KEK 3DES ou une KEK AES.

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

2. Lignes directrices pour la clé HMAC

La [RFC2104] suggère que la clé soit au moins aussi longue que le résultat (L) de la fonction de hachage utilisé. Lorsque des clés plus longues que la taille de bloc de l'algorithme de hachage sont utilisées, elles sont hachées et la valeur de hachage résultante est utilisée. Utiliser des clés beaucoup plus longues que L n'apporte aucun avantage pour la sécurité sauf si la fonction aléatoire utilisée pour créer la clé a un faible résultat d'entropie.

3. Enveloppement et désenveloppement de clé HMAC avec Triple-DES

Cette section spécifie les algorithmes pour envelopper et désenvelopper une clé HMAC avec une KEK 3DES [3DES].

L'enveloppement 3DES de clés HMAC se fonde sur l'algorithme défini à la Section 3 de la [RFC3217]. Les différences majeures sont dues au fait qu'une clé HMAC est de longueur variable et que la clé HMAC n'a pas de parité particulière.

Dans la description de l'algorithme, "a || b" est utilisé pour représenter 'a' enchaîné avec 'b'.

3.1 Enveloppement d'une clé HMAC avec une clé de chiffrement de clé Triple-DES

Cet algorithme chiffre une clé HMAC avec une KEK 3DES. L'algorithme est :

1. On appelle KEY la clé HMAC. Soit LONG la longueur de KEY en octets. LONG est d'un seul octet.

2. Soit LKEY = LONG || KEY.

3. Soit LKEYPAD = LKEY || PAD. Si la longueur de LKEY est un multiple de 8, PAD a une longueur de zéro. Si la longueur de LKEY n'est pas un multiple de 8, alors PAD contient le plus petit nombre d'octets aléatoires pour faire de la longueur de LKEYPAD un multiple de 8.

4. Calculer une valeur de somme de contrôle de clé de 8 octets sur LKEYPAD comme décrit à la Section 2 de la [RFC3217], on appelle le résultat ICV.

5. Soit LKEYPADICV = LKEYPAD || ICV.

6. Générer 8 octets au hasard, on appelle le résultat IV.

7. Chiffrer LKEYPADICV en mode CBC en utilisant la KEK 3DES. Utiliser la valeur aléatoire générée à l'étape précédente comme vecteur d'initialisation (IV). On appelle le texte chiffré TEMP1.

8. Soit TEMP2 = IV || TEMP1.

9. Inverser l'ordre des octets dans TEMP2. C'est-à-dire, l'octet de poids fort (le premier) est échangé avec l'octet de moindre poids (le dernier), et ainsi de suite. On appelle le résultat TEMP3.

10. Chiffrer TEMP3 en mode CBC en utilisant la KEK 3DES. Utiliser un vecteur d'initialisation (IV) de 0x4adda22c79e82105.

Note : Lorsque la même clé HMAC est enveloppée dans des KEK 3DES différentes, un vecteur d'initialisation (IV) frais doit être généré pour chaque invocation de l'algorithme d'enveloppe de clé HMAC(étape 6).

3.2 Désenveloppement d'une clé HMAC avec une clé de chiffrement de clé Triple-DES

Cet algorithme déchiffre une clé HMAC en utilisant une KEK 3DES. L'algorithme est :

1. Si la clé enveloppée n'est pas un multiple de 8 octets, il y a erreur.

2. Déchiffrer la clé enveloppée en mode CBC en utilisant la KEK 3DES. Utiliser un vecteur d'initialisation (IV) de 0x4adda22c79e82105. On appelle le résultat TEMP3.

3. Inverser l'ordre des octets dans TEMP3. C'est-à-dire, l'octet de poids fort (le premier) est échangé avec l'octet de moindre poids (le dernier) et ainsi de suite. On appelle le résultat TEMP2.

4. Décomposer le TEMP2 en IV et TEMP1. IV est les 8 octets de poids fort (les premiers) et TEMP1 est composé des octets restants.

5. Déchiffrer TEMP1 en mode CBC en utilisant la KEK 3DES. Utiliser la valeur d'IV de l'étape précédente comme vecteur d'initialisation. On appelle le texte en clair LKEYPADICV.

6. Décomposer le LKEYPADICV en LKEYPAD, et ICV. ICV est les 8 octets de moindre poids (les derniers). LKEYPAD est composé des octets restants.

7. Calculer une valeur de somme de contrôle de clé de 8 octets sur LKEYPAD comme décrit à la Section 2 de la [RFC3217]. Si la valeur de la somme de contrôle de la clé calculée ne correspond pas à la valeur de la somme de contrôle de la clé déchiffrée, ICV, il y a erreur.

8. Décomposer le LKEYPAD en LONG, KEY, et PAD. LONG est l'octet de poids fort (le premier). KEY est la valeur de LONG d'octets suivants. PAD est les octets restants, si il en est.

9. Si la longueur de PAD fait plus de 7 octets, il y a erreur.
10. Utiliser KEY comme clé HMAC.

3.3 Enveloppement de clé HMAC avec identifiant d'algorithme Triple-DES

Certains protocoles de sécurité emploient l'ASN.1 [X.208-88], [X.209-88], et ces protocoles emploient des identifiants d'algorithme pour désigner les algorithmes cryptographiques. Pour prendre en charge ces protocoles, l'enveloppe de clé HMAC avec algorithme Triple-DES a reçu les identifiants d'algorithme suivants :

IDENTIFIANT D'OBJET id-alg-HMACwith3DESwrap ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) alg(3) 11 }

Le champ de paramètre AlgorithmIdentifier DOIT être NUL.

3.4 Enveloppement de clé HMAC avec vecteur d'essai Triple-DES

KEK : 5840df6e 29b02af1 ab493b70 5bf16ea1 ae8338f4 dcc176a8

HMAC_KEY : c37b7e64 92584340 bed12207 80894115 5068f738

IV : 050d8c79 e0d56b75

PAD : be62fe

ICV : 1f363a31 cdaa9037

LKEYPADICV : 14c37b7e 64925843 40bed122 07808941 155068f7 38be62fe 1f363a31 cdaa9037

TEMP1 : 157a8210 f432836b a618b096 475c864b 6612969c dfa445b1 5646bd00 500b2cc1

TEMP3 : c12c0b50 00bd4656 b145a4df 9c961266 4b865c47 96b018a6 6b8332f4 10827a15 756bd5e0 798c0d05

Clé enveloppée : 0f1d715d 75a0aaf6 6f02e371 c08b79e2 a1253dc4 3040136b dc161118 601f2863 e2929b3b dd17697c

4. Enveloppement et développement de clé HMAC avec AES

Cette section spécifie les algorithmes pour envelopper et désenvelopper une clé HMAC avec une KEK AES [RFC3394].

L'enveloppement AES des clés HMAC se fonde sur l'algorithme défini dans la [RFC3394]. La différence majeure est l'inclusion de bourrage du fait que la longueur d'une clé HMAC peut n'être pas un multiple de 64 bits.

Dans la description de l'algorithme, "a || b" est utilisé pour représenter 'a' enchaîné avec 'b'.

4.1 Enveloppement de clé HMAC avec une clé de chiffrement de clé AES

Cet algorithme chiffre une clé HMAC avec une KEK AES. L'algorithme est :

1. Appelons KEY la clé HMAC, et appelons LONG la longueur de KEY en octets. LONG fait un seul octet.

2. Soit LKEY = LONG || KEY.

3. Soit LKEYPAD = LKEY || PAD. Si la longueur de LKEY est un multiple de 8, le PAD a une longueur de zéro. Si la longueur de LKEY n'est pas un multiple de 8, PAD contient alors le plus petit nombre d'octets aléatoires pour faire de la longueur de LKEYPAD un multiple de 8.

4. Chiffrer LKEYPAD en utilisant l'algorithme d'enveloppe de clé AES spécifié au paragraphe 2.2.1 de la [RFC3394], en utilisant la KEK AES comme clé de chiffrement. Le résultat est plus long de 8 octets que LKEYPAD.

4.2 Désenveloppement d'une clé HMAC avec une clé AES

L'algorithme de désenveloppement de clé AES déchiffre une clé HMAC en utilisant une KEK AES. L'algorithme de désenveloppement de clé AES est :

1. Si la clé enveloppée n'est pas un multiple de 8 octets, il y a une erreur.

2. Déchiffrer la clé enveloppée en utilisant l'algorithme de désenveloppement de clé AES spécifié au paragraphe 2.2.2 de la [RFC3394], en utilisant la KEK AES comme clé de déchiffrement. Si la vérification d'intégrité interne de l'algorithme de désenveloppement échoue, il y a une erreur, sinon, on, appelle le résultat LKEYPAD.

3. Décomposer LKEYPAD en LONG, KEY, et PAD. LONG est l'octet de poids fort (le premier). KEY est la valeur de LONG d'octets suivants. PAD est les octets restants, si il en est.

4. Si la longueur de PAD fait plus de 7 octets, il y a erreur.

5. Utiliser KEY comme clé HMAC.

4.3 Enveloppement de clé HMAC avec identifiant d'algorithme AES

Certains protocoles de sécurité emploient l'ASN.1 [X.208-88], [X.209-88], et ces protocoles emploient des identifiants d'algorithme pour désigner les algorithmes cryptographiques. Pour prendre en charge ces protocoles, il a été alloué à l'enveloppement de clé HMAC avec algorithme AES l'identifiant de algorithme suivant :
IDENTIFIANT D'OBJET id-alg-HMACwithAESwrap ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) alg(3) 12 }

Le champ de paramètre AlgorithmIdentifier DOIT être NUL.

4.4 Enveloppement de clé HMAC avec vecteur d'essai AES

KEK : 5840df6e 29b02af1 ab493b70 5bf16ea1 ae8338f4 dcc176a8

HMAC_KEY : c37b7e64 92584340 bed12207 80894115 5068f738

PAD : 050d8c

LKEYPAD : 14c37b7e 64925843 40bed122 07808941 155068f7 38050d8c

Clé enveloppée : 9fa0c146 5291ea6d b55360c6 cb95123c d47b38cc e84dd804 fbcec5e3 75c3cb13

5. Considérations sur la sécurité

Les mises en œuvre doivent protéger la clé de chiffrement de clé (KEK). La compromission de la KEK peut résulter en la divulgation de toutes les clés HMAC qui ont été enveloppées avec la KEK, ce qui peut conduire à la perte de la protection de l'intégrité des données.

L'utilisation de ces fonctions d'enveloppement de clé assure la protection de la confidentialité et de l'intégrité des données, mais elle ne fournit pas nécessairement l'authentification de l'origine des données. Tout possesseur de la KEK peut créer un message qui réussit aux vérifications d'intégrité. Si on désire aussi l'authentification de l'origine des données, le mécanisme de distribution de la KEK doit fournir l'authentification de l'origine des données de la KEK. Autrement, une signature numérique peut être utilisée.

Les mises en œuvre doivent générer les vecteurs d'initialisation (les IV) et le bourrage de façon aléatoire. La génération de nombres aléatoires de qualité est difficile.

La [RFC1750] offre d'importantes lignes directrices dans ce domaine, et l'Appendice 3 de FIPS Pub 186 [DSS] fournit une technique de PRNG de qualité.

L'algorithme d'enveloppement de clé spécifié dans le présent document a été revisité pour l'utilisation avec Triple-DES et AES, et n'a pas été revu pour l'utiliser avec d'autres algorithmes de chiffrement.

6. Références

6.1 Références normatives

[3DES] American National Standards Institute. ANSI X9.52-1998, "Triple Data Encryption Algorithm Modes of Operation". 1998.

[AES] National Institute of Standards and Technology. FIPS Pub 197, "Advanced Encryption Standard (AES). 26 novembre 2001.

[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.

[RFC3217] R. Housley, "Enveloppe de clé en Triple-DES et RC2", décembre 2001. *(Information)*

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

6.2 Références pour information

[DSS] National Institute of Standards and Technology. FIPS Pub 186, "Digital Signature Standard". 19 mai 1994.

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

[RFC2026] S. Bradner, "Le processus de normalisation de l'Internet -- Révision 3", (BCP0009) octobre 1996. *(Remplace* RFC1602, RFC1871) (*MàJ par* RFC3667, RFC3668, RFC3932, RFC3979, RFC3978, RFC5378, RFC6410)

[X.208-88] Recommandation UIT-T X.208, "Specification of Abstract Syntax Notation One (ASN.1)". 1988.

[X.209-88] Recommandation UIT-T X.209, "Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)". 1988.

7. Adresses des auteurs

Jim Schaad
Soaring Hawk Consulting
mél : jimsch@exmsft.com

Russell Housley
Vigil Security
918 Spring Knoll Drive
Herndon, VA 20170 USA
mél : housley@vigilsec.com

8. Déclaration complète de droits de reproduction

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