Groupe de travail Réseau
Request for Comments : 3217
Catégorie : Information

R. Housley, RSA Laboratories
décembre 2001
Traduction Claude Brière de L'Isle

Enveloppe de clé Triple-DES et RC2

Statut de ce mémoire
Le présent mémoire apporte des informations pour la communauté de l'Internet. Il ne spécifie aucune sorte de norme de l'Internet. La distribution du présent mémoire n'est soumise à aucune restriction.
(La présente traduction incorpore l'errata 639 du 28/10/2007)

Résumé
Le présent document spécifie l'algorithme pour envelopper une clé Triple-DES avec une autre clé Triple-DES et l'algorithme pour envelopper une clé RC2 avec une autre clé RC2. Ces algorithmes d'enveloppe de clé ont été à l'origine publiés au paragraphe 12.6 de la RFC2630. Ils sont republiés car ces algorithmes d'enveloppe de clé se sont trouvés être utiles dans des contextes qui vont au delà de ceux pris en charge par la RFC2630.

1. Introduction

La gestion des clés de chiffrement symétriques conduit souvent à des situations où une clé symétrique est utilisée pour en chiffrer (ou envelopper) une autre. Les algorithmes d'enveloppe de clé sont couramment utilisés dans deux situations. D'abord, les algorithmes d'accord de clé (comme Diffie-Hellman [RFC2631]) génèrent une paire de clés de chiffrement de clé, et un algorithme d'enveloppe de clé est utilisé pour chiffrer la clé de chiffrement du contenu ou une clé en diffusion groupée avec la paire de clés de chiffrement de clé. Ensuite, un algorithme d'enveloppe de clé est utilisé pour chiffrer la clé de chiffrement de contenu, la clé en diffusion groupée, ou une clé de session dans une mémorisation de clé de chiffrement de clé générée localement ou une clé de chiffrement de clé qui a été distribuée hors bande.

Le présent document spécifie l'algorithme pour envelopper une clé Triple-DES avec une autre clé Triple-DES [3DES], et il spécifie l'algorithme pour envelopper une clé RC2 avec une autre clé RC2 [RFC2268]. Le chiffrement d'une clé Triple-DES avec une autre clé Triple-DES utilise l'algorithme spécifié à la section 3. Le chiffrement d'une clé RC2 avec une autre clé RC2 utilise l'algorithme spécifié à la section 4. Ces deux algorithmes s'appuient sur l'algorithme de somme de contrôle de clé spécifié à la section 2. Les clés de chiffrement de contenu Triple-DES et RC2 sont chiffrées en mode de chaînage de bloc de chiffrement (CBC, *Cipher Block Chaining*) [MODES].

Dans le présent document, les mots clés DOIT, NE DOIT PAS, EXIGÉ, DEVRAIT, NE DEVRAIT PAS, RECOMMANDE, et PEUT sont à interpréter comme décrit par Scott Bradner dans la [RFC2119].

2. Somme de contrôle de clé

L'algorithme de somme de contrôle de clé est utilisé pour fournir une valeur de vérification d'intégrité de clé. L'algorithme est :

1. Calculer un résumé de message SHA-1 [SHA1] de 20 octets sur la clé qui est à envelopper.
2. Utiliser les huit octets de poids fort (les premiers) de la valeur de résumé de message comme valeur de somme de contrôle.

3. Enveloppement et développement de clé Triple-DES

Cette section spécifie les algorithmes d'enveloppement et de développement d'une clé Triple-DES avec une autre clé Triple-DES [3DES].

Le même algorithme d'enveloppe de clé est utilisé pour les deux clés de Triple-DES à trois clés et Triple-DES à deux clés. Lorsque une clé Triple-DES à deux clés doit être enveloppée, une troisième clé DES avec la même valeur que la première clé DES est créée. Donc, toutes les clés Triple-DES enveloppées comportent trois clés DES. Cependant, une clé Triple-

DES à deux clés NE DOIT PAS être utilisée pour envelopper une clé Triple-DES à trois clés qui se compose de trois clés DES uniques.

3.1 Enveloppement de clé Triple-DES

L'algorithme d'enveloppe de clé Triple-DES chiffre une clé Triple-DES avec une clé de chiffrement de clé Triple-DES. L'algorithme d'enveloppe de clé Triple-DES est :

1. Établir l'imparité pour chacun des octets de la clé DES composant la clé Triple-DES à trois clé qui est à envelopper ; on appelle CEK le résultat.
2. Calculer une valeur de somme de contrôle de clé de 8 octets sur CEK comme décrit ci-dessus à la Section 2 ; on appelle ICV le résultat.
3. Soit CEKICV = CEK || ICV.
4. Générer 8 octets au hasard, on appelle IV le résultat.
5. Chiffrer CEKICV en mode CBC en utilisant la clé de chiffrement de clé. Utiliser la valeur générée au hasard à l'étape précédente comme vecteur d'initialisation (IV). On appelle TEMP1 le texte chiffré.
6. Soit TEMP2 = IV || TEMP1.
7. Inverser l'ordre des octets dans TEMP2. C'est-à-dire que l'octet de poids fort (le premier) est échangé avec l'octet de moindre poids (le dernier) et ainsi de suite. On appelle TEMP3 le résultat.
8. Chiffrer TEMP3 en mode CBC en utilisant la clé de chiffrement de clé. Utiliser un vecteur d'initialisation (IV) de 0x4adda22c79e82105. Le texte chiffré est long de 40 octets.

Note : Lorsque la même clé Triple-DES à trois clés est enveloppée dans différentes clés de chiffrement de clé, un vecteur d'initialisation (IV) frais doit être généré pour chaque invocation de l'algorithme d'enveloppe de clé.

3.2 Développement de clé Triple-DES

L'algorithme de développement de clé Triple-DES déchiffre une clé Triple-DES en utilisant une clé de chiffrement de clé Triple-DES. L'algorithme de développement de clé Triple-DES est :

1. Si la clé enveloppée ne fait pas 40 octets, il y a erreur.
2. Déchiffrer la clé enveloppée en mode CBC en utilisant la clé de chiffrement de clé. Utiliser un vecteur d'initialisation (IV) de 0x4adda22c79e82105. On appelle TEMP3 le résultat.
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 TEMP2 le résultat.
4. Décomposer TEMP2 en IV et TEMP1. IV est composé des huit octets de poids fort (les premiers) et TEMP1 est composé des 32 octets de moindre poids (les derniers).
5. Déchiffrer TEMP1 en mode CBC en utilisant la clé de chiffrement de clé. Utiliser la valeur de IV de l'étape précédente comme vecteur d'initialisation. On appelle CEKICV le texte chiffré.
6. Décomposer CEKICV en CEK et ICV. CEK est les 24 octets de poids fort (les premiers) et ICV est les huit octets de moindre poids (les derniers).
7. Calculer une valeur de somme de contrôle de clé de 8 octets sur CEK comme décrit ci-dessus à la Section 2. Si la valeur calculée de somme de contrôle de clé ne correspond pas à la valeur de la somme de contrôle déchiffrée, ICV, il y a erreur.
8. Vérifier l'imparité de chaque octet de la clé DES composant CEK. Si la parité est incorrecte, il y a erreur.
9. Utiliser CEK comme clé Triple-DES.

3.3 Identifiant d'algorithme de clé 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'algorithme d'enveloppe de clé Triple-DES a reçu l'identifiant d'algorithme suivant :

IDENTIFIANT D'OBJET id-alg-CMS3DESwrap ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) alg(3) 6 }

Le champ de paramètre AlgorithmIdentifier DOIT être NUL.

3.4 Exemple d'enveloppe de clé Triple-DES

Ce paragraphe contient un exemple d'enveloppe de clé Triple-DES. Les valeurs intermédiaires qui correspondent aux éléments désignés au paragraphe 3.1 sont données en hexadécimal.

CEK : 2923 bf85 e06d d6ae 5291 49f1 f1ba e9ea b3a7 da3d 860d 3e98
KEK : 255e 0d1c 07b6 46df b313 4cc8 43ba 8aa7 1f02 5b7c 0838 251f
ICV : 181b 7e96 86e0 4a4e
CEKICV : 2923 bf85 e06d d6ae 5291 49f1 f1ba e9ea b3a7 da3d 860d 3e98 181b 7e96 86e0 4a4e
IV : 5dd4 cbfc 96f5 453b
TEMP1 : cfc1 a789 c675 dd2a b49a 3204 ef92 cc03 5c1f 973b 7a79 60f6 a44d cc5f 729d 8449
TEMP2 : 5dd4 cbfc 96f5 453b cfc1 a789 c675 dd2a b49a 3204 ef92 cc03 5c1f 973b 7a79 60f6 a44d cc5f 729d 8449
TEMP3 : 4984 9d72 5fcc 4da4 f660 797a 3b97 1f5c 03cc 92ef 0432 9ab4 2add 75c6 89a7 c1cf 3b45 f596 fccb d45d
RESULT : 6901 0761 8ef0 92b3 b48c a179 6b23 4ae9 fa33 ebb4 1596 0403 7db5 d6a8 4eb3 aac2 768c 6327 75a4 67d4

4. Enveloppement et développement de clé RC2

Cette section spécifie les algorithmes pour envelopper et développer une clé RC2 avec une autre clé RC2 [RFC2268].

RC2 accepte des longueurs de clé variables. Les clés RC2 à 128 bits DOIVENT être utilisées comme clé de chiffrement de clé ; cependant, la clé RC2 enveloppée PEUT être de n'importe quelle taille.

4.1 Enveloppement de clé RC2

L'algorithme d'enveloppe de clé RC2 chiffre une clé RC2 avec une clé de chiffrement de clé RC2. L'algorithme d'enveloppe de clé RC2 est :

1. Soit CEK la clé RC2, et soit LENGTH la longueur de CEK en octets. LENGTH est un seul octet.
2. Soit LCEK = LENGTH || CEK.
3. Soit LCEKPAD = LCEK || PAD. Si la longueur de LCEK est un multiple de 8, la longueur de PAD est zéro. Si la longueur de LCEK n'est pas un multiple de 8, alors PAD contient le plus petit nombre d'octets aléatoires pour faire de la longueur de LCEKPAD un multiple de 8.
4. Calculer une valeur de somme de contrôle de clé de 8 octets sur LCEKPAD comme décrit à la Section 2 ; on appelle ICV le résultat.
5. Soit LCEKPADICV = LCEKPAD || ICV.
6. Générer 8 octets au hasard ; on appelle IV le résultat.
7. Chiffrer LCEKPADICV en mode CBC en utilisant la clé de chiffrement de clé. Utiliser la valeur aléatoire générée à l'étape précédente comme vecteur d'initialisation (IV). On appelle TEMP1 le texte chiffré.
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 TEMP3 le résultat.
10. Chiffrer TEMP3 en mode CBC en utilisant la clé de chiffrement de clé. Utiliser un vecteur d'initialisation (IV) de 0x4adda22c79e82105.

Note : Lorsque la même clé RC2 est enveloppée dans des clés de chiffrement de clé différentes, un nouveau vecteur d'initialisation (IV) doit être généré pour chaque invocation de l'algorithme d'enveloppe de clé.

4.2. Développement de clé RC2

L'algorithme de développement de clé RC2 déchiffre une clé RC2 en utilisant une clé de chiffrement de clé RC2. L'algorithme de développement de clé RC2 est le suivant :

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 clé de chiffrement de clé. Utiliser un vecteur d'initialisation (IV) de 0x4adda22c79e82105. On appelle TEMP3 le résultat.
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 TEMP2 le résultat.
4. Décomposer TEMP2 en IV et TEMP1. IV est les huit octets de poids fort (les premiers) et TEMP1 est le reste.
5. Déchiffrer TEMP1 en mode CBC en utilisant la clé de chiffrement de clé. Utiliser la valeur de IV trouvée à l'étape précédente comme vecteur d'initialisation. On appelle LCEKPADICV le texte en clair.

6. Décomposer LCEKPADICV en LCEKPAD et ICV. ICV est les huit octets de moindre poids (les derniers). LCEKPAD est le reste.
7. Calculer une somme de contrôle de clé de 8 octets sur LCEKPAD comme décrit à la Section 2. Si la valeur de somme de contrôle de clé calculée ne correspond pas à la valeur de somme de contrôle de clé déchiffrée, ICV, il y a erreur.
8. Décomposer LCEKPAD en LENGTH, CEK, et PAD. LENGTH est l'octet de poids fort (le premier). CEK est les LENGTH octets qui suivent. PAD est les octets restants, s'il en est.
9. Si la longueur de PAD est supérieure à 7 octets, il y a erreur.
10. Utiliser CEK comme clé RC2.

4.3 Identifiant d'algorithme d'enveloppe de clé RC2

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'algorithme d'enveloppe de clé RC2 les identifiants d'algorithme suivants :

IDENTIFIANT D'OBJET id-alg-CMSRC2wrap ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) alg(3) 7 }

Le champ de paramètre AlgorithmIdentifier DOIT être RC2wrapParameter :

RC2wrapParameter ::= RC2ParameterVersion

RC2ParameterVersion ::= ENTIER

Les bits de clé efficaces RC2 (la taille de clé) supérieurs à 32 et inférieurs à 256 sont codés en RC2ParameterVersion. Pour les bits de clé efficaces de 40, 64, et 128, les valeurs de rc2ParameterVersion sont respectivement 160, 120, et 58. Ces valeurs ne sont pas simplement la longueur de clé RC2. Noter que la valeur 160 doit être codée par deux octets (00 A0), parce que le codage sur un octet (A0) représente un nombre négatif.

4.4 Exemple d'enveloppe de clé RC2

Ce paragraphe contient un exemple d'enveloppe de clé RC2. Les valeurs intermédiaires qui correspondent aux éléments désignés au paragraphe 4.1 sont données en hexadécimal. Dans cet exemple, le paramètre de longueur de clé efficace pour l'algorithme RC2 devrait être de 40 bits.

Bits de clé RC2 efficaces : 40

CEK est (16 octets) : b7 0a 25 fb c9 d8 6a 86 05 0c e0 d7 11 ea d4 d9

LENGTH est : 16

LCEK est (17 octets) : 10 b7 0a 25 fb c9 d8 6a 86 05 0c e0 d7 11 ea d4 d9

PAD est (7 octets) : 48 45 cc e7 fd 12 50

LCEKPAD est (24 octets) : 10 b7 0a 25 fb c9 d8 6a 86 05 0c e0 d7 11 ea d4 d9 48 45 cc e7 fd 12 50

SHA-1 Digest est (20 octets) : 0a 6f f1 9f db 40 49 88 a2 fa ee 2e 53 37 12 98 7e ca 48 06

ICV est (8 octets) : 0a 6f f1 9f db 40 49 88

LCEKPADICV est (32 octets) : 10 b7 0a 25 fb c9 d8 6a 86 05 0c e0 d7 11 ea d4 d9 48 45 cc e7 fd 12 50 0a 6f f1 9f db 40 49 88

IV est (8 octets) : c7 d9 00 59 b2 9e 97 f7

KEK (16 octets) : fd 04 fd 08 06 07 07 fb 00 03 fe ff fd 02 fe 05

TEMP1 (32 octets) : a0 1d a2 59 37 93 12 60 e4 8c 55 f5 04 ce 70 b8 ac 8c d7 9e ff 8e 99 32 9f a9 8a 07 a3 1f f7 a7

TEMP2 (40 octets) : c7 d9 00 59 b2 9e 97 f7 a0 1d a2 59 37 93 12 60 e4 8c 55 f5 04 ce 70 b8 ac 8c d7 9e ff 8e 99 32 9f a9 8a 07 a3 1f f7 a7

TEMP3 (40 octets) : a7 f7 1f a3 07 8a a9 9f 32 99 8e ff 9e d7 8c ac b8 70 ce 04 f5 55 8c e4 60 12 93 37 59 a2 1d a0 f7 97 9e b2 59 00 d9 c7

FinalIV (8 octets) : 4a dd a2 2c 79 e8 21 05

KEK (16 octets) : fd 04 fd 08 06 07 07 fb 00 03 fe ff fd 02 fe 05

RESULT (40 octets) : 70 e6 99 fb 57 01 f7 83 33 30 fb 71 e8 7c 85 a4 20 bd c9 9a f0 5d 22 af 5a 0e 48 d3 5f 31 38 98 6c ba af b4 b2 8d 4f 35

==========

Bits de clé RC2 efficaces : 128

CEK est (16 octets) : b7 0a 25 fb c9 d8 6a 86 05 0c e0 d7 11 ea d4 d9

LENGTH est : 16

LCEK est (17 octets) : 10 b7 0a 25 fb c9 d8 6a 86 05 0c e0 d7 11 ea d4 d9

PAD est (7 octets) : 48 45 cc e7 fd 12 50

LCEKPAD est (24 octets) : 10 b7 0a 25 fb c9 d8 6a 86 05 0c e0 d7 11 ea d4 d9 48 45 cc e7 fd 12 50

SHA-1 Digest est (20 octets) : 0a 6f f1 9f db 40 49 88 a2 fa ee 2e 53 37 12 98 7e ca 48 06

ICV est (8 octets) : 0a 6f f1 9f db 40 49 88

LCEKPADICV est (32 octets) : 10 b7 0a 25 fb c9 d8 6a 86 05 0c e0 d7 11 ea d4 d9 48 45 cc e7 fd 12 50 0a 6f f1 9f db 40 49 88

IV est (8 octets) : c7 d9 00 59 b2 9e 97 f7

KEK (16 octets) : fd 04 fd 08 06 07 07 fb 00 03 fe ff fd 02 fe 05

TEMP1 (32 octets) : 03 5e 97 2a b1 5c c4 c9 c4 a0 3d ba a3 5a 21 66 67 e4 3e bc a2 67 46 ae 86 08 db c8 9e 64 ca 29

TEMP2 (40 octets) : c7 d9 00 59 b2 9e 97 f7 03 5e 97 2a b1 5c c4 c9 c4 a0 3d ba a3 5a 21 66 67 e4 3e bc a2 67 46 ae 86 08 db c8 9e 64 ca 29

TEMP3 (40 octets) : 29 ca 64 9e c8 db 08 86 ae 46 67 a2 bc 3e e4 67 66 21 5a a3 ba 3d a0 c4 c9 c4 5c b1 2a 97 5e 03 f7 97 9e b2 59 00 d9 c7

FinalIV (8 octets) : 4a dd a2 2c 79 e8 21 05

KEK (16 octets) : fd 04 fd 08 06 07 07 fb 00 03 fe ff fd 02 fe 05

RESULT (40 octets) : f4 d8 02 1c 1e a4 63 d2 17 a9 eb 69 29 ff a5 77 36 d3 e2 03 86 c9 09 93 83 5b 4b e4 ad 8d 8a 1b c6 3b 25 de 2b f7 79 93

5. Références

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

[DES] American National Standards Institute. ANSI X3.106, "Information Systems - Data Link Encryption". 1983.

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

[MODES] National Institute of Standards et Technology. FIPS Pub 81:" DES Modes of Operation". 2 décembre 1980.

[RFC1750] D. Eastlake, 3rd et autres, "Recommandations d'aléa pour la sécurité", décembre 1994. *(Info., remplacée par la RFC 4086)*

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

[RFC2268] R. Rivest, "Description de l'algorithme de chiffrement RC2(r)", mars 1998. *(Information)*

[RFC2630] R. Housley, "Syntaxe de message cryptographique", juin 1999. *(Obsolète, voir 3369, 3370) (P.S.)*

[RFC2631] E. Rescorla, "Méthode d'accord de clé Diffie-Hellman", juin 1999. *(P.S.)*

[SHA1] National Institute of Standards et Technology. FIPS Pub 180-1: "Secure Hash Standard". 17 avril 1995.

[X.208-88] CCITT. Recommandation X.208: "Spécification de la notation de la syntaxe abstraite n° 1 (ASN.1)". 1988.

[X.209-88] CCITT. Recommandation X.209: "Spécification des règles de codage de base pour la notation de la syntaxe abstraite n° 1 (ASN.1)". 1988.

6. Considérations pour la sécurité

Les mestes en œuvre doivent protéger la clé de chiffrement de clé. La compromestsion de la clé de chiffrement de clé peut résulter en la divulgation de toutes les clés qui ont été enveloppées avec la clé de chiffrement de clé, ce qui peut conduire à la divulgation de tout le trafic protégé avec ces clé enveloppées.

Les mestes en œuvre doivent générer au hasard les vecteurs d'initialestation (IV) et le bourrage. 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

Si la clé de chiffrement de clé et la clé enveloppée sont associées à différents algorithmes de chiffrement symétriques, la sécurité effective fournie aux données chiffrées avec la clé enveloppée est déterminée par le plus faible des deux algorithmes. Si, par exemple, des données sont chiffrées avec une clé Triple-DES de 168 bits et si cette clé Triple-DES est enveloppée par une clé RC2 de 40 bits, la protection fournie est au plus de 40 bits. Une recherche triviale pour déterminer la valeur de la clé RC2 de 40 bits peut retrouver la clé Triple-DES, et ensuite la clé Triple-DES peut être utilestée pour déchiffrer le contenu. Les mestes en œuvre doivent donc s'assurer que les algorithmes de chiffrement de clé sont aussi forts ou plus forts que les algorithmes de chiffrement de contenu.

Les algorithmes d'enveloppe de clé spécifiés dans le présent document ont été revus pour une utilestation avec Triple-DES et RC2, et n'ont pas été revus pour une utilestation avec d'autres algorithmes de chiffrement. De même, les algorithmes d'enveloppe de clé font usage du mode CBC [MODES], et ils n'ont pas été revus dans la perspective d'une utilestation avec d'autres modes cryptographiques.

7. Remerciements

Le présent document est le résultat de contributions de nombreux professionnels. Je remercie de leur dur labeur tous les membre du groupe de travail S/MIME de l'IETF. Je doest des remerciements particuliers à Carl Elleston, Peter Gutmann, Bob Jueneman, Don Johnson, Burt Kalestki, John Pawling, et Jim Schaad pour leur soutient à la définition de ces algorithmes et la préparation de cette spécification.

8 Adresse de l'auteur

Russell Housley
RSA Laboratories
918 Spring Knoll Drive
Herndon, VA 20170
USA
mél : rhousley@rsasecurity.com

9. Déclaration de droits de reproduction

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