Groupe de travail Réseau
Request for Comments : 3565
Catégorie : En cours de normalisation

J. Schaad, Soaring Hawk Consulting
juillet 2003
Traduction Claude Brière de L'Isle

Utilisation de l'algorithme de chiffrement de la norme de chiffrement évoluée (AES) dans la syntaxe de message cryptographique (CMS)

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

Résumé
Le présent document spécifie les conventions pour l'utilisation de l'algorithme de la norme de chiffrement évoluée (AES, *Advanced Encryption Standard*) pour le chiffrement avec la syntaxe de message cryptographique (CMS, *Cryptographic Message Syntax*).

Conventions utilisées dans le document
Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "PEUT", et "FACULTATIF" dans le présent document sont à interpréter comme décrit dans la [RFC2119].

1. Généralités

Le présent document spécifie les conventions pour l'utilisation de l'algorithme de chiffrement de contenu de la norme de chiffrement évoluée (AES, *Advanced Encryption Standard*) avec les types de contenu Données enveloppées et Données chiffrées de la syntaxe de message cryptographique [RFC3369].

Les valeurs de CMS sont générées en utilisant l'ASN.1 [X.208-88], les règles de codage de base (BER, *Basic Encoding Rules*) [X.209-88] et les règles de codage distinctif (DER, *Distinguished Encoding Rules*) [X.509-88].

1.1 AES

La norme de chiffrement évoluée [AES] a été développée pour remplacer [DES]. La publication de la norme fédérale de traitement de l'information (FIPS, *Federal Information Processing Standard*) AES spécifie un algorithme de chiffrement à l'usage des organisations gouvernementales des USA. Cependant, AES sera aussi largement utilisée par des organisations, institutions, et individus en dehors du gouvernement américain.

Les deux chercheurs qui ont développé et publié l'algorithme de Rijndael pour qu'il soit pris en considération sont tous deux des cryptographes belges : Dr. Joan Daemen, de Proton World International, et Dr. Vincent Rijmen, chercheur postdoctorant du département d'ingénierie électrique de l'Université catholique de Louvain.

L'Institut National des normes et technologie (NIST) a choisi l'algorithme de Rijndael pour AES parce qu'il offre une combinaison de sécurité, performances, efficacité, facilité de mise en œuvre, et souplesse. En particulier, Rijndael paraît avoir avec constance de très bonnes performance, tant dans le matériel que le logiciel, dans une large gamme d'environnements informatiques, sans considération de son utilisation en mode avec retour ou sans retour. Son temps d'établissement de clés est excellent, et son agilité de clés est bonne. Les très faibles exigences de mémoire de l'algorithme de Rijndael font qu'il est très bien adapté pour les environnements en espace restreint, dans lesquels il montre aussi d'excellentes performances. Les opérations de l'algorithme de Rijndael sont parmi les plus faciles à défendre contre les attaques en puissance et en temps. De plus, il apparaît que des défenses peuvent être fournies contre de telles attaques sans impacter de façon significative les performances de l'algorithme. Finalement, la structure interne circulaire de l'algorithme paraît avoir un bon potentiel pour bénéficier du parallélisme au niveau instruction.

AES spécifie trois tailles de clés : 128, 192 et 256 bits.

2. Conventions de données enveloppées

Le type de contenu CMS Données enveloppées consiste en un contenu chiffré et des clés de chiffrement de contenu enveloppées pour un ou plusieurs receveurs. L'algorithme AES est utilisé pour chiffrer le contenu.

Un logiciel conforme DOIT satisfaire aux exigences pour construire le type de contenu de données enveloppés décrit à la section 6 de la [RFC3369] , "Type de contenu Données enveloppées".

La clé AES de chiffrement de contenu DOIT être générée au hasard pour chaque instance d'un type de contenu Données enveloppées. La clé de chiffrement de contenu (CEK, *content-encryption key*) est utilisée pour chiffrer le contenu.

AES peut être utilisé avec le type de contenu Données enveloppées avec toutes les techniques de gestion de clé suivantes définies à la section 6 de la [RFC3369].

1) Transport de clé : La CEK AES est enveloppée de façon univoque pour chaque receveur en utilisant la clé publique RSA du receveur et d'autres valeurs. Des détails supplémentaires sont donnés au paragraphe 2.2.

2) Accord de clés : La CEK AES est enveloppée de façon univoque pour chaque receveur en utilisant une clé de chiffrement de clé (KEK, *key-encryption key*) symétrique par paire générée en utilisant une clé privée générée au hasard par l'origine (ES-DH [RFC2631]) ou une clé privée générée à priori (SS-DH [RFC2631]) la clé publique DH du receveur, et d'autres valeurs. Le paragraphe 2.3 contient des détails complémentaires.

3) KEK symétrique à distribution préalable : La CEK AES est enveloppée en utilisant une KEK symétrique distribuée préalablement (comme une clé de liste de messagerie). Les méthodes par lesquelles la KEK symétrique est générée et distribuée sortent du domaine d'application du présent document. Les détails supplémentaires figurent au paragraphe 2.4.

4) Chiffrement de mot de passe : La CEK AES est enveloppée en utilisant une KEK déduite d'un mot de passe ou d'un secret partagé. Les détails supplémentaires figurent au paragraphe 2.5.

Les documents qui définissent l'utilisation de la structure Informations d'autre receveur devront définir l'utilisation appropriée de l'algorithme AES si désiré.

2.1 Champs EnvelopedData

Le type de contenu Données enveloppées est codé en ASN.1 en utilisant la syntaxe EnvelopedData. Les champs de la syntaxe EnvelopedData DOIVENT être remplis comme suit :

- La version EnvelopedData est déterminée sur la base d'un certain nombre de facteurs. Voir au paragraphe 6.1 de la [RFC3369] l'algorithme pour déterminer cette valeur.
- Le CHOIX recipientInfos de EnvelopedData dépend de la technique de gestion de clé utilisée. Les informations complémentaires figurent aux paragraphes 2.2, 2.3, 2.4 et 2.5.
- Le champ contentEncryptionAlgorithm de encryptedContentInfo de EnvelopedData DOIT spécifier un algorithme de chiffrement symétrique. Les mises en œuvre DOIVENT prendre en charge le chiffrement de contenu avec AES, mais elles PEUVENT aussi prendre en charge d'autres algorithmes.
- Les unprotectedAttrs de EnvelopedData PEUVENT être présents.

2.2 Champs KeyTransRecipientInfo

Le type de contenu Données enveloppées est codé en ASN.1 en utilisant la syntaxe EnvelopedData. Les champs de la syntaxe EnvelopedData DOIVENT être remplis comme suit :

- La version de KeyTransRecipientInfo DOIT être 0 ou 2. Si le RecipientIdentifier est le CHOIX issuerAndSerialNumber, la version DOIT alors être 0. Si le RecipientIdentifier est subjectKeyIdentifier, la version DOIT alors être 2.
- Le RecipientIdentifier de KeyTransRecipientInfo donne un choix pour spécifier le certificat du receveur, et donc la clé publique du receveur. Le certificat du receveur DOIT contenir une clé publique RSA. La CEK est chiffrée avec la clé publique RSA du receveur. L'autre solution issuerAndSerialNumber identifie le certificat du receveur par le nom distinctif du producteur et le numéro de série du certificat ; le subjectKeyIdentifier identifie le certificat du receveur par la valeur d'extension X.509 du subjectKeyIdentifier.
- Le champ keyEncryptionAlgorithm du KeyTransRecipientInfo spécifie l'algorithme de transport de clé (c'est-à-dire, RSAES-OAEP [RFC3560]) et les paramètres associés utilisés pour chiffrer la CEK pour le receveur.
- La encryptedKey de KeyTransRecipientInfo est le résultat du chiffrement de la CEK avec la clé publique RSA du receveur.

2.3 Champs KeyAgreeRecipientInfo

Ce paragraphe décrit les conventions pour utiliser ES-DH sur SS-DH et AES avec le type de contenu CMS Données enveloppées pour la prise en charge de l'accord de clés. Lorsque l'accord de clés est utilisé, le CHOIX keyAgreeRecipientInfo de RecipientInfo DOIT être utilisé.

La version de KeyAgreeRecipient DOIT être 3.

Le champ originatorInfo de EnvelopedData DOIT être la solution originatorKey. Les champs de l'algorithme originatorKey DOIVENT contenir l'identifiant d'objet dh-public-number avec les paramètres absents. La publicKey de originatorKey DOIT contenir la clé publique éphémère de l'origine.

L'ukm EnvelopedData PEUT être présent.

Le keyEncrytionAlgorithm de EnvelopedData DOIT être l'identifiant d'algorithme id-alg-ESDH [RFC3370].

2.3.1 Déduction de clé AES-DH/AES

La génération de la KEK AES à utiliser avec l'algorithme d'enveloppe de clé AES est faite en utilisant la méthode décrite dans la [RFC2631].

2.3.1.1 Exemple 1

ZZ est constitué des 20 octets 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13

L'algorithme d'enveloppe de clé est AES-128, de sorte qu'on a besoin de 128 bits (16 octets) de matériel de clé.

Aucun partyAInfo n'est utilisé, par conséquent, l'entrée à SHA-1 est :

```
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13      ; ZZ
  30 1b
    30 11
      06 09 60 86 48 01 65 03 04 01 05                          ; OID d'enveloppe AES-128
      04 04
        00 00 00 01                                             ; Compteur
    a2 06
      04 04
      00 00 00 80                                               ; longueur de clé
```

et le résultat est les 32 octets suivants : d6 d6 b0 94 c1 02 7a 7d e6 e3 11 72 94 a3 53 64 49 08 50 f9

Par conséquent, K= d6 d6 b0 94 c1 02 7a 7d e6 e3 11 72 94 a3 53 64

2.3.1.2 Exemple 2

ZZ est constitué des 20 octets 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13

L'algorithme d'enveloppe de clé est AES-256, de sorte qu'on a besoin de 256 bits (32 octets) de matériel de clé.

Le partyAInfo utilisé est les 64 octets

```
01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 01
01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 01
01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 01
01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 01
```

Par conséquent, l'entrée de la première invocation de SHA-1 est :

```
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13      ; ZZ
30 5f
  30 11
    06 09 60 86 48 01 65 03 04 01 2d                            ; OID d'enveloppe de AES-256
    04 04
```

```
      00 00 00 01                                     ; Compteur
  a0 42
    04 40
      01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 01 ; partyAInfo
      01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 01
      01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 01
      01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 01
  a2 06
    04 04
      00 00 01 00                                     ; longueur de clé
```

Et le résultat sont les 20 octets :

88 90 58 5C 4E 28 1A 5C 11 67 CA A5 30 BE D5 9B 32 30 D8 93

L'entrée de la seconde invocation de SHA-1 est :

```
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13   ; ZZ
30 5f
  30 11
    06 09 60 86 48 01 65 03 04 01 2d                  ; OID d'enveloppe de AES-256
    04 04
      00 00 00 02                                     ; Compteur
  a0 42
    04 40
      01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 01 ; partyAInfo
       01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 01
       01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 01
       01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 01
   a2 06
     04 04
       00 00 01 00                                    ; longueur de clé
```

Et le résultat est les 20 octets :

CB A8 F9 22 BD 1B 56 A0 71 C9 6F 90 36 C6 04 2C AA 20 94 37

Par conséquent,

K = 88 90 58 5C 4E 28 1A 5C 11 67 CA A5 30 BE D5 9B 32 30 D8 93 CB A8 F9 22 BD 1B 56 A0

2.3.2 Processus d'enveloppe de CEK AES

L'algorithme d'enveloppe de clé AES chiffre une clé AES dans une autre clé AES. L'algorithme produit un résultat plus long de 64 bits que la CEK AES d'entrée, les bits supplémentaires étant une somme de contrôle. L'algorithme utilise 6*n opérations de chiffrement/déchiffrement AES où n est un nombre de blocs de 64 bits dans la CEK AES. Tous les détails de l'algorithme AES d'enveloppe de clé sont disponibles dans la [RFC3394].

Le NIST a alloué les OID suivants pour définir l'algorithme d'enveloppe de clé AES.

```
IDENTIFIANT D'OBJET id-aes128-wrap ::= { aes 5 }
IDENTIFIANT D'OBJET id-aes192-wrap ::= { aes 25 }
IDENTIFIANT D'OBJET id-aes256-wrap ::= { aes 45 }
```

Dans tous les cas, le champ Paramètres DOIT être absent. L'OID donne la taille de la clé KEK, mais ne dit rien sur la taille de la CEK AES enveloppée. Les mises en œuvre PEUVENT utiliser des tailles de KEK et de CEK différentes. Les mises en œuvre DOIVENT accepter des CEK ayant la même longueur que la KEK. Si des longueurs différentes sont acceptées, la KEK DOIT être d'une longueur égale ou supérieure à celle de la CEK.

2.4 Champs KEKRecipientInfo

Ce paragraphe décrit les conventions pour l'utilisation d'AES avec le type de contenu CMS Données enveloppées pour la prise en charge des KEK symétriques préalablement distribuées. Lorsque une KEK symétrique préalablement distribuée est utilisée pour envelopper la CEK AES, le CHOIX KEKRecipientInfo de RecipientInfo DOIT être utilisé. Les méthodes utilisées pour générer et distribuer la KEK symétrique sortent du domaine d'application du présent document. Une méthode possible de distribution des clés est documentée dans la [RFC5275].

Le champ KEKRecipientInfo DOIT être rempli comme spécifié dans la [RFC3369] paragraphe 6.2.3, "Type KEKRecipientInfo".

Le champ d'algorithme keyEncryptionAlgorithm de KEKRecipientInfo DOIT être un des OID définis au paragraphe 2.3.2 indiquant que la fonction d'enveloppe AES est utilisée pour envelopper la CEK AES. Le champ de paramètres keyEncryptionAlgorithm de KEKRecipientInfo DOIT être absent.

Le champ encryptedKey de KEKRecipientInfo DOIT inclure la CEK AES enveloppée en utilisant la KEK symétrique distribuée préalablement comme entrée de la fonction d'enveloppe AES.

2.5 Champs PasswordRecipientInfo

Ce paragraphe décrit les conventions pour utiliser AES avec le type de contenu CMS Données enveloppées pour prendre en charge la gestion de clé fondée sur le mot de passe.

Lorsque on utilise une KEK déduite d'un mot de passe pour envelopper la CEK AES, le CHOIX PasswordRecipientInfo de RecipientInfo DOIT être utilisé.

Le champ d'algorithme keyEncryptionAlgorithm DOIT être un des OID définis au paragraphe 2.3.2 indiquant la fonction d'enveloppe AES qui est utilisée pour envelopper la CEK AES. Le champ des paramètres keyEncryptionAlgorithm DOIT être absent.

Le champ encryptedKey DOIT être le résultat de l'algorithme d'enveloppe de clé AES appliqué à la valeur de la CEK AES.

3. Conventions de données chiffrées

Le type de contenu CMS Données chiffrées consiste en un contenu chiffré avec une gestion de clé implicite. L'algorithme AES est utilisé pour chiffrer le contenu.

Un logiciel conforme DOIT satisfaire aux exigences pour la construction d'un type de contenu de données enveloppées décrites à la Section 8 de la [RFC3369], "Type de contenu Données chiffrées".

Le type de contenu Données chiffrées est codé en ASN.1 en utilisant la syntaxe EncryptededData. Les champs de la syntaxe EncryptedData DOIVENT être remplies comme suit :

- La version de EncryptedData est déterminée sur la base d'un certain nombre de facteurs. Voir au paragraphe 9.1 de la [RFC3369] l'algorithme pour déterminer cette valeur.
- Le champ contentEncryptionAlgorithm des encryptedContentInfo de EncryptedData DOIT spécifier un algorithme de chiffrement symétrique. Les mises en œuvre DOIVENT accepter le chiffrement qui utilise AES, mais elles PEUVENT aussi accepter d'autres algorithmes.
- L'attribut unprotectedAttrs de EncryptedData PEUT être présent.

4. Identifiants et paramètres d'algorithme

Cette section spécifie les identifiants d'algorithme pour l'algorithme de chiffrement AES.

4.1 Identifiants et paramètres d'algorithme AES

L'algorithme AES est défini dans [AES]. RSAES-OAEP [RFC3560] PEUT être utilisé pour transporter les clés AES.

AES s'ajoute à l'ensemble des algorithmes de chiffrement de contenu symétriques défini dans la [RFC3370]. L'algorithme AES de chiffrement de contenu, en mode de chaînage de bloc de chiffrement (CBC, *Cipher Block Chaining*) pour les trois différentes tailles de clé est identifié par les identifiants d'objet suivants :

IDENTIFIANT D'OBJET id-aes128-CBC ::= { aes 2 }
IDENTIFIANT D'OBJET id-aes192-CBC ::= { aes 22 }
IDENTIFIANT D'OBJET id-aes256-CBC ::= { aes 42 }

Le champ de paramètres AlgorithmIdentifier DOIT être présent, et le champ de paramètres DOIT contenir un vecteur d'initialisation AES :

AES-IV ::= CHAINE D'OCTETS (TAILLE(16))

Les identifiants d'algorithme de chiffrement de contenu sont localisés dans les champs EncryptedContentInfo, contentEncryptionAlgorithm de EnvelopedData et EncryptedContentInfo et contentEncryptionAlgorithm de EncryptedData.

Les algorithmes de chiffrement de contenu sont utilisés pour chiffrer le contenu situé dans les champs EncryptedContentInfo et encryptedContent de EnvelopedData et les champs EncryptedContentInfo et encryptedContent de EncryptedData.

5. Conventions d'attribut de capacités S/MIME

Un client S/MIME DEVRAIT annoncer l'ensemble de fonctions cryptographiques qu'il accepte en utilisant l'attribut de capacités S/MIME. Cet attribut donne une liste partielle des identifiants d'objet de fonctions cryptographiques et DOIT être signé par le client. Les OID d'algorithme DEVRAIENT être séparés logiquement en catégories fonctionnelles et DOIVENT être ordonnés selon leur ordre de préférence.

Le paragraphe 2.5.2 de la [RFC 2633] définit l'attribut signé SMIMECapabilities (défini comme une SEQUENCE de SEQUENCES SMIMECapability) à utiliser pour spécifier une liste partielle d'algorithmes que le logiciel qui annonce les SMIMECapabilities peut prendre en charge.

5.1 Attributs de capacité S/MIME AES

Si un client S/MIME est obligé de prendre en charge un chiffrement symétrique avec AES, l'attribut de capacités DOIT contenir l'identifiant d'objet AES spécifié ci-dessus dans la catégorie des algorithmes symétriques. Le paramètre avec ce codage DOIT être absent.

Les codages pour les taille de clé obligatoires sont :

Taille de clé	Capacité
128	30 0B 06 09 60 86 48 01 65 03 04 01 02
196	30 0B 06 09 60 86 48 01 65 03 04 01 16
256	30 0B 06 09 60 86 48 01 65 03 04 01 2A

Lorsque un agent envoyeur crée un message chiffré, il doit décider quel type d'algorithme de chiffrement utiliser. En général, le processus de décision implique des informations obtenues des listes de capacités incluses dans les messages reçus du receveur, ainsi que d'autres information telles que les accords privés, les préférences de l'utilisateur, les restrictions légales, et ainsi de suite. Si les usagers exigent AES pour un chiffrement symétrique, les clients S/MIME des deux côtés envoyeur et receveur DOIVENT le prendre en charge, et il DOIT être établi dans les préférences de l'usager.

6. Considérations de sécurité

Si RSA-OAEP [RFC2437] et RSA PKCS #1 v1.5 [RFC2313] sont tous deux utilisés pour transporter la même CEK, un attaquant peut alors quand même utiliser l'attaque de Bleichenbacher contre la clé chiffrée RSA PKCS #1 v1.5. Il n'est généralement pas conseillé de mélanger RSA-OAEP et RSA PKCS#1 v1.5 dans le même ensemble de receveurs.

Les mises en œuvre doivent protéger la clé privée RSA et la CEK. La compromission de la clé privée RSA peut conduire à la divulgation de tous les messages protégés par cette clé. La compromission de la CEK peut conduire à la divulgation du contenu chiffré associé.

La génération des CEK AES s'appuie sur des nombres aléatoires. L'utilisation de générateurs de nombres pseudo-aléatoires

(PRNG, *pseudo-random number generator*) inadéquats pour générer ces valeurs peut résulter en peu ou pas du tout de sécurité. Un attaquant peut trouver beaucoup plus facile de reproduire l'environnement du PRNG qui a produit les clés, en cherchant le petit ensemble de possibilités résultant, plutôt qu'une recherche en force brute sur tout l'espace de clés. La génération de nombres aléatoires de qualité est difficile. La [RFC1750] offre des lignes directrices importantes dans ce domaine.

Lorsque on enveloppe une CEK avec une KEK, la KEK DOIT toujours être au moins de la même longueur que la CEK. Un attaquant va généralement travailler sur le point le plus faible d'un système de chiffrement. Cela serait la plus petite des deux tailles de clés pour une attaque en force brute.

Références normatives

[AES] National Institute of Standards. FIPS Pub 197: Advanced Encryption Standard (AES). 26 novembre 2001.

[DES] National Institute of Standards and Technology. FIPS Pub 46: Data Encryption Standard. 15 janvier 1977.

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

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

[RFC3369] R. Housley, "Syntaxe de message cryptographique (CMS)", août 2002. (*Obsolète, voir* RFC5652) *(P.S.)*

[RFC3370] R. Housley, "Algorithmes de syntaxe de message cryptographique (CMS)", août 2002. *(P.S.)*

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

[RFC3560] R. Housley, "Utilisation de l'algorithme de transport de clé RSAES-OAEP dans la syntaxe de message cryptographique (CMS)", juillet 2003. *(P.S.)*

[X.208-88] Recommandation UIT-T X.208, "Spécification de la notation de syntaxe abstraite numéro un (ASN.1)". 1988.

[X.209-88] Recommandation UIT-T X.209, "Spécification des règles de codage de base pour la notation de syntaxe abstraite numéro un (ASN.1)". 1988.

[X.509-88] Recommandation UIT-T X.509, L'Annuaire – Cadre d'authentification". 1988.

Références pour information

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

[RFC2313] B. Kaliski, "PKCS n° 1 : Chiffrement RSA version 1.5", mars 1998.

[RFC2437] B. Kaliski et J. Staddon, "PKCS n° 1 : Spécifications de la cryptographie RSA version 2.0", octobre 1998. *(Obsolète, voir* RFC3447*) (Information)*

[RFC2633] B. Rmasdell, "Spécification de message S/MIME version 3", juin 1999. (*Obsolète, voir* RFC3851) *(P.S.)*

[RFC5275] S. Turner, "Gestion et distribution de clés symétriques sur CMS", juin 2008. *(P.S.)*

Remerciements

Le présent document est le résultat de contributions de nombreux professionnels. On remercie de leur dur labeur tous les membres du groupe de travail S/MIME de l'IETF.

Appendice A Module ASN.1

CMSAesRsaesOaep {iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) modules(0) id-mod-cms-aes(19) }

DEFINITIONS DES ÉTIQUETTES IMPLICITES ::=
DÉBUT

-- EXPORTE TOUT --
IMPORTES
-- PKIX
AlgorithmIdentifier
FROM PKIXExplicit88 {iso(1) identified-organization(3) dod(6)
internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
id-pkix1-explicit(18)};

-- Identifiants d'objets d'informationq AES --

IDENTIFIANT D'OBJET aes ::= { joint-iso-itu-t(2) country(16) us(840) organization(1) gov(101) csor(3)_nistAlgorithms(4) 1 }

-- AES avec le mode de chaînage CBC pour les tailles de clés de 128, 192, 256

IDENTIFIANT D'OBJET id-aes128-CBC ::= { aes 2 }
IDENTIFIANT D'OBJET id-aes192-CBC ::= { aes 22 }
IDENTIFIANT D'OBJET id-aes256-CBC ::= { aes 42 }

-- AES-IV est le paramètre pour tous les identifiants d'objete ci-dessus.

AES-IV ::= CHAINE D'OCTETS (TAILLE(16))

-- AES Key Wrap Algorithm Identifiers - Parameter is absent

IDENTIFIANT D'OBJET id-aes128-wrap ::= { aes 5 }
IDENTIFIANT D'OBJET id-aes192-wrap ::= { aes 25 }
IDENTIFIANT D'OBJET id-aes256-wrap ::= { aes 45 }

FIN

Adresse de l'auteur

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

Déclaration complète de droits de reproduction

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