Groupe de travail Réseau
Request for Comments : 3826
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

U. Blumenthal, Lucent Technologies
F. Maino, Andiamo Systems, Inc.
K. McCloghrie, Cisco Systems, Inc.
juin 2004

Algorithme de chiffrement de la norme de chiffrement évolué (AES) dans le modèle SNMP de sécurité fondée sur l'utilisateur

Statut du présent mémoire
Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître 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 un protocole de chiffrement symétrique qui s'ajoute aux protocoles décrits dans le modèle de sécurité fondé sur l'utilisateur (USM, *User-based Security Model*) qui est un sous système de sécurité pour la version 3 du protocole simple de gestion de réseau à utiliser dans l'architecture SNMP. Le protocole de chiffrement symétrique décrit dans le présent document se fonde sur l'algorithme de chiffrement de la norme de chiffrement évoluée (AES, *Advanced Encryption Standard*) utilisé dans le mode rebouclage du chiffrement (CFB, *Cipher FeedBack Mode*) avec une taille de clé de 128 bits.

Table des Matières

1. Introduction

Au sein de l'architecture de description des cadres de gestion de l'Internet [RFC3411], le modèle de sécurité fondé sur l'utilisateur (USM) [RFC3414] pour SNMPv3 est défini comme un sous système de sécurité au sein d'un moteur SNMP. La RFC3414 décrit l'utilisation de HMAC-MD5-96 et de HMAC-SHA-96 comme les protocoles d'authentification initiaux, et l'utilisation de CBC-DES comme protocole initial de confidentialité. Le modèle de sécurité fondé sur l'utilisateur permet cependant que d'autres protocoles semblables soient utilisés à la place, ou concurremment avec ces protocoles.

Le présent mémoire décrit l'utilisation de CFB128-AES-128 comme protocole de confidentialité de remplacement pour le modèle de sécurité fondé sur l'utilisateur.

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 les

BCP 14, [RFC2119].

1.1 Objectifs et contraintes

Le principal objectif du présent mémoire est de fournir un nouveau protocole de confidentialité pour l'USM fondé sur la norme de chiffrement évolué (AES, *Advanced Encryption Standard*) [FIPS-AES].

La contrainte majeure est de conserver une interchangeabilité complète du nouveau protocole défini dans ce mémoire avec les protocoles existants d'authentification et de confidentialité déjà définis dans l'USM.

Pour un certain utilisateur, le protocole de confidentialité fondé sur AES DOIT être utilisé avec un des protocoles d'authentification définis dans la RFC 3414 ou un algorithme/protocole fournissant une fonctionnalité équivalente.

1.2 Localisation de clé

Comme défini dans la [RFC3414], une clé localisée est une clé secrète partagée entre un usager U et un moteur SNMP d'autorité E. Même si un usager peut avoir seulement une paire de mots de passe d'authentification et de confidentialité (et par conséquent seulement une paire de clés) pour le réseau entier, les secrets réels partagés entre l'usager et chaque moteur SNMP d'autorité seront différents. Ceci est réalisé par la localisation de clé.

Si le protocole d'authentification défini pour un usager U au moteur SNMP d'autorité E est un des protocoles d'authentification définis dans la [RFC3414], la localisation de clé est effectuée conformément au processus en deux étapes décrit au paragraphe 2.6 de la [RFC3414].

1.3 Entropie et mémorisation de mot de passe

La sécurité des diverses fonctions cryptographiques repose dans la force des fonctions elles mêmes contre diverses formes d'attaques, et aussi, peut-être plus important, dans le matériel de chiffrement utilisé avec elles. Bien que des attaques théoriques contre les fonctions cryptographiques soient possibles, il est plus probable que deviner la clé est la principale menace.

Les recommandations suivantes sont faites aux utilisateurs de mots de passe :
- la longueur du mot de passe DEVRAIT être d'au moins 12 octets ;
- le partage de mot de passe DEVRAIT être interdit afin que des mots de passe ne soient pas partagés par plusieurs utilisateurs de SNMP ;
- les mises en œuvre DEVRAIENT prendre en charge l'utilisation de mots de passe générés au hasard comme forme supérieure de sécurité.

Il vaut la peine de se rappeler que, comme spécifié dans la [RFC3414], si le mot de passe d'un usager, ou une clé non localisée, est divulgué, la localisation de clé sera sans utilité et la sécurité du réseau peut être compromise. Donc, le mot de passe d'un usager ou une clé non localisée NE DOIT PAS être mémorisé sur un appareil/nœud géré. À la place, la clé localisée DEVRA être mémorisée (si elle l'est) de telle sorte que, en cas de compromission d'un appareil, aucun autre appareil géré ou gérant ne soit compromis.

2. Définitions

La présente MIB est écrite en SMIv2 [RFC2578].

```
DEFINITIONS DE SNMP-USM-AES-MIB  ::= DEBUT
IMPORTATIONS
   IDENTITE DE MODULE, IDENTITE D'OBJET,
    snmpModules           DE SNMPv2-SMI         -- [RFC2578]
    snmpPrivProtocols     DE SNMP-FRAMEWORK-MIB; -- [RFC3411]

IDENTITE DE MODULE snmpUsmAesMIB
DERNIERE MISE A JOUR "200406140000Z"
ORGANISATION "IETF"
CONTACT-INFO "Uri Blumenthal
              Lucent Technologies / Bell Labs
              67 Whippany Rd.
```

14D-318
Whippany, NJ 07981, USA
973-386-2163
uri@bell-labs.com

Fabio Maino
Andiamo Systems, Inc.
375 East Tasman Drive
San Jose, CA 95134, USA
408-853-7530
fmaino@andiamo.com

Keith McCloghrie
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706, USA
408-526-5260
kzm@cisco.com"

DESCRIPTION "Définitions des identités d'objets nécessaires pour l'utilisation d'AES par le modèle de sécurité fondé sur l'utilisateur de SNMP. Copyright (C) The Internet Society (2004). Cette version du module de MIB fait partie de la RFC 3826 ; voir dans la RFC elle-même les notices légales complètes. Des informations supplémentaires sont disponibles à http://www.ietf.org/copyrights/ianamib.html ".
REVISION : "200406140000Z"
DESCRIPTION : "Version initiale, publiée comme RFC3826"

::= { snmpModules 20 }

IDENTITE D'OBJET usmAesCfb128Protocol
STATUT : actuel
DESCRIPTION : "Protocole de confidentialité CFB128-AES-128".
REFERENCE : "Specification for the ADVANCED ENCRYPTION STANDARD. Federal Information Processing Standard (FIPS) Publication 197. (November 2001). Dworkin, M., NIST Recommendation for Block Cipher Modes of Operation, Methods and Techniques. NIST Special Publication 800-38A (December 2001)".
::= { snmpPrivProtocols 4 }

FIN

3. Protocole de chiffrement symétrique CFB128-AES-128

Cette section décrit un protocole de chiffrement symétrique fondé sur l'algorithme de chiffrement AES [FIPS-AES], utilisé en mode de rebouclage de chiffrement comme décrit dans [AES-MODE], en utilisant des clés de chiffrement d'une taille de 128 bits.

Ce protocole est identifié par usmAesCfb128PrivProtocol.

Le protocole usmAesCfb128PrivProtocol est une solution de remplacement du protocole de confidentialité défini dans la [RFC3414].

3.1 Mécanismes

Un algorithme de chiffrement est nécessaire pour la prise en charge de la confidentialité des données. Une portion appropriée du message est chiffrée avant qu'il soit transmis. Le modèle de sécurité fondé sur l'utilisateur spécifie que scopedPDU est la portion du message qui a besoin d'être chiffrée.

Une valeur secrète est partagée par tous les moteurs SNMP qui peuvent légitimement générer des messages au nom de l'utilisateur approprié. Cette valeur secrète, en combinaison avec une valeur d'opportunité et un entier de 64 bits, est utilisé pour créer la clé (localisée) de chiffrement/déchiffrement et la valeur d'initialisation (IV).

3.1.1 Protocole de chiffrement symétrique fondé sur AES

Le protocole de chiffrement symétrique défini dans le présent mémoire fournit la prise en charge de la confidentialité des données. La portion désignée d'un message SNMP est chiffrée et incluse au titre du message envoyé au destinataire.

La norme de chiffrement évoluée (AES, *Advanced Encryption Standard*) est l'algorithme de chiffrement symétrique que l'Institut *(américain)* des normes et technologies (NIST, *National Institute of Standards and Technology*) a choisi après un processus de sélection de quatre ans comme remplacement de la norme de chiffrement de données (DES, *Data Encryption Standard*).

La page d'accueil d'AES, à http://www.nist.gov/aes, contient quantité d'informations sur AES incluant la norme fédérale de traitement des informations [FIPS-AES] qui est la spécification complète de la norme de chiffrement évolué.

Les paragraphes qui suivent contiennent la description des caractéristiques pertinentes des chiffrements AES utilisés dans le protocole de chiffrement symétrique décrit dans le présent mémoire.

3.1.1.1 Mode de fonctionnement

La publication spéciale NIST 800-38A [AES-MODE] recommande cinq modes de fonctionnement de confidentialité à utiliser avec AES : le mode dictionnaire (ECB, *Electronic Codebook*), le chaînage de bloc de chiffrement (CBC, *Cipher Block Chaining*, le mode rebouclage du chiffrement (CFB, *Cipher Feedback*), le mode rebouclage de la sortie (OFB, *Output Feedback*), et le mode compteur (CTR, *Counter*).

Le protocole de chiffrement symétrique décrit dans le présent mémoire utilise AES en mode CFB avec le paramètre *s* (nombre de bits de retour) réglé à 128 conformément à la définition du mode CFB donnée dans [AES-MODE]. Ce mode exige une valeur d'initialisation (IV) de même taille que la taille de bloc de l'algorithme de chiffrement.

3.1.1.2 Taille de clé

Dans le protocole de chiffrement décrit dans le présent mémoire, AES est utilisé avec une taille de clé de 128 bits, comme recommandé dans [AES-MODE].

3.1.1.3 Taille de bloc et bourrage

La taille de bloc des algorithmes de chiffrement AES utilisés dans le protocole de chiffrement décrit dans le présent mémoire est de 128 bits, comme recommandé dans [AES-MODE].

3.1.1.4 Tours

Ce paramètre détermine combien de fois un bloc est chiffré. Le protocole de chiffrement décrit dans le présent mémoire utilise 10 tours, comme recommandé dans [AES-MODE].

3.1.2 Clés localisées, clé de chiffrement AES, et valeur d'initialisation

La taille de la clé localisée (Kul) d'un usager SNMP, comme décrit dans la [RFC3414], dépend du protocole d'authentification défini pour cet usager U au moteur SNMP d'autorité E.

Le protocole de chiffrement décrit dans le présent mémoire DOIT être utilisé avec un protocole d'authentification qui génère une clé localisée d'au moins 128 bits. Les protocoles d'authentification décrits dans la [RFC3414] satisfont cette exigence.

3.1.2.1 Clé de chiffrement et IV AES

Les 128 bits premiers bits de la clé localisée Kul sont utilisés comme clé de chiffrement AES. La IV de 128 bits est obtenue par l'enchaînement des 32 bits de snmpEngineBoots du moteur SNMP d'autorité, des 32 bits du snmpEngineTime du moteur SNMP, et d'un entier local de 64 bits. L'entier de 64 bits est initialisé comme une valeur pseudo aléatoire au moment de l'amorçage.

L'IV est enchaînée comme suit : les 32 bits du snmpEngineBoots sont convertis en les quatre premiers octets (octet de poids fort en premier) les 32 bits de snmpEngineTime sont convertis en les quatre octets suivants (octet de poids fort en premier) et les 64 bits de l'entier sont ensuite convertis en les huit derniers octets (octet de poids fort en premier). L'entier de 64 bits est alors mis dans le champ msgPrivacyParameters codé comme une CHAINE D'OCTETS d'une longueur de 8 octets. L'entier est alors modifié pour le message suivant. On recommande qu'il soit incrémenté de un jusqu'à ce qu'il atteigne sa valeur

maximum, moment auquel il revient à sa valeur initiale.

Une mise en œuvre peut utiliser toute méthode pour faire varier la valeur de l'entier local de 64 bits, pourvu que la méthode choisie ne génère jamais une IV dupliquée pour la même clé.

Une IV dupliquée peut résulter en l'événement très improbable que plusieurs gestionnaires, communiquant avec un seul moteur d'autorité, choisissent tous accidentellement le même entier de 64 bits dans la même seconde. La probabilité d'un tel événement est très faible, et n'affecte pas de façon significative la robustesse du mécanisme proposé.

L'entier de 64 bits doit être placé dans le champ privParameters pour permettre à l'entité receveuse de calculer l'IV correcte et déchiffrer le message. Cette valeur de 64 bits est appelée le "sel" dans le présent document.

Noter qu'envoyeur et receveur doivent utiliser la même valeur d'IV, c'est-à-dire, ils doivent tous deux utiliser les mêmes valeurs de composants individuels utilisés pour créer l'IV. En particulier, envoyeur et receveur doivent utiliser les valeurs de snmpEngineBoots, snmpEngineTime, et l'entier de 64 bits qui sont contenus dans le message pertinent (respectivement dans les champs msgAuthoritativeEngineBoots, msgAuthoritativeEngineTime, et privParameters).

3.1.3 Chiffrement des données

Les données à chiffrer sont traitées comme une séquence d'octets.

Les données sont chiffrées en mode de rebouclage de chiffrement avec le paramètre *s* réglé à 128 conformément à la définition du mode CFB données au paragraphe 6.3 de [AES-MODE]. Un diagramme clair du processus de chiffrement et de déchiffrement est donné à la Figure 3 de [AES-MODE].

Le texte source est divisé en blocs de 128 bits. Le dernier bloc peut avoir moins de 128 bits, et aucun bourrage n'est requis.

Le premier bloc entré est l'IV, et l'opération de suite du chiffrement est appliquée à l'IV pour produire le premier bloc de résultat. Le premier bloc de texte chiffré est produit par l'opération OU exclusif sur le premier bloc de texte source avec le premier bloc de résultat. Le bloc de texte chiffré est aussi utilisé comme bloc d'entrée pour l'opération suivante de chiffrement.

Le processus est répété avec les blocs successifs d'entrée jusqu'à ce qu'un segment de texte chiffré soit produit pour chaque segment de texte source.

Le dernier bloc de texte chiffré est produit en effectuant l'opération OU exclusif sur le dernier segment de texte source de *r* bits (r est inférieur ou égal à 128) avec le segment des r bits de poids fort du dernier bloc de résultat.

3.1.4 Déchiffrement des données

En déchiffrement de CFB, l'IV est le premier bloc d'entrée, le premier texte chiffré est utilisé comme second bloc d'entrée, le second texte chiffré est utilisé pour le troisième bloc d'entrée, etc. La fonction de chiffrement vers l'avant est appliquée à chaque bloc d'entrée pour produire les blocs de résultat. Les blocs de résultat sont traités avec l'opération OU exclusif avec les blocs de texte chiffré correspondants pour retrouver les blocs de texte source.

Le dernier bloc de texte chiffré (dont la taille r est inférieure ou égale à 128) est traitée avec l'opération OU exclusif avec le segment des *r* bits de poids fort du dernier bloc de résultat pour retrouver le dernier bloc de texte source de *r* bits.

3.2 Éléments du protocole de confidentialité AES

Ce paragraphe contient les définitions nécessaires pour réaliser les modules de confidentialité définis par le présent mémoire.

3.2.1 Utilisateurs

Le chiffrement/déchiffrement des données en utilisant ce protocole de chiffrement symétrique fait usage d'un ensemble défini de noms d'utilisateurs. Pour tout utilisateur au nom duquel un message doit être chiffré/déchiffré à un moteur SNMP particulier, ce moteur SNMP doit avoir connaissance de cet utilisateur. Un moteur SNMP qui a besoin de communiquer avec un autre moteur SNMP doit aussi avoir connaissance d'un utilisateur connu de ce moteur SNMP, incluant la connaissance des attributs applicables de cet utilisateur.

Un utilisateur et ses attributs sont définis comme suit :

<userName> *(nom d'utilisateur)* une chaîne d'octets représentant le nom de l'utilisateur.

<privAlg> *(algorithme privé)* l'algorithme utilisé pour protéger contre la divulgation les messages générés au nom de l'utilisateur.

<privKey> *(clé privée)* la clé secrète de l'utilisateur à utiliser comme entrée pour la génération de la clé localisée pour chiffrer/déchiffrer les messages générés au nom de l'utilisateur. La longueur de cette clé DOIT être supérieure ou égale à 128 bits (16 octets).

<authAlg> *(algorithme d'authentification)* c'est l'algorithme utilisé pour authentifier les messages générés au nom de l'utilisateur, qui est aussi utilisé pour générer la version localisée de la clé secrète.

3.2.2 msgAuthoritativeEngineID

La valeur msgAuthoritativeEngineID *(identifiant de moteur d'autorité de message)* contenue dans un message authentifié spécifie le moteur SNMP d'autorité pour ce message particulier (voir la définition de SnmpEngineID dans le document d'architecture SNMP [RFC3411]).

La clé de confidentialité (privée) de l'utilisateur est différente sur chaque moteur SNMP d'autorité, et donc le snmpEngineID est utilisé pour choisir la clé appropriée pour le processus de chiffrement/déchiffrement.

3.2.3 Messages SNMP utilisant ce protocole de confidentialité

Les messages qui utilisent ce protocole de confidentialité portent un champ msgPrivacyParameters au titre du msgSecurityParameters. Pour ce protocole, le champ privParameters est la CHAINE D'OCTETS qui représente le "sel" qui a été utilisé pour créer l'IV.

3.2.4 Services fournis par les modules de confidentialité d'AES

Ce paragraphe décrit les entrées et les résultats que le module de confidentialité AES attend et produit lorsque le module de sécurité fondée sur l'utilisateur invoque un des modules de confidentialité AES pour des services.

3.2.4.1 Services pour chiffrer les données sortantes

Le protocole de confidentialité AES suppose que le choix de la clé privée est fait par l'appelant, et que celui-ci passe la clé secrète localisée à utiliser.

Lorsque c'est fini, le module de confidentialité retourne les informations d'état (statusInformation) et, si le processus de chiffrement a réussi, la encryptedPDU et les msgPrivacyParameters codés comme CHAINE D'OCTETS. La primitive de service abstrait est :

```
statusInformation =              -- succès ou échec
encryptData(
    IN    encryptKey             -- clé secrète pour le chiffrement
    IN    dataToEncrypt          -- données à chiffrer (scopedPDU)
    OUT   encryptedData          -- données chiffrées (encryptedPDU)
    OUT   privParameters         -- rempli par le fournisseur de service
      )
```

Les éléments de données abstraits sont :

statusInformation : indication de succès ou d'échec du processus de chiffrement. En cas d'échec, c'est une indication de l'erreur.

encryptKey : clé secrète à utiliser par l'algorithme de chiffrement. La longueur de cette clé DOIT être de 16 octets.

dataToEncrypt : ce sont les données à chiffrer.

encryptedData : ce sont les données chiffrées après achèvement réussi du processus.

privParameters : ce sont les paramètres de confidentialité codés en une CHAINE D'OCTETS.

3.2.4.2 Services pour déchiffrer les données entrantes

Ce protocole de confidentialité AES suppose que le choix de la clé privée est fait par l'appelant et que l'appelant passe la clé localisée à utiliser.

Quand le processus est achevé, le module de confidentialité retourne les informations d'état et, si le processus de déchiffrement a réussi, les scopedPDU en texte source. La primitive de service abstraite est :

```
statusInformation =
    decryptData(
    IN   decryptKey          -- clé secrète pour le déchiffrement
    IN   privParameters      -- comme reçus sur le réseau
    IN   encryptedData       -- données chiffrées (encryptedPDU)
    OUT  decryptedData       -- données déchiffrées (scopedPDU)
         )
```

Les éléments de données abstraits sont :

statusInformation : indique si les données ont été bien déchiffrées, et sinon, l'indication de l'erreur.

decryptKey : la clé secrète à utiliser par l'algorithme de déchiffrement. La longueur de cette clé DOIT être 16 octets.

privParameters : entier de 64 bits à utiliser pour calculer l'IV.

encryptedData : les données à déchiffrer.

decryptedData : les données déchiffrées.

3.3 Éléments de procédure

Ce paragraphe décrit les procédures du protocole de confidentialité AES pour le modèle de sécurité fondé sur l'utilisateur de SNMP.

3.3.1 Traitement d'un message sortant

Ce paragraphe décrit la procédure suivie par un moteur SNMP chaque fois qu'il doit chiffrer une partie d'un message sortant en utilisant le usmAesCfb128PrivProtocol.

1) La encryptKey secrète est utilisée pour construire la clé de chiffrement AES, comme décrit au paragraphe 3.1.2.1.

2) Le champ privParameters est rangé en séries conformément aux règles d'une CHAINE D'OCTETS de la [RFC3417] représentant l'entier de 64 bits qui va être utilisé dans l'IV comme décrit au paragraphe 3.1.2.1.

3) La scopedPDU est chiffrée (comme décrit au paragraphe 3.1.3) et les données chiffrées sont rangées en série conformément aux règles de la [RFC3417] comme une CHAINE D'OCTETS.

4) La CHAINE D'OCTETS rangée en série qui représente la scopedPDU chiffrée ainsi que les privParameters et statusInformation indiquant le succès est retournée au module appelant.

3.3.2 Traitement d'un message entant

Ce paragraphe décrit la procédure suivie par un moteur SNMP chaque fois qu'il doit déchiffrer une partie d'un message entrant en utilisant le usmAesCfb128PrivProtocol.

1) Si le champ privParameters n'est pas une CHAINE D'OCTETS de 8 octets, une indication d'erreur (decryptionError) est alors retournée au module appelant.

2) L'entier de 64 bits est extrait du champ privParameters.

3) La decryptKey secrète et l'entier de 64 bits sont alors utilisés pour construire la clé de déchiffrement AES et l'IV qui est calculée comme décrit au paragraphe 3.1.2.1.

4) La encryptedPDU est alors déchiffrée (comme décrit au paragraphe 3.1.4).

5) Si la encryptedPDU ne peut pas être déchiffrée, une indication d'erreur (decryptionError) est retournée au module appelant.

6) La scopedPDU déchiffrée et les statusInformation indiquant la réussite sont retournées au module appelant.

4. Considérations sur la sécurité

La sécurité des fonctions cryptographiques définies dans le présent document réside à la fois dans la force des fonctions elles-mêmes contre diverses formes d'attaque, et aussi, peut-être plus important, dans le matériel de chiffrement utilisé avec elles. Les recommandations du paragraphe 1.3 DEVRAIENT être suivies pour assurer une entropie maximum aux mots de passe choisis, et pour protéger les mots de passe mémorisés.

La sécurité du mode CFB repose sur l'utilisation d'une IV unique pour chaque message chiffré avec la même clé [CRYPTO-B]. Si la IV n'est pas unique, un cryptanalyste peut récupérer le texte en clair correspondant.

Le paragraphe 3.1.2.1 définit une procédure pour déduire la IV d'un entier local de 64 bits (le sel) initialisé comme valeur pseudo aléatoire au moment de l'amorçage. Une mise en œuvre peut utiliser toute méthode pour faire varier la valeur de l'entier local de 64 bits, pourvu que la méthode choisie ne génère jamais une IV dupliquée pour la même clé.

La procédure du paragraphe 3.1.2.1 suggère une méthode pour faire varier la valeur de l'entier local de 64 bits qui génère des IV uniques pour chaque message. Cette méthode peut résulter en une IV dupliquée dans le cas très improbable où plusieurs gestionnaires, communiquant avec un seul moteur d'autorité, vont choisir accidentellement ensemble le même entier de 64 bits dans la même seconde. La probabilité d'un tel événement est très faible, et n'affecte pas significativement la robustesse des mécanismes proposés.

Ce protocole de confidentialité fondé sur AES DOIT être utilisé avec un des protocoles d'authentification définis dans la [RFC3414] ou avec un algorithme/protocole fournissant une fonctionnalité équivalente (incluant l'intégrité) parce que le mode de chiffrement CFB ne détecte pas les modifications de texte chiffré.

Pour plus de considérations sur la sécurité, le lecteur est invité à lire la [RFC3414], et les documents qui décrivent les algorithmes réels de chiffrement.

5. Considérations relatives à l'IANA

L'IANA a alloué l'OID 20 au module snmpUsmAesMIB dans la sous arborescence snmpModules, tenu dans le registre à http://www.iana.org/assignments/smi-numbers.

L'IANA a alloué l'OID 4 pour le protocole usmAesCfb128Protocol sous le point d'enregistrement snmpPrivProtocols, comme défini dans la [RFC3411].

6. Remerciements

Des portions de ce texte, ainsi que sa structure générale, ont été tirées sans complexe de la [RFC3414]. Les auteurs remercient les membres du groupe de travail SNMPv3 de leur aide, en particulier Wes Hardaker, Steve Moulton, Randy Presuhn, David Town, et Bert Wijnen. Des discussions sur la sécurité avec Steve Bellovont aidé à peaufiner le présent protocole.

7. Références

7.1 Références normatives

[AES-MODE] Dworkin, M., "NIST Recommendation for Block Cipher Modes of Operation, Methods and Techniques", NIST Special Publication 800-38A, décembre 2001.

[FIPS-AES] "Specification for the ADVANCED ENCRYPTION STANDARD (AES)", Federal Information Processing Standard (FIPS) Publication 197, novembre 2001.

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

[RFC2578] K. McCloghrie, D. Perkins, J. Schoenwaelder, "Structure des informations de gestion, version 2 (SMIv2)", avril 1999. (STD0058)

[RFC3411] D. Harrington, R. Presuhn, B. Wijnen, "Architecture de description des cadres de gestion du protocole simple de gestion de réseau (SNMP)", décembre 2002. (*MàJ par* RFC5343) (STD0062)

[RFC3414] U. Blumenthal, B. Wijnen, "Modèle de sécurité fondée sur l'utilisateur (USM) pour la version 3 du protocole simple de gestion de réseau (SNMPv3)", décembre 2002. (STD0062)

[RFC3417] R. Presuhn, éd. "Transpositions de transport pour le protocole simple de gestion de réseau (SNMP)", décembre 2002. (*MàJ par* RFC4789) (STD0062)

7.2 Référence pour information

[CRYPTO-B] Bellovin, S., "Probable Plaintext Cryptanalysis of the IP Security Protocols", Proceedings of the Symposium on Network and Distributed System Security, San Diego, CA, pp. 155-160, février 1997.

8. Adresse des auteurs

Uri Blumenthal
Lucent Technologies / Bell Labs
67 Whippany Rd.
14D-318
Whippany, NJ 07981, USA
téléphone : 973-386-2163
mél : uri@bell-labs.com

Fabio Maino
Andiamo Systems, Inc.
375 East Tasman Drive
San Jose, CA. 95134 USA
téléphone : 408-853-7530
mél : fmaino@andiamo.com

Keith McCloghrie
Cisco Systems, Inc.
170 East Tasman Drive
San Jose, CA. 95134-1706 USA
téléphone : 408-526-5260
mé kzm@cisco.com

9. Déclaration complète de droits de reproduction

droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf- ipr@ietf.org .

Remerciement

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