Groupe de travail Réseau
Request for Comments : 3686
Catégorie : En cours de normalisation

R. Housley, Vigil Security
janvier 2004
Traduction Claude Brière de L'Isle

Utilisation du mode Compteur de la norme de chiffrement évolué (AES) avec l'encapsulation de la charge utile de sécurité (ESP) dans IPsec

Résumé
Le présent document décrit l'utilisation du mode compteur de la norme de chiffrement évolué (AES, *Advanced Encryption Standard*) avec un vecteur d'initialisation explicite, comme mécanisme de confidentialité d'encapsulation de charge utile de sécurité (ESP, *Encapsulating Security Payload*) IPsec.

Table des matières

1. Introduction

L'Institut National des normes et des technologies (NIST) a récemment choisi la norme de chiffrement évolué (AES, *Advanced Encryption Standard*) [AES], aussi appelée Rijndael. AES est un chiffrement de bloc, et il peut être utilisé dans de nombreux modes différents. Le présent document décrit l'utilisation de AES en mode compteur (AES-CTR) avec un vecteur d'initialisation (IV, *initialization vector*) explicite, comme mécanisme de confidentialité à encapsulation de charge utile de sécurité (ESP, *Encapsulating Security Payload*) IPsec [RFC2406].

Le présent document ne donne pas de présentation d'IPsec. Cependant, des informations sur la façon dont les divers composants d'IPsec et la façon dont ils procurent collectivement des services de sécurité sont disponibles dans les [RFC2401] et [RFC2411].

1.1 Conventions utilisées dans le document

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

2. Chiffrement de bloc AES

La présente section contient une brève description des caractéristiques pertinentes du chiffrement de bloc AES. Les exigences de mise en œuvre sont aussi présentées.

2.1 Mode compteur

Le NIST a défini cinq modes de fonctionnement pour AES et les autres chiffrements de bloc [MODES] approuvés par FIPS. Chacun de ces modes a des caractéristiques différentes. Les cinq modes sont le mode dictionnaire (ECB, *Electronic Code Book*) le chaînage de bloc de chiffrement (CBC, *Cipher Block Chaining*) le chiffrement à rebouclage par le chiffre (CFB, *Cipher FeedBack*) le rebouclage de la sortie (OFB, *Output FeedBack*) et le mode compteur (CTR, *Counter*).

Seul le mode compteur AES (AES-CTR) est discuté dans la présente spécification. AES-CTR exige du chiffreur qu'il génère une valeur unique par paquet, et qu'il communique cette valeur au déchiffreur. La présente spécification appelle cette valeur par paquet un vecteur d'initialisation (IV). La même combinaison d'IV et de clé NE DOIT PAS être utilisée plus d'une fois. Le chiffreur peut générer l'IV de toute façon qui assure l'unicité. Les approches courantes de génération d'IV incluent d'incrémenter un compteur pour chaque paquet et registre à décalage avec réinjection linéaire (LFSR, *linear feedback shift register*).

La présente spécification invite à utiliser un nom occasionnel *(nonce)* pour une protection supplémentaire contre les attaques de pré calcul. La valeur du nom occasionnel n'a pas besoin d'être secrète. Cependant, le nom occasionnel DOIT être imprévisible avant l'établissement de l'association de sécurité IPsec qui fait usage de AES-CTR.

AES-CTR a de nombreuses propriétés qui le rendent attractif comme algorithme de chiffrement pour un réseautage à grande vitesse. AES-CTR utilise le chiffrement de bloc AES pour créer un chiffrement de flux. Les données sont chiffrées et déchiffrées en les soumettant à l'opération OUX avec le flux de clé produit par AES en chiffrant les valeurs de bloc de compteur séquentiel. AES-CTR est aisé à mettre en œuvre, et AES-CTR peut être traité en parallèle. AES-CTR prend aussi en charge le pré calcul de flux de clés.

Le traitement en parallèle est possible parce que AES est à plusieurs tours (voir le paragraphe 2.2). Une mise en œuvre matérielle (et certaines mises en œuvre de logiciel) peuvent créer un canal parallèle en déroulant la boucle impliquée par cette structure en tours. Par exemple, après qu'un bloc de 16 octets a été mis en entrée, un tour après, un autre bloc de 16 octets peut être entré, et ainsi de suite. Dans AES-CTR, ces entrées sont les valeurs séquentielles de bloc de compteur utilisées pour générer le flux de clés.

Plusieurs mises en œuvre de chiffrement AES indépendantes peuvent aussi être utilisées pour améliorer les performances. Par exemple, on pourrait utiliser deux mises en œuvre de chiffrement AES en parallèle pour traiter une séquence de valeurs de bloc compteur, doublant le débit effectif.

L'envoyeur peut pré calculer le flux de clés. Comme le flux de clés ne dépend d'aucune données du paquet, le flux de clés peut être pré calculé une fois que le nom occasionnel et l'IV sont alloués. Ce pré calcul peut réduire la latence de paquet. Le receveur ne peut pas effectuer un pré calcul similaire parce que l'IV ne sera pas connu avant que le paquet n'arrive.

AES-CTR utilise la seule opération de chiffrement d'AES (pour le chiffrement et le déchiffrement) rendant les mises en œuvre de AES-CTR plus petites que les mises en œuvre de beaucoup des autres modes d'AES.

Lorsque il est utilisé correctement, AES-CTR fournit un haut niveau de confidentialité. Malheureusement, il est facile d'utiliser AES-CTR de façon incorrecte. Comme c'est un chiffrement de flux, toute réutilisation d'une valeur par paquet,

appelée l'IV, avec le même nom occasionnel et la même clé, est catastrophique. Une collision d'IV donne immédiatement lieu à des fuites d'informations sur le texte en clair dans les deux paquets. Pour cette raison, il est inapproprié d'utiliser ce mode de fonctionnement avec des clés statiques. Des mesures extraordinaires seraient nécessaires pour empêcher la réutilisation d'une valeur d'IV avec la clé statique à travers les cycles de chiffrement. Pour être sûres, les mises en œuvre DOIVENT utiliser des clés fraîches avec AES-CTR. Le protocole d'échange de clés Internet (IKE, *Internet Key Exchange*) [RFC2409] peut être utilisé pour établir des clés fraîches. IKE peut aussi fournir la valeur du nom occasionnel.

Avec AES-CTR, il est trivial d'utiliser un texte chiffré valide pour falsifier d'autres textes chiffrés (valides pour le déchiffreur). Donc, il est également catastrophique d'utiliser AES-CTR sans une fonction d'authentification qui l'accompagne. Les mises en œuvre DOIVENT utiliser AES-CTR en conjonction avec une fonction d'authentification telle que HMAC-SHA-1-96 [RFC2404].

Pour chiffrer une charge utile avec AES-CTR, le chiffreur partage le texte en clair (PT, *plaintext*) en blocs de 128 bits. Le bloc final n'a pas besoin de faire 128 bits; il peut en avoir moins.

```
PT = PT[1] PT[2] ... PT[n]
```

Chaque bloc de PT est OUXé avec un bloc du flux de clés pour générer le texte chiffré (CT, *ciphertext*). Le chiffrement en AES de chaque bloc compteur résulte en 128 bits de flux de clé. Les 96 bits de poids fort du bloc compteur sont réglés à la valeur du nom occasionnel, qui est de 32 bits, suivie par la valeur de l'IV par paquet, qui est de 64 bits. Les 32 bits de moindre poids du bloc compteur sont initialement réglés à un. Cette valeur de compteur est incrémentée de un pour générer les blocs compteur suivants, chacun résultant en un autre flux de clés de 128 bit. Le chiffrement de n blocs de texte en clair peut être résumé par :

```
CTRBLK := NONCE || IV || UN
POUR i := 1 à n-1 FAIRE
  CT[i] := PT[i] OUX AES(CTRBLK)
  CTRBLK := CTRBLK + 1
FIN
CT[n] := PT[n] OUX TRUNC(AES(CTRBLK))
```

La fonction AES() effectue le chiffrement AES avec la clé fraîche.

La fonction TRUNC() tronque le résultat de l'opération de chiffrement AES à la même longueur que le bloc final de texte en clair, retournant les bits de poids fort.

Le déchiffrement est similaire. Le déchiffrement de n blocs de texte chiffré peut être résumé par :

```
CTRBLK := NONCE || IV || UN
POUR i := 1 à n-1 FAIRE
  PT[i] := CT[i] OUX AES(CTRBLK)
 CTRBLK := CTRBLK + 1
FIN
PT[n] := CT[n] OUX TRUNC(AES(CTRBLK))
```

2.2 Taille de clé et tours

AES prend en charge trois tailles de clés : 128 bits, 192 bits, et 256 bits. La taille de clé par défaut est de 128 bits, et toutes les mises en œuvre DOIVENT prendre en charge cette taille de clé. Une mise en œuvre PEUT aussi prendre en charge les tailles de clé de 192 bits et 256 bits.

AES utilise un nombre de tours différent pour chacune des tailles de clé définies. Lorsque une clé de 128 bits est utilisée, la mise en œuvre DOIT utiliser 10 tours. Lorsque une clé de 192 bits est utilisée, la mise en œuvre DOIT utiliser 12 tours. Lorsque une clé de 256 bits est utilisée, la mise en œuvre DOIT utiliser 14 tours.

2.3 Taille de bloc

AES a une taille de bloc de 128 bits (16 octets). À ce titre, lorsque on utilise AES-CTR, chaque opération de chiffrement AES génère 128 bits de flux de clés. Le chiffrement AES-CTR est le OUX du flux de clé avec le texte en clair. Le déchiffrement de AES-CTR est le OUX du flux de clés avec le texte chiffré. Si le flux de clés généré est plus long que le

texte en clair ou que le texte chiffré, les bits supplémentaires du flux de clés sont simplement éliminés. Pour cette raison, AES-CTR n'exige pas que le texte en clair soit bourré jusqu'à un multiple de la taille de bloc. Cependant, pour fournir la confidentialité à un flux de trafic limité, le bourrage PEUT être inclus, comme spécifié dans la [RFC2406].

3. Charge utile ESP

La charge utile ESP se compose de l'IV suivi par le texte chiffré. Le champ Charge utile, comme défini dans la [RFC2406], est structuré comme indiqué à la Figure 1.

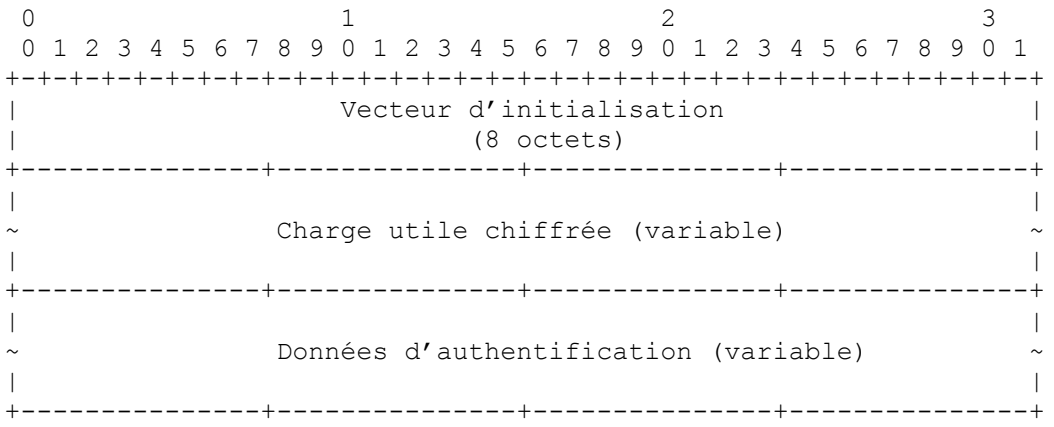

Figure 1 : Charge utile ESP chiffrée avec AES-CTR

3.1 Vecteur d'initialisation

Le champ IV d'AES-CTR DOIT faire huit octets. L'IV DOIT être choisi par le chiffreur de manière à assurer que la même valeur d'IV est utilisée une seule fois pour une clé donnée. Le chiffreur peut générer l'IV de toute manière qui assure l'unicité. Les approches courantes de génération d'IV incluent d'incrémenter un compteur pour chaque paquet et registre à décalage avec réinjection linéaire (LFSR, *linear feedback shift register*).

Inclure l'IV dans chaque paquet assure que le déchiffreur peut générer le flux de clés nécessaire pour le déchiffrement, même lorsque quelques paquets sont perdus ou décalés.

3.2 Charge utile chiffrée

La charge utile chiffrée contient le texte chiffré.

Le mode AES-CTR n'exige pas de bourrage du texte en clair. Cependant, ESP exige le bourrage pour aligner les données d'authentification sur des mots de 32 bits. Le bourrage, la Longueur de bourrage, et le Prochain en-tête DOIVENT être enchaînés avec le texte en clair avant d'effectuer le chiffrement, comme décrit dans la [RFC2406].

3.3 Données d'authentification

Comme il est trivial de construire un texte chiffré AES-CTR falsifié à partir d'un texte chiffré AES-CTR valide, les mises en œuvre de AES-CTR DOIVENT employer une méthode d'authentification ESP non NULLE. HMAC-SHA-1-96 [RFC2404] est un bon choix probable.

4. Format de bloc compteur

Chaque paquet porte le IV qui est nécessaire pour construire la séquence de blocs compteurs utilisée pour générer le flux de clés nécessaire pour déchiffrer la charge utile. Le bloc de chiffrement de bloc compteur AES est de 128 bits. La Figure 2 montre le format du bloc compteur.

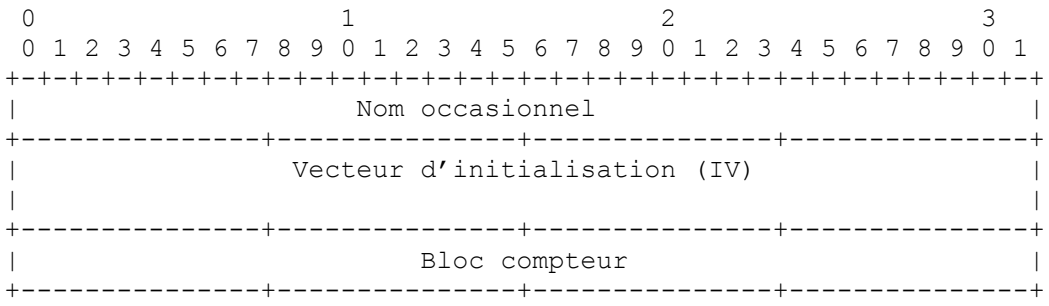

Figure 2 : Format de bloc compteur

Les composants du bloc compteur sont les suivants :

Nom occasionnel *(Nonce)*
Le champ Nom occasionnel est de 32 bits. Comme son nom l'implique, le nom occasionnel est une valeur à usage unique. C'est à dire qu'une valeur de nom occasionnel fraîche DOIT être allouée pour chaque association de sécurité. Elle DOIT être allouée au début de l'association de sécurité. La valeur du nom occasionnel n'a pas besoin d'être secrète, mais elle DOIT être imprévisible avant le début de l'association de sécurité.

Vecteur d'initialisation
Le champ IV fait 64 bits. Comme décrit au paragraphe 3.1, le IV DOIT être choisi par le chiffreur d'une manière qui assure que la même valeur d'IV n'est utilisée qu'une seule fois pour une certaine clé.

Bloc compteur
Le champ Bloc compteur est fait des 32 bits de moindre poids du bloc compteur. Le bloc compteur commence à la valeur de un, et il est incrémenté pour générer les portions suivantes du flux de clés. Le bloc compteur est une valeur d'entier de 32 bits gros boutiens.

En utilisant le processus de chiffrement décrit au paragraphe 2.1, cette construction permet que chaque paquet comporte jusqu'à :

(2^32)-1 blocs = 4 294 967 295 blocs = 68 719 476 720 octets

Cette construction peut produire assez de flux de clés pour chaque paquet pour traiter tout jumbogramme [RFC2675] IPv6.

5. Conventions IKE

Cette section décrit les conventions utilisées pour générer le matériel de clés et de noms occasionnels à utiliser avec AES-CTR en utilisant le protocole d'échange de clés Internet (IKE, *Internet Key Exchange*) [RFC2409]. Les identifiants et attributs nécessaires pour négocier une association de sécurité qui utilise AES-CTR sont aussi définis.

5.1 Matériel de clé et nom occasionnel

Comme décrit au paragraphe 2.1, les mises en œuvres DOIVENT utiliser des clés fraîches avec AES-CTR. IKE peut être utilisé pour établir des clés fraîches. Ce paragraphe décrit les conventions pour obtenir de IKE une valeur de nom occasionnel imprévisible. Noter que cette convention fournit une valeur de nom occasionnel qui est secrète autant qu'imprévisible.

IKE utilise une fonction pseudo-aléatoire (PRF, *pseudo-random function*) pour déduire le matériel de clé. La PRF est utilisée de façon itérative pour déduire du matériel de clé de longueur arbitraire, appelé KEYMAT. Le matériel de clé est extrait de la chaîne de résultat sans considération des limites.

La taille du KEYMAT requis DOIT être plus longue de quatre octets que ce qui est nécessaire pour la clé AES associée. Le matériel de clé est utilisé comme suit :

AES-CTR avec une clé de 128 bits
Le KEYMAT requis pour chaque clé AES-CTR fait 20 octets. Les 16 premiers octets sont la clé AES de 128 bits, et les quatre octets restants sont utilisés comme valeur du nom occasionnel dans le bloc compteur.

AES-CTR avec clé de 192 bits
Le KEYMAT requis pour chaque clé AES-CTR fait 28 octets. Les 24 premiers octets sont la clé AES de 192 bits, et les quatre octets restants sont utilisés comme valeur de nom occasionnel dans le bloc compteur.

AES-CTR avec clé de 256 bits
Le KEYMAT requis pour chaque clé AES-CTR fait 36 octets. Les 32 premiers octets sont la clé AES de 256 bits, et les quatre octets restants sont utilisés comme valeur de nom occasionnel dans le bloc compteur.

5.2 Identifiant de phase 1

Le présent document ne spécifie pas les conventions pour utiliser AES-CTR pour les négociations IKE phase 1. Pour qu'AES-CTR soit utilisé de cette manière, une spécification distincte est nécessaire, et un identifiant d'algorithme de chiffrement devra être alloué.

5.3 Identifiant de phase 2

Pour la négociation IKE phase 2, l'IANA a alloué un identifiant de transformation ESP de 13 pour AES-CTR avec un IV explicite.

5.4 Attribut Longueur de clé

Comme AES prend en charge trois longueurs de clé, l'attribut Longueur de clé DOIT être spécifié dans l'échange IKE phase 2 [RFC2407]. L'attribut Longueur de clé DOIT avoir une valeur de 128, 192, ou 256.

6. Vecteurs d'essai

La présente section contient neuf vecteurs d'essai, qui peuvent être utilisés pour confirmer qu'une mise en œuvre applique correctement AES-CTR. Les trois premiers vecteurs d'essai utilisent AES avec une clé de 128 bits ; les trois vecteurs d'essai suivants utilisent AES avec une clé de 192 bits ; et les trois derniers vecteurs d'essai utilisent AES avec une clé de 256 bits

Vecteur d'essai n° 1 : Chiffrer 16 octets en utilisant AES-CTR avec une clé de 128 bits
Clé AES : AE 68 52 F8 12 10 67 CC 4B F7 A5 76 55 77 F3 9E
IV AES-CTR : 00 00 00 00 00 00 00 00
Nom occasionnel : 00 00 00 30
Chaîne de texte en clair : 'Single block msg'
Texte en clair *(hex)* : 53 69 6E 67 6C 65 20 62 6C 6F 63 6B 20 6D 73 67
Bloc compteur (1) : 00 00 00 30 00 00 00 00 00 00 00 00 00 00 00 01
Flux de clé (1) : B7 60 33 28 DB C2 93 1B 41 0E 16 C8 06 7E 62 DF
Texte chiffré : E4 09 5D 4F B7 A7 B3 79 2D 61 75 A3 26 13 11 B8

Vecteur d'essai n° 2 : Chiffrer 32 octets en utilisant AES-CTR avec une clé de 128 bits
Clé AES : 7E 24 06 78 17 FA E0 D7 43 D6 CE 1F 32 53 91 63
IV AES-CTR : C0 54 3B 59 DA 48 D9 0B
Nom occasionnel : 00 6C B6 DB
Texte en clair : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
Bloc compteur (1) : 00 6C B6 DB C0 54 3B 59 DA 48 D9 0B 00 00 00 01
Flux de clé (1) : 51 05 A3 05 12 8F 74 DE 71 04 4B E5 82 D7 DD 87
Bloc compteur (2) : 00 6C B6 DB C0 54 3B 59 DA 48 D9 0B 00 00 00 02
Flux de clé (2) : FB 3F 0C EF 52 CF 41 DF E4 FF 2A C4 8D 5C A0 37
Texte chiffré : 51 04 A1 06 16 8A 72 D9 79 0D 41 EE 8E DA D3 88 EB 2E 1E FC 46 DA 57 C8 FC E6 30 DF 91 41 BE 28

Vecteur d'essai n° 3 : Chiffrer 36 octets en utilisant AES-CTR avec une clé de 128 bits
Clé AES : 76 91 BE 03 5E 50 20 A8 AC 6E 61 85 29 F9 A0 DC
IV AES-CTR : 27 77 7F 3F 4A 17 86 F0

Nom occasionnel : 00 E0 01 7B
Texte en clair : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23
Bloc compteur (1) : 00 E0 01 7B 27 77 7F 3F 4A 17 86 F0 00 00 00 01
Flux de clé (1) : C1 CE 4A AB 9B 2A FB DE C7 4F 58 E2 E3 D6 7C D8
Bloc compteur (2) : 00 E0 01 7B 27 77 7F 3F 4A 17 86 F0 00 00 00 02
Flux de clé (2) : 55 51 B6 38 CA 78 6E 21 CD 83 46 F1 B2 EE 0E 4C
Bloc compteur (3) : 00 E0 01 7B 27 77 7F 3F 4A 17 86 F0 00 00 00 03
Flux de clé (3) : 05 93 25 0C 17 55 36 00 A6 3D FE CF 56 23 87 E9
Texte chiffré : C1 CF 48 A8 9F 2F FD D9 CF 46 52 E9 EF DB 72 D7 45 40 A4 2B DE 6D 78 36 D5 9A 5C EA AE F3 10 53 25 B2 07 2F

Vecteur d'essai n° 4 : Chiffrer 16 octets en utilisant AES-CTR avec une clé de 192 bits
Clé AES : 16 AF 5B 14 5F C9 F5 79 C1 75 F9 3E 3B FB 0E ED : 86 3D 06 CC FD B7 85 15
IV AES-CTR IV : 36 73 3C 14 7D 6D 93 CB
Nom occasionnel : 00 00 00 48
Chaîne de texte en clair : 'Single block msg'
Texte en clair : 53 69 6E 67 6C 65 20 62 6C 6F 63 6B 20 6D 73 67
Bloc compteur (1) : 00 00 00 48 36 73 3C 14 7D 6D 93 CB 00 00 00 01
Flux de clé (1) : 18 3C 56 28 8E 3C E9 AA 22 16 56 CB 23 A6 9A 4F
Texte chiffré : 4B 55 38 4F E2 59 C9 C8 4E 79 35 A0 03 CB E9 28

Vecteur d'essai n° 5 : Chiffrer 32 octets en utilisant AES-CTR avec une clé de 192 bits
Clé AES : 7C 5C B2 40 1B 3D C3 3C 19 E7 34 08 19 E0 F6 9C : 67 8C 3D B8 E6 F6 A9 1A
IV AES-CTR : 02 0C 6E AD C2 CB 50 0D
Nom occasionnel : 00 96 B0 3B
Texte en clair : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
Bloc compteur (1) : 00 96 B0 3B 02 0C 6E AD C2 CB 50 0D 00 00 00 01
Flux de clé (1) : 45 33 41 FF 64 9E 25 35 76 D6 A0 F1 7D 3C C3 90
Bloc compteur(2) : 00 96 B0 3B 02 0C 6E AD C2 CB 50 0D 00 00 00 02
Flux de clé (2) : 94 81 62 0F 4E C1 B1 8B E4 06 FA E4 5E E9 E5 1F
Texte chiffré : 45 32 43 FC 60 9B 23 32 7E DF AA FA 71 31 CD 9F 84 90 70 1C 5A D4 A7 9C FC 1F E0 FF 42 F4 FB 00

Vecteur d'essai n° 6 : Chiffrer 36 octets en utilisant AES-CTR avec une clé de 192 bits
Clé AES : 02 BF 39 1E E8 EC B1 59 B9 59 61 7B 09 65 27 9B : F5 9B 60 A7 86 D3 E0 FE
IV AES-CTR : 5C BD 60 27 8D CC 09 12
Nom occasionnel : 00 07 BD FD
Texte en clair : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23
Bloc compteur (1) : 00 07 BD FD 5C BD 60 27 8D CC 09 12 00 00 00 01
Flux de clé (1) : 96 88 3D C6 5A 59 74 28 5C 02 77 DA D1 FA E9 57
Bloc compteur (2) : 00 07 BD FD 5C BD 60 27 8D CC 09 12 00 00 00 02
Flux de clé (2) : C2 99 AE 86 D2 84 73 9F 5D 2F D2 0A 7A 32 3F 97
Bloc compteur (3) : 00 07 BD FD 5C BD 60 27 8D CC 09 12 00 00 00 03
Flux de clé (3) : 8B CF 2B 16 39 99 B2 26 15 B4 9C D4 FE 57 39 98
Texte chiffré : 96 89 3F C5 5E 5C 72 2F 54 0B 7D D1 DD F7 E7 58 D2 88 BC 95 C6 91 65 88 45 36 C8 11 66 2F 21 88 AB EE 09 35

Vecteur d'essai n° 7 : Chiffrer 16 octets en utilisant AES-CTR avec une clé de 256 bits
Clé AES : 77 6B EF F2 85 1D B0 6F 4C 8A 05 42 C8 69 6F 6C 6A 81 AF 1E EC 96 B4 D3 7F C1 D6 89 E6 C1 C1 04AES-CTR IV DB 56 72 C9 7A A8 F0 B2
Nom occasionnel : 00 00 00 60
Chaîne de texte en clair : 'Single block msg'
Texte en clair : 53 69 6E 67 6C 65 20 62 6C 6F 63 6B 20 6D 73 67
Bloc compteur (1): 00 00 00 60 DB 56 72 C9 7A A8 F0 B2 00 00 00 01
Flux de clé (1): 47 33 BE 7A D3 E7 6E A5 3A 67 00 B7 51 8E 93 A7
Texte chiffré : 14 5A D0 1D BF 82 4E C7 56 08 63 DC 71 E3 E0 C0

Vecteur d'essai n° 8 : Chiffrer 32 octets en utilisant AES-CTR avec une clé de 256 bits
Clé AES : F6 D6 6D 6B D5 2D 59 BB 07 96 36 58 79 EF F8 86 : C6 6D D5 1A 5B 6A 99 74 4B 50 59 0C 87 A2 38 84
IV AES-CTR : C1 58 5E F1 5A 43 D8 75
Nom occasionnel : 00 FA AC 24

Texte en clair : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
Bloc compteur (1) : 00 FA AC 24 C1 58 5E F1 5A 43 D8 75 00 00 00 01
Flux de clé (1) : F0 5F 21 18 3C 91 67 2B 41 E7 0A 00 8C 43 BC A6
Bloc compteur (2) : 00 FA AC 24 C1 58 5E F1 5A 43 D8 75 00 00 00 02
Flux de clé (2) : A8 21 79 43 9B 96 8B 7D 4D 29 99 06 8F 59 B1 03
Texte chiffré : F0 5E 23 1B 38 94 61 2C 49 EE 00 0B 80 4E B2 A9 B8 30 6B 50 8F 83 9D 6A 55 30 83 1D 93 44 AF 1C

Vecteur d'essai n° 9: Chiffrer 36 octets en utilisant AES-CTR avec une clé de 256 bits
Clé AES : FF 7A 61 7C E6 91 48 E4 F1 72 6E 2F 43 58 1D E2 AA 62 D9 F8 05 53 2E DF F1 EE D6 87 FB 54 15 3D
IV AES-CTR : 51 A5 1D 70 A1 C1 11 48
Nom occasionnel : 00 1C C5 B7
Texte en clair : 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23
Bloc compteur (1) : 00 1C C5 B7 51 A5 1D 70 A1 C1 11 48 00 00 00 01
Flux de clé (1) : EB 6D 50 81 19 0E BD F0 C6 7C 9E 4D 26 C7 41 A5
Bloc compteur (2) : 00 1C C5 B7 51 A5 1D 70 A1 C1 11 48 00 00 00 02
Flux de clé (2) : A4 16 CD 95 71 7C EB 10 EC 95 DA AE 9F CB 19 00
Bloc compteur (3) : 00 1C C5 B7 51 A5 1D 70 A1 C1 11 48 00 00 00 03
Flux de clé (3) : 3E E1 C4 9B C6 B9 CA 21 3F 6E E2 71 D0 A9 33 39
Texte chiffré : EB 6C 52 82 1D 0B BB F7 CE 75 94 46 2A CA 4F AA B4 07 DF 86 65 69 FD 07 F4 8C C0 B5 83 D6 07 1F 1E C0 E6 B8

7. Considérations pour la sécurité

Lorsque il est utilisé correctement, le mode AES-CTR assure une forte confidentialité. Bellare, Desai, Jokipii, Rogaway ont montré dans [BDJR] que les garanties de confidentialité fournies par le mode compteur sont au moins aussi fortes que celles du mode CBC lorsque on utilise le même chiffrement de bloc.

Malheureusement, il est très facile de mésuser de ce mode compteur. Si les valeurs de bloc compteur sont utilisées pour plus d'un paquet avec la même clé, le même flux de clés sera alors utilisé pour chiffrer les deux paquets, et la garantie de confidentialité disparait.

Qu'arrive t-il si le chiffreur OUX le même flux de clés avec deux textes en clair différents ? Supposons que deux séquences d'octets de texte en clair P1, P2, P3 et Q1, Q2, Q3 sont toutes deux chiffrées avec les flux de clés K1, K2, K3. Les deux textes chiffrés correspondants sont :

(P1 OUX K1), (P2 OUX K2), (P3 OUX K3)
(Q1 OUX K1), (Q2 OUX K2), (Q3 OUX K3)

Si ces deux flux de texte chiffré sont exposés à in attaquant, il en résulte alors une catastrophique défaillance de confidentialité, car :

(P1 OUX K1) OUX (Q1 OUX K1) = P1 OUX Q1
(P2 OUX K2) OUX (Q2 OUX K2) = P2 OUX Q2
(P3 OUX K3) OUX (Q3 OUX K3) = P3 OUX Q3

Une fois que l'attaquant a obtenu les deux textes en clair combinés ensemble par l'opération OUX, il est relativement facile de les séparer. Donc, utiliser n'importe quel chiffrement de flux, y compris AES-CTR, pour chiffrer des textes en clair sous le même flux de clés divulgue le texte en clair.

Donc, les chiffrements de flux, y compris AES-CTR, ne devraient pas être utilisés avec des clés statiques. Il est inapproprié d'utiliser AES-CTR avec des clés statiques. Des mesures extraordinaires seraient nécessaires pour empêcher la réutilisation d'une valeur de bloc compteur avec la clé statique au fil des cycles d'alimentation. Pour être sûres, les mises en œuvre d'ESP DOIVENT utiliser des clés fraîches avec AES-CTR. Le protocole d'échange de clés Internet (IKE) [RFC2409] peut être utilisé pour établir des clés fraîches. IKE peut aussi être utilisé pour établir le nom occasionnel au début de l'association de sécurité.

Lorsque IKE est utilisé pour établir des clés fraîches entre deux entités homologues, des clés séparées sont établies pour les deux flux de trafic. Lorsque un mécanisme autre que IKE est utilisé pour établir des clés fraîches, et que ce mécanisme établit seulement une clé pour chiffrer les paquets, il est alors fortement probable que les homologues vont choisir les mêmes valeurs d'IV pour certains paquets. Donc, pour éviter des collisions de bloc compteur, les mises en œuvre ESP qui

permettent d'utilisation de la même clé pour chiffrer le trafic sortant et déchiffrer le trafic entrant avec le même homologue DOIVENT s'assurer que les deux homologues allouent des valeur de nom occasionnel différentes aux associations de sécurité.

La falsification des données est triviale avec le mode CTR. La démonstration de cette attaque est similaire à la discussion de la réutilisation de flux de clés données ci-dessus. Si une séquence d'octets de texte en clair connue P1, P2, P3 est chiffrée avec le flux de clés K1, K2, K3, l'attaquant peut alors remplacer le texte clair par un texte de son choix. Le texte chiffré est :

(P1 OUX K1), (P2 OUX K2), (P3 OUX K3)

L'attaquant combine simplement par l'opération OUX une séquence Q1, Q2, Q3 choisie avec le texte chiffré pour obtenir :

(Q1 OUX (P1 OUX K1)), (Q2 OUX (P2 OUX K2)), (Q3 OUX (P3 OUX K3))

Qui est la même chose que :

((Q1 OUX P1) OUX K1), ((Q2 OUX P2) OUX K2), ((Q3 OUX P3) OUX K3)

Le déchiffrement du texte chiffré généré par l'attaquant va donner exactement ce que voulait l'attaquant :

(Q1 OUX P1), (Q2 OUX P2), (Q3 OUX P3)

En conséquence les mises en œuvre de ESP DOIVENT utiliser AES-CTR en conjonction avec l'authentification ESP.

De plus, comme AES a une taille de bloc de 128 bits, sans considération du mode employé, le texte chiffré généré par le chiffrement AES devient distinguable des valeurs aléatoires après que 2^64 blocs ont été chiffrés avec une seule clé. Comme ESP avec des numéros de séquence améliorés permet jusqu'à 2^64 paquets dans une seule association de sécurité, il y a un réel potentiel pour que plus de 2^64 blocs soient chiffrés avec une clé. Donc, les mises en œuvre DEVRAIENT générer une clé fraîche avant que 2^64 blocs soient chiffrés avec la même clé. Noter que ESP avec des numéros de séquence de 32 bits ne va pas excéder 2^64 blocs même si tous les paquets sont de la longueur maximum de jumbogrammes IPv6 [RFC2675].

Il y a des attaques de précalcul très génériques contre tous les modes de chiffrement de bloc qui permettent une attaque par interposition contre la clé. Ces attaques exigent la création et la recherche dans d'énormes tableaux de texte chiffré associés à des textes en clair connus et des clés connues. En supposant que les ressources en mémoire et en capacité de calcul sont disponibles pour une attaque de précalcul, la force théorique de AES-CTR (et de tout autre mode de chiffrement de bloc) est limitée à 2^(n/2) bits, où n est le nombre de bits de la clé. L'utilisation de longues clés est la meilleure contre mesure contre les attaques de précalcul. Donc, les mises en œuvre qui emploient des clés AES de 128 bits devraient prendre des précautions pour rendre les attaques de précalcul plus difficiles. La valeur imprévisible du nom occasionnel dans le boc compteur augmente de façon significative la taille du tableau que l'attaquant doit calculer pour monter une attaque ayant des chances de réussir.

8. Motifs de la conception

Dans le cours du développement de la présente spécification, l'utilisation du champ Numéro de séquence ESP plutôt qu'un champ IV explicite a été examinée. Ce choix n'est pas un problème de sécurité cryptographique, car l'une et l'autre approches vont empêcher les collisions de bloc compteur.

Dans un modèle très prudent de sécurité du chiffrement, au plus 2^64 blocs devraient être chiffrés avec AES-CTR sous une seule clé. Sous cette contrainte, pas plus de 64 bits ne sont nécessaires pour identifier chaque paquet au sein d'une association de sécurité. Comme le numéro de séquence ESP étendu est de 64 bits, il est un candidat évident comme IV implicite. Cela impliquerait une seule méthode pour l'allocation de valeurs par paquet dans le bloc compteur. L'utilisation d'un IV explicite n'implique pas une telle méthode, qui est souhaitable pour plusieurs raisons.

1. Seul le chiffreur peut assurer que la valeur n'est pas utilisée pour plus d'un paquet, de sorte qu'il n'y a pas d'avantage à choisir un mécanisme qui permet au déchiffreur de déterminer si les valeurs de bloc compteur entrent en collision. Le dommage résultant de la collision est fait, que le déchiffreur la détecte ou non.

2. Elle permet les ajouts, les LFSR, et toutes les autres technique qui satisfont le budget temps du chiffreur, pour autant que la technique résulte en une valeur unique pour chaque paquet. Les ajouts sont simples et directs à mettre en œuvre,

mais du fait des reports, ils ne s'exécutent pas dans un délai constant. Les LFSR offrent une solution de remplacement qui s'exécute en durée constante.

3. La complexité est sous le contrôle de la mise en œuvre. De plus, la décision prise par la mise en œuvre de chiffreur ne rend pas le déchiffreur plus (ou moins) complexe.

4. Lorsque le chiffreur a plus d'un appareil de matériel cryptographique, un préfixe d'IV est alloué à chaque appareil, assurant que des collisions ne vont pas se produire. Donc, comme le déchiffreur n'a pas besoin d'examiner la structure de l'IV, il n'est pas affecté par la structure d'IV choisie par le chiffreur. On ne peut pas faire usage de la même technique avec les numéros de séquence ESP, parce que leur sémantique exige la génération de valeurs séquentielles.

5. Les limites d'assurance sont très importantes pour les mises en œuvre qui seront évaluées par rapport au normes FIPS Pub 140-1 ou FIPS Pub 140-2 [SECRQMTS]. L'allocation de la valeur de bloc compteur par paquet doit être comprise dans les limites d'assurance. Certaines mises en œuvre allouent le numéro de séquence dans les limites d'assurance, mais d'autres ne le font pas. Une collision de numéro de séquence n'a pas de conséquences sévères, mais, comme décrit à la section 6, une collision des valeurs de bloc compteur a des conséquences désastreuses.

6. Le couplage avec le numéro de séquence est possible dans les architectures où l'allocation de numéro de séquence est effectuée au sein des limites d'assurance. Dans cette situation, le numéro de séquence et le champ IV vont contenir la même valeur.

7. Le découplage du numéro de séquence est possible dans les architectures où l'allocation de numéro de séquence est effectuée en dehors des limites d'assurance.

L'utilisation d'un champ IV explicite directement suivi par le découplage du numéro de séquence et la valeur du bloc compteur par paquet. La redondance associée à 64 bits pour le champ IV est acceptable. Cette redondance est significativement moindre que celle associée au mode de chaînage de bloc de chiffrement (CBC, *Cipher Block Chaining*). Employé normalement, CBC requiert un bloc complet pour le IV et, en moyenne, la moitié d'un bloc pour le bourrage. AES-CTR avec un IV explicite a environ un tiers de la redondance de AES-CBC, et la redondance est constante pour chaque paquet.

L'inclusion du nom occasionnel fournit une contre-mesure faible contre les attaques de précalcul. Pour que cette contre mesure soit efficace, l'attaquant ne doit pas être capable de prédire la valeur du nom occasionnel avant l'établissement de l'association de sécurité. L'utilisation d clés longues donne une contre mesure forte aux attaques de précalcul, et AES offre des tailles de clé qui étouffent ces attaques pour encore de nombreuses décennies.

Une valeur de bloc compteur de 28 bits est suffisante pour la génération d'un flux de clés pour chiffrer le plus grand jumbogramme IPv6 possible [RFC2675] ; cependant, un champ de 32 bits est utilisé. Cette taille est pratique aussi bien pour les mises en œuvre de matériels que de logiciels.

9. Considérations relatives à l'IANA

L'IANA a alloué le numéro 13 à la transformation ESP pour AES-CTR avec un IV explicite.

10. Déclaration de droits de propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourrait être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à http://www.ietf.org/ipr .

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres 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.

11. Remerciements

Le présent document est le résultat de longues discussions et de compromis. Bien que tous les participants ne soient pas complètement satisfaits de son résultat, le document est meilleur grâce à leurs contributions.

L'auteur tient à remercier les membres du groupe de travail IPsec pour leurs contributions à sa conception, avec une mention particulière pour les efforts de (par ordre alphabétique) Steve Bellovin, David Black, Niels Ferguson, Charlie Kaufman, Steve Kent, Tero Kivinen, Paul Koning, David McGrew, Robert Moskowitz, Jesse Walker, et Doug Whiting.

L'auteur remercie Alireza Hodjat, John Viega, et Doug Whiting de leur assistance sur les vecteurs d'essai.

12. Références

Cette section donne les références normatives et pour information.

12.1 Références normatives

[AES] NIST, FIPS PUB 197, "Advanced Encryption Standard (AES)", novembre 2001.

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

[RFC2407] D. Piper, "Le domaine d'interprétation de sécurité IP de l'Internet pour ISAKMP", novembre 1998. *(Obsolète, voir 4306)*

[RFC2406] S. Kent et R. Atkinson, "Encapsulation de charge utile de sécurité IP (ESP)", novembre 1998. *(Obsolète, voir* RFC4303)

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

12.2 Références pour information

[BDJR] Bellare, M, Desai, A., Jokipii, E. and P. Rogaway, "A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation", Proceedings 38th Annual Symposium on Foundations of Computer Science, 1997.

[RFC2401] S. Kent et R. Atkinson, "Architecture de sécurité pour le protocole Internet", novembre 1998. (*Obsolète, voir* RFC4301)

[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*)*

[RFC2411] R. Thayer, N. Doraswamy, R. Glenn, "Feuille de route pour la sécurité sur IP", novembre 1998. *(Remplacée par* RFC6071*)*

[RFC2675] D. Borman, S. Deering, R. Hinden, "Jumbogrammes IPv6", août 1999. *(P.S.)*

[SECRQMTS] National Institute of Standards and Technology. FIPS Pub 140-1: "Security Requirements for Cryptographic Modules". 11 janvier 1994.
National Institute of Standards and Technology. FIPS Pub 140-2: "Security Requirements for Cryptographic Modules". 25 mai 2001. [Remplace FIPS Pub 140-1]

13. Adresse de l'auteur

Russell Housley
Vigil Security, LLC
918 Spring Knoll Drive
Herndon, VA 20170
USA

mél : housley@vigilsec.com

14. Déclaration complète de droits de reproduction

Remerciement

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