L'ANALYSE DE L'ALGORITHME AES - MODE CWC DE CHIFFRAGE

Cucuetu Irina, Grinico Radu

Université Technique de Moldova

Résumé : *Les algorithmes de chiffrement symétrique se fondent sur une même clé pour chiffrer et déchiffrer un message. Algorithmes de chiffrement symétrique très utilisés : Chiffre de Wegman, DES, 3DES, AES, RC4, RC5, MISTY1 et d'autres. Le Carter-Wegman + compteur à double usage (mode CWC) est un mode de fonctionnement pour assurer le chiffrement et l'intégrité des messages. Cette mode est parallélisable, rapide à la fois logiciel et matériel. Il peut 10Gbits/second processus en utilisant la technologie ASIC. Il est une simple combinaison de techniques bien connues, est bien adapté pour les environnements de mémoire limitées, est conçu pour promouvoir l'interopérabilité. Autrement dit, il est un ensemble minimal de paramètres. CWC est actuellement le seul mode comme ayant les cinq caractéristiques suivantes : sécurité prouvée, parallélisme, de haute performance dans le matériel, en logiciels de haute performance, et aucun problème de propriété intellectuelle.*

Mots clé : *Cryptographie, CWC, AES, HASH, ASIC, clé secrète.*

Introduction

La cryptographie est une des disciplines de la cryptologie s'attachant à protéger des messages (assurant confidentialité, authenticité et intégrité) en s'aidant souvent de secrets ou clés.

Algorithmes de cryptographie symétrique (à clé secrète)

Les algorithmes de chiffrement symétrique se fondent sur une même clé pour chiffrer et déchiffrer un message. L'un des problèmes de cette technique est que la clé, qui doit rester totalement confidentielle, doit être transmise au correspondant de façon sûre.

Quelques algorithmes de chiffrement symétrique très utilisés :

Chiffre de Wegman(le seul offrant une sécurité théorique absolue, à condition que la clé ait au moins la même longueur que le message, qu'elle ne soit utilisée qu'une seule fois à chiffrer et qu'elle soit totalement aléatoire), DES, 3DES, AES, RC4, RC5, MISTY1 et d'autres.

Description de l'algorithme CWC

Le Carter-Wegman + compteur à double usage (mode CWC) est un mode de fonctionnement pour assurer le chiffrement et l'intégrité des messages. Cette mode est parallélisable, rapide à la fois logiciel et matériel, sans être entravées par brevets et attestée attacher à une bonne tenue sous l'hypothèse que le chiffrement par bloc sous-jacent est une permutation pseudo aléatoire.

Cette construction présente les avantages suivants :

1. CWC est rapide dans les logiciels et le matériel. En matériel, il peut 10Gbits/second processus en utilisant la technologie ASIC.
2. CWC est parallélisable à un degré quelconque, tout en maintenant compléter l'interopérabilité.
3. CWC est une simple combinaison de techniques bien connues.
4. CWC n'exige que le cryptage AES. Par exemple, les matérielles implémentations ne doivent mettre en place un module de chiffrement. Implémentations logicielles pouvez également éviter la surcharge de la mémoire de AES tableau 4 K de recherche décryptage.
5. CWC peut rejeter faux messages sans déchiffrer le texte chiffré.
6. CWC a expansion minimale (chaque cryptogramme est aussi longue que le en clair, avec addition d'une authentification de message).

7. CWC est dégagée des brevets au meilleur de notre connaissance.

8. CWC peut authentifier les messages chiffrés et associés donnés de texte en clair comme en-têtes.

Alpha

9. CWC est bien adapté pour les environnements de mémoire limitées.

10. CWC est conçu pour promouvoir l'interopérabilité. Autrement dit, il est un ensemble minimal de paramètres.

L'algorithme

CWC prend deux paramètres :

Y, la longueur de clé AES à utiliser en octets (16, 24 ou 32).

Z, la taille de l'étiquette d'authentification de message. C'est à Z détermine le nombre d'octets sont ajoutés à la fin du message à des fins de détection de modification. Z DOIT être un même nombre compris entre 4 et 16.

L'opération CWC-ENCRYPT

CWC-ENCRYPT prend les entrées suivantes :

K, une clé qui est Y octets de longueur.

A, une chaîne de longueur arbitraire constitué de données à authentifiée, mais pas crypté. La longueur de A ne doit pas dépasser 2 ^ 36 à 16 octets.

M, une chaîne de longueur arbitraire composé de texte en clair message. Ce message sera à la fois chiffré et authentifié. La longueur de M doit pas dépasser 2 ^ 36 à 16 octets.

N, un nonce, 11 octets de long. Chaque valeur de N NE DOIT PAS être utilisée plus d'une fois pour toute donnée clé K. La disposition du nonce est indéterminée, mais nous vous recommandons d'utiliser une partie du nonce pour un sel d'au moins 4 octets qui est choisie aléatoirement lors de l'installation clé temps et en utilisant le reste pour un compteur de messages.

CWC-ENCRYPT est calculé comme suit :

1) C = CWC-CTR (K, N, M)
2) T = CWC-MAC (K, A, N, C)
3) OUTPUT = C || T

(Voir l'implémentation en C# Fig.1)

L'opération CWC-DECRYPT

CWC-DECRYPT prend les entrées suivantes :

C, une chaîne de longueur arbitraire jusqu'à 2 ^ + Z octets 36 à 16, consistant en un texte chiffré à décrypter et authentifié.

Notez que si A ou C est plus longue que celle spécifiée ci-dessus, l'authentification échouera, car aucun message ne peut être que long. CWC-DECRYPT est calculé comme suit :

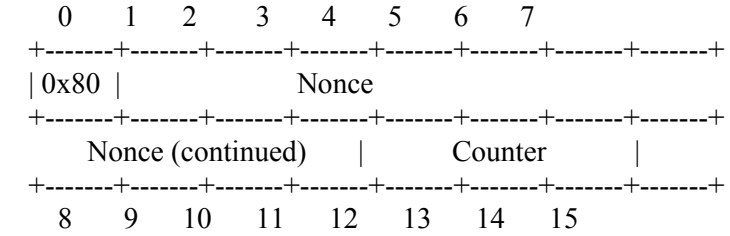
1) IF LEN(C) < Z THEN FAIL
2) C' = C [0 : LEN(C)-Z]
3) T' = C [LEN(C)-Z : LEN(C)]
4) T = CWC-MAC (K, A, N, C')
6) IF T <> T' THEN FAIL
7) OUTPUT = CWC-CTR (K, N, C')

(Voir l'implémentation en C# Fig.2)

L'opération CWC-CTR

Les deux premiers bits du bloc en cours de cryptage sont utilisés pour distinguer les différents types de chiffrement AES.

Voici une représentation visuelle des octets dans le compteur clair blocs :



CWC-CTR (K, N, M):
1) J = CEILING (LEN (M)/16)
2) S = ""
3) FOR I in 1 TO J: S = S || AES_K (0x80 || N || I)
4) OUTPUT = S [0: LEN (M)] XOR M
(Voir l'implémentation en C# Fig.3)

L'OPÉRATION CWC-MAC

L'opération CWC-MAC prend les résultats de CWC-HASH, puis effectue deux post-traitements des opérations AES. La première opération directement crypte le résultat de hachage.

CWC-MAC (K, A, N, C) :
1) R = AES_K (CWC-HASH (K, A, C))
2) OUTPUT = (AES_K (0x80||N||0x00000000) XOR R) [0 : Z]
(Voir l'implémentation en C# Fig.4)

L'OPÉRATION CWC-HASH

Cette fonction de hachage est le polynôme de hachage traditionnel Carter-Wegman, avec un champ de GF (2 ^ 127-1). Dans la pratique, cela signifie que, pour chaque 12 octets du message (si ces octets sont traités comme un nombre en notation « big endian »), on ajoute le bloc de message à l' résultat en cours, modulo 2 ^ 127-1. Nous multiplions ensuite par la touche dièse (lui-même traité comme un numéro), encore une fois modulo 2 ^ 127-1.

Le résultat est le résultat en cours, représenté comme une grosse 16-octet valeur « endian ». Le bit le plus significatif sera toujours 0.

CWC-HASH (K, A, C) :
1) Z = AES_K (0xC0000000000000000000000000000000) &
0x7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
2) X = CWC-HPAD (A) || CWC-HPAD(C)
3) B = LEN(X) / 12
4) OUTPUT = 0
5) FOR I FROM 0 TO B-1:
a) OUTPUT = OUTPUT + X [12*I: 12*I+12] MOD 2^127-1
b) OUTPUT = OUTPUT * Z MOD 2^127-1
6) OUTPUT = OUTPUT + (LEN(C) + 2^64*LEN (A)) MOD 2^127-1
(Voir l'implémentation en C# Fig.5)

L'OPÉRATION CWC-HPAD

L'entrée à la fonction de hachage doit être un multiple de 96 bits.

CWC-HPAD(STR) :
1) OUTPUT = STR
2) WHILE LEN (OUTPUT) MOD 12 : OUTPUT = OUTPUT + 0x00
(Voir l'implémentation en C# Fig.6, 7)

CONCLUSION

Le chiper mode CWC est un mode de chiffrement par bloc nouveau de fonctionnement pour la protection de la vie privée et à la fois l'authenticité des données encapsulées. CWC est actuellement le seul

mode comme ayant les cinq caractéristiques suivantes : sécurité prouvée, parallélisme, de haute performance dans le matériel, en logiciels de haute performance, et aucun problème de propriété intellectuelle.

Après que nous avons mis en œuvre l'algorithme en C #, suivi de sa phase de test et amélioration du code selon les spécifications du langage. À la suite de tests, nous avons remarqué que crypter un fichier volumineux prend plus de temps que la documentation de l'algorithme spécifie. La cause de cela est que le code est écrit purement en C #, qui est un langage de haut niveau, ce qui conduit à une augmentation du temps d'exécution.

BIBLIOGRAPHIE

1. *C# in a Nutshell*, Joseph Albahari and Ben Albahari, January 2010
2. *CWC : A high-performance conventional authenticated encryption mode*, Tadayoshi Kohno, John Viega, Doug Whiting, January 15, 2004
3. http://tools.ietf.org/html/draft-irtf-cfrg-cwc-00
4. http://etutorials.org/Programming/secure+programming/Chapter+5.+Symmetric+Encryption/5.10+Using+CWC+Mode/
5. http://eprint.iacr.org/2003/106
6. http://www.cryptage.org/cle-secrete.html
7. http://gladman.plushost.co.uk/oldsite/AES/index.php
8. http://msdn.microsoft.com/en-us/magazine/cc164055.aspx#S1
9. http://msdn.microsoft.com/en-us/magazine/cc164846.aspx?code=true&level=root%2cAesLib&file=Aes.cs