Fiche 212 Niveau : 2 version 2 (11/01/2012) **Orange**

Chiffrement à clé secrète par blocs en mode Galois Counter

1 Présentation du problème, notations

Lorsqu'on dispose d'une primitive de chiffrement à clé secrète par blocs, on sait chiffrer des blocs de la taille de l'entrée des données, taille qui dans les primitives actuelles est en général de 128 bits. Que faire quand on doit chiffrer une masse de données qui ne se limite pas à un bloc ? Une solution très simple consiste à découper les données en blocs de la taille d'entrée puis de chiffrer chacun des blocs. Ce mode d'opération est appelé le mode Electronic Codebook (ECB). Ce mode n'est pas recommandé pour diverses raisons de sécurité, en particulier car on sait reconnaître sur les chiffrés si deux blocs clairs sont identiques. Le NIST définit plusieurs façon de faire et en particulier le mode Galois Counter (GCM). Le mode Galois Counter développé par D. McGrew et J. Viega, combine le mode counter, très rapide et très sûr, avec un contrôle d'intégrité également très rapide et très sûr.
Nous supposerons que nous disposons d'une primitive de chiffrement $\mathcal{E}$ ayant pour fonction de déchiffrement $\mathcal{D}$. Ainsi, si nous avons une clé K ainsi qu'un bloc d'entrée x, le chiffré de ce bloc est $y = \mathcal{E}(K, x)$ noté encore $y = \mathcal{E}_K(x)$. La fonction de déchiffrement appliquée avec la clé K à y redonne x, c'est-à-dire : $\mathcal{D}(K, y) = x$ ou encore $\mathcal{D}_K(y) = x$. Bien entendu les fonctions $\mathcal{E}$ et $\mathcal{D}$ sont publiques, seule la clé K est secrète, c'est-à-dire les fonctions $\mathcal{E}_K$ et $\mathcal{D}_K$.

2 Le fonctionnement général du mode Galois Counter

Bien que le mode GMC (Galois Counter Mode) accepte plusieurs tailles pour certains des paramètres qui interviennent, nous donnons ici les paramètres les plus conseillés (quand un choix est possible nous mettrons entre parenthèses qu'il s'agit d'une valeur conseillée). En outre ces paramètres conseillés sont parfois en pratique obligatoires pour éviter certaines attaques. (Voir l'article d'Antoine Joux : "Authentication Failures in NIST version of GCM"). On suppose que les données comportent : un texte clair P à chiffrer, dont on doit assurer l'intégrité et dont le nombre t de bits est tel que $0 \leq t \leq 2^{39} - 2^8$; des données additionnelles A à transmettre sans chiffrement, mais dont on doit aussi assurer l'intégrité, dont le nombre a de bits vérifie $0 \leq a \leq 2^{64}$.
On dispose d'un circuit de chiffrement par blocs (par exemple AES 128 bits) dont une clé de session K a été choisie. La taille des blocs chiffrés est 128 bits.
On veut récupérer en sortie : le chiffré C de P ayant la même taille que P ; les données additionnelles A ; un tag T qui assure l'intégrité de la chaîne $C||A$.

2.1 Le chiffrement

Pour le chiffré tout d'abord, on dispose d'une zone mémoire Y de 128 bits (taille du bloc à chiffrer) qu'on va faire évoluer à partir d'une valeur Y_0 en $Y_1, Y_2, \ldots$. Pour chaque nouveau message à chiffrer

avec une même clé K on choisit une nouvelle valeur initiale (non encore utilisée avec cette clé) IV de 96 bits (taille conseillée). Par ailleurs on a une valeur initiale de 32 bits :

$$Z_0 = 00\ldots01$$

On construit Y_0 par

$$Y_0 = IV||Z_0.$$

Par récurrence on construit Y_i en prenant

$$Y_i = IV||Z_i,$$

où

$$Z_i = Z_{i-1} + 1 \mod 2^{32}.$$

On notera $\mathrm{inc}(Y_{i-1})$ l'application qui fait passer de Y_{i-1} à Y_i. Le texte clair P étant découpé en $n-1$ blocs P_i de taille 128 bits et en un dernier bloc P_n^* de taille $0 < b \leq 128$ bits, on calcule les blocs de chiffré par :

$$C_i = P_i \oplus \mathcal{E}(K, Y_i) \quad i = 1, \ldots, n-1,$$
$$C_n^* = P_n^* \oplus MSB_b\left(\mathcal{E}(K, Y_n)\right),$$

où $\mathcal{E}(K, *)$ est la fonction de chiffrement d'un bloc avec la clé secrète K. On calcule tout d'abord le chiffré du bloc nul avec la clé K

$$H = \mathcal{E}(K, 0^{128}).$$

Pour la vérification de l'intégrité on calcule la marque T (que nous prendrons de taille 128 bits, valeur conseillée) de la manière suivante :

$$T = \mathrm{GHASH}(H, A, C) \oplus \mathcal{E}(K, Y_0),$$

où GHASH est une fonction que nous allons définir par la suite.
Remarquons que ce système avec une entrée P de taille 0 permet de ne faire que de l'intégrité sur A.

2.2 Le déchiffrement

Pour chaque nouveau message à déchiffrer avec une même clé K on récupère la valeur initiale IV qui a été utilisée au chiffrement. On utilise la valeur de 32 bits :

$$Z_0 = 00\ldots01$$

pour construire Y_0 tel que :

$$Y_0 = IV||Z_0.$$

Par récurrence on construit Y_i en prenant

$$Y_i = IV||Z_i,$$

où

$$Z_i = Z_{i-1} + 1 \mod 2^{32}.$$

On notera $\mathrm{inc}(Y_{i-1})$ l'application qui fait passer de Y_{i-1} à Y_i. On calcule tout d'abord le chiffré du bloc nul avec la clé K

$$H = \mathcal{E}(K, 0^{128}).$$

Pour la vérification de l'intégrité on calcule la marque T', de taille 128 bits, de la manière suivante :

$$T' = \mathrm{GHASH}(H, A, C) \oplus \mathcal{E}(K, Y_0),$$

et on compare avec la valeur T qui a été transmise. Si $T' \neq T$ on arrête là, le message n'est pas intègre. Sinon on déchiffre de la manière suivante. Le texte chiffré C étant découpé en $n-1$ blocs C_i de taille 128 bits et en un dernier bloc C_n^* de taille $0 < b \leq 128$ bits, on calcule les blocs de texte clair par :

$$P_i = C_i \oplus \mathcal{E}(K, Y_i) \quad i = 1, \ldots, n-1,$$
$$P_n^* = C_n^* \oplus MSB_b\left(\mathcal{E}(K, Y_n)\right),$$

où $\mathcal{E}(K, *)$ est la fonction de chiffrement d'un bloc avec la clé secrète K.

3 La fonction GHASH

3.1 Le corps fini à 2^{128} éléments

On prend comme polynôme minimal le polynôme

$$P(X) = X^{128} + X^7 + X^2 + X + 1.$$

Ce polynôme est irréductible sur $\mathbb{F}_2$. Le corps à 2^{128} éléments peut être représenté par

$$\mathbb{F}_{2^{128}} = \mathbb{F}_2[X]/\left(P(X)\right).$$

Autrement dit les éléments sont les polynômes à coefficients dans $\mathbb{F}_2$ de degré ≤ 127, l'addition est l'addition des polynômes et la multiplication est la multiplication des polynômes modulo $P(X)$. On notera α la classe du polynôme X dans ce passage au quotient. Tout élément $a \in \mathbb{F}_{2^{128}}$ s'écrit donc de manière unique sous la forme

$$a = \sum_{i=0}^{127} a_i \alpha^i.$$

En particulier l'élément α^{128} s'écrit

$$\alpha^{128} = 1 + \alpha + \alpha^2 + \alpha^7.$$

On notera $r = 1 + \alpha + \alpha^2 + \alpha^7$. Pour construire un algorithme efficace de multiplication nous allons commencer par la multiplication d'un élément a par α qu'on notera $m(a)$.

procédure $\mathbf{m(a)}$ {
si $a_{127} = 0$
renvoyer $\mathrm{shr}(a, 1)$;
sinon
$b = \mathrm{shr}(a, 1)$;
renvoyer $b \oplus r$;
finsi
}

Maintenant nous allons effectuer la multiplication $c = b.a$ où

$$a = \sum_{i=0}^{127} a_i \alpha^i, \quad b = \sum_{i=0}^{127} b_i \alpha^i.$$

Pour cela nous allons reconstituer b par l'algorithme de Hörner

procédure $\mathbf{mult}(\mathbf{b}, \mathbf{a})$ {
$c = 0$;
faire depuis $i = 0$ *jusqu'à* 127
$c = m(c)$;
si $b_{127-i} = 1$
$c = c \oplus a$;
finsi
finfaire
}

3.2 La fonction GHASH

Pour terminer la description complète du système Galois Counter Mode, il faut maintenant décrire la façon de calculer le tag T

$$T = \mathrm{GHASH}(H, A, C) \oplus \mathcal{E}(K, Y_0),$$

c'est-à-dire décrire la fonction $\mathrm{GHASH}(H, A, C)$. Pour cela on rappelle que n est le nombre de blocs de texte chiffré

$$C = C_1 C_2 \dots C_{n-1} C_n^*,$$

où les blocs P_i sont de taille 128 bits, le dernier bloc P_n^* étant de taille $0 < t \leq 128$; m est le nombre de blocs de données additionnelles

$$A = A_1 A_2 \dots A_{m-1} A_m^*,$$

où les blocs A_i sont de taille 128 bits, le dernier bloc A_m^* étant de taille $0 < t \leq 128$.
On définit alors par récurrence la suite finie $(X_i)_{i=0,\dots,m+n+1}$ par

$$X_i = \begin{cases} 0 & \text{si } i = 0 \\ (X_{i-1} \oplus A_i) \,.\, H & \text{si } i = 1, \dots, m-1 \\ (X_{m-1} \oplus (A_m^* || 0 \dots 0)) \,.\, H & \text{si } i = m \\ (X_{i-1} \oplus C_i) \,.\, H & \text{si } i = m+1, \dots, m+n-1 \\ \left(X_{m+n-1} \oplus (C_m^* || 0 \dots 0)\right) \,.\, H & \text{si } i = m+n \\ \left(X_{m+n} \oplus (\mathrm{taille}(A) || \mathrm{taille}(C))\right) \,.\, H & \text{si } i = m+n+1 \end{cases}$$

où la multiplication par H se fait dans le corps fini à 128 éléments et où taille(x) représente la taille en bits de l'argument x. On pose alors

$$\mathrm{GHASH}(H, A, C) = X_{m+n+1}.$$

4 Autres paramètres

Il n'est pas conseillé d'utiliser d'autres paramètres que ceux indiqués dans les sections précédentes, c'est-à-dire taille quelconque de IV et taille $s < 128$ bits pour taille de la marque T. Cependant le standard décrit par le NIST prévoit ces cas.
Si la taille de IV est différente de 96 bits alors on construit Y_0 par :

$$Y_0 = GHASH(H, \{\}, IV).$$

Si la taille s (en bits) de la marque est prévue telle que $s < 128$ alors on construit T ayant 128 bits comme indiqué dans la section précédente :

$$T = \mathrm{GHASH}(H, A, C) \oplus \mathcal{E}(K, Y_0),$$

et on ne garde que les s bits les plus significatifs :

$$T \leftarrow MSB_s(T).$$