Fiche 211 Niveau : 2 version 2 (11/01/2012) **Orange**

Chiffrement à clé secrète par blocs en Mode CBC

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 Cipher Block Chaining (CBC). C'est un des modes les plus utilisés, tout au moins jusqu'à présent, mais qui tend actuellement à être remplacé par le mode Galois Counter que nous verrons dans une autre fiche.
Notons un texte clair B sous la forme de blocks :

$$B = B_1 B_2 \cdots B_n.$$

Pour le moment nous ne détaillons pas la façon dont est constitué le dernier bloc qui est toujours un bloc contenant des données de "bourrage", même si le message initial avait pour longueur un multiple de la taille du bloc.
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 mode CBC

Le fonctionnement est le suivant (voir aussi NIST Special Publication 800-38A 2001 Edition ainsi que le RFC 3852) : le chiffrement du premier bloc, noté B_1, consiste à faire opérer $\mathcal{E}_K$ sur $(B_1 \oplus IV)$, où $\oplus$ représente l'opération « ou exclusif » et IV est un vecteur d'initialisation, c'est-à-dire une chaîne imprévisible de 128 bits. Le chiffrement des autres blocs s'effectue d'une manière similaire, le vecteur d'initialisation étant simplement remplacé par le chiffrement du blocs précédent. Ainsi, si on note $\mathcal{E}_K$ la fonction de chiffrement (avec une clé K), B_i le ième bloc du message, et C_i le chiffré du ième bloc, on a

$$C_1 = \mathcal{E}_K(B_1 \oplus IV),$$
$$C_i = \mathcal{E}_K(C_{i-1} \oplus B_i).$$

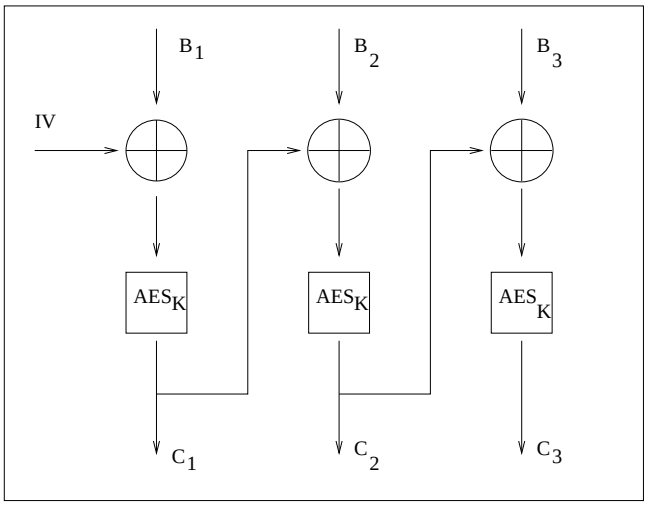

FIG. 1 – AES-CBC

Le déchiffrement se fait de la manière suivante. Soit $\mathcal{D}$ la fonction de déchiffrement, les blocs clairs B_i sont obtenus par les formules

$$B_i = C_{i-1} \oplus \mathcal{D}_K(C_i),$$
$$B_1 = IV \oplus \mathcal{D}_K(C_1).$$

Remarque 2.1 *Pour déchiffrer en mode CBC on utilise la fonction $\mathcal{D}$. Pour certaines primitives de chiffrement et pour certaines implémentations (AES par exemple) $\mathcal{D}$ s'exécute plus lentement que $\mathcal{E}$ Ceci tient à des valeurs des paramètres qui ont été optimisées pour $\mathcal{E}$ et qui ne le sont pas pour $\mathcal{D}$. De ce fait, cela introduit un avantage aux modes qui comme Galois counter n'utilisent que $\mathcal{E}$, que ce soit pour chiffrer ou pour déchiffrer.*

3 Découpage en blocs et bourrage

Le mode de chiffrement CBC nécessite de disposer en entrée d'un nombre exact de blocs (en gfénéral de blocs de 128 bits c'est-à-dire 16 octets). Soit donc m la longueur en octets du texte clair M. Soit $l = m \mod 16$. On a donc $m = k * 16 + l$ où $0 \leq l < 16$.
La transformation de M en un texte clair de taille un nombre exact de blocs suit les recommandations de RFC3852 (successeur de pkcs#7).
Le texte M est transformé en un texte M_1 de longueur $16 * (k + 1)$ octets. Les m premiers octets de M_1 sont ceux de M, les $16 - l$ derniers octets de M_1 contiennent tous la même valeur $16 - l$. Remarquons que si M a un nombre exact de blocs (cas $l = 0$) alors M_1 a un bloc de plus formé de 16 octets contenant tous 16 : il y a toujours un « padding » (le nombre d'octets ajoutés est toujours ≥ 1), ce qui assure qu'on peut toujours trouver le nombre d'octets ajoutés, dans le dernier octet de M_1.