CRYPTO 1 - TD 1
2024-2025

Énoncé

La notion naturelle de sécurité que doit garantir un système de chiffrement par bloc est la variante calculatoire de la sécurité parfaite : la vue d'un texte chiffré ne doit pas révéler d'information sur le texte clair. Cette notion est formalisée sous le nom de *sécurité sémantique* : étant donnés deux messages M_0 et M_1 choisis par l'adversaire et un chiffré C d'un message M_b pour $b \in \{0,1\}$, un adversaire ne peut pas obtenir la valeur du bit b avec une probabilité significativement meilleure que $1/2$. Nous distinguons généralement quatre types d'attaques différentes pour les systèmes de chiffrement par bloc :

- une attaque **à textes chiffrés connus** où l'attaquant dispose d'un ou plusieurs textes chiffrés ;
- une attaque **à textes clairs connus** où l'attaquant dispose d'un ou plusieurs textes chiffrés et des textes clairs correspondants ;
- une attaque **à textes clairs choisis** où l'attaquant peut obtenir le chiffrement de textes clairs de son choix ;
- une attaque **à textes chiffrés choisis** où l'attaquant peut également obtenir le déchiffrement de textes chiffrés de son choix ;

Exercice 1 : Modes opératoires et propriétés de sécurité

Considérons un système de chiffrement par bloc $\mathcal{E}$ qui chiffre des blocs de n bits (*i.e.* $\mathcal{M} = \{0,1\}^n$).

1.a] Montrer que le mode opératoire ECB n'assure pas la sécurité sémantique.

1.b] Supposons que $\mathcal{E}$ est utilisé en mode compteur CTR. Montrer que si le nombre de blocs de suite chiffrante est suffisamment grand, alors il est facile de distinguer la suite chiffrante d'une suite aléatoire. Donner la longueur de la suite chiffrante pour que le distingueur ait un avantage supérieur à $1/2$ si le chiffrement par bloc $\mathcal{E}$ opère sur des blocs de 64 bits (comme le DES).

1.c] Montrer que le mode opératoire CBC n'assure pas la sécurité sémantique pour des messages suffisamment longs.

Indication : On pourra étudier le cas où deux blocs du chiffré sont égaux.

Exercice 2 : Mode opératoire CBC*

Un inconvénient majeur du mode opératoire CBC est qu'il est intrinsèquement séquentiel et ne permet pas de paralléliser les opérations de chiffrement. Considérons le mode de chiffrement modifié CBC* décrit dans la figure (1). Il permet d'effectuer plusieurs opérations de chiffrement ou de déchiffrement en parallèle.

2.a] Décrire comment le déchiffrement est effectué pour le mode opératoire CBC*.

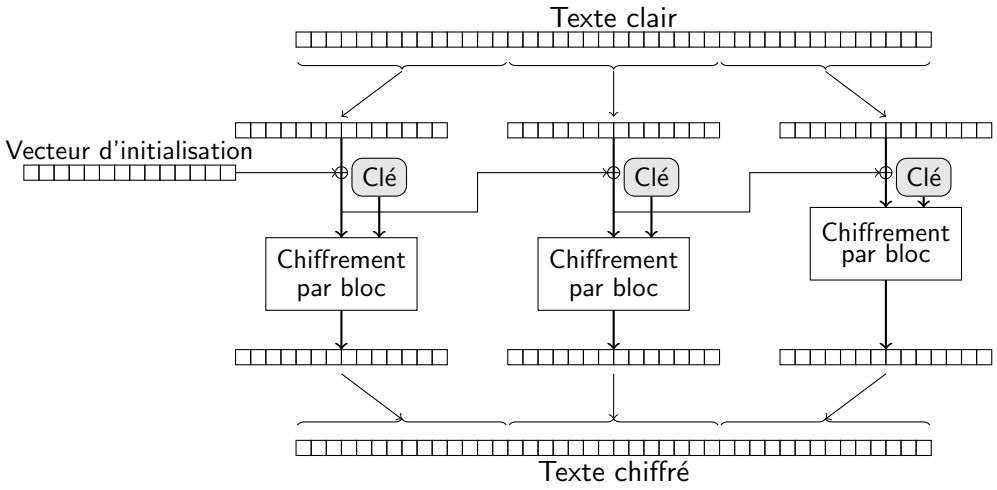

FIGURE 1 – Mode opératoire CBC*

2.b] Montrer que ce mode opératoire n'assure pas la sécurité sémantique.

Même en utilisant un mode opératoire pour étendre l'espace des textes clairs, un système de chiffrement par bloc qui opère sur des blocs de n bits ne peut chiffrer que des messages dont la longueur est un multiple de n. Il est donc nécessaire de définir un processus de *bourrage* ou *remplissage* (*padding*, en anglais) qui transforme un message de longueur arbitraire en un message dont la longueur est un multiple de n.

Dans la norme RFC2040, le processus de bourrage suivant a été proposé lorsque les messages à chiffrer sont constitués d'octets et que n est un multiple de 8. Étant donné un message M de longueur 8ℓ, nous effectuons la division euclidienne de 8ℓ par n et obtenons $8\ell = \alpha n + 8i$ de sorte qu'il manque i octets au dernier bloc de message. En notant $m_1 m_2 \dots m_{b-i}$ les octets du dernier bloc de message (avec $b = n/8$ la taille d'un bloc en octets), le processus de bourrage consiste à concaténer i fois l'octet [1] `0i` à la fin du dernier bloc. Ainsi pour un algorithme de chiffrement qui opère sur des blocs de 128 bits, soit 16 octets, le bloc de clair $m_1 \dots m_{12}$ sera transformé en $m_1 \dots m_{12} \| \texttt{04040404}$.

S. VAUDENAY a proposé une attaque contre ce processus de bourrage si le mode opératoire utilisé est le mode CBC. L'attaquant a accès à un oracle qui détermine si le message clair associé à un chiffré donné est bien formé pour cet encodage. Un tel oracle est plus faible qu'un oracle de déchiffrement (il ne retourne qu'un bit d'information) et en pratique si une erreur dans le calcul de l'encodage existe, certains protocoles de communication émettent un message d'erreur et peuvent donc jouer le rôle de cet oracle.

Exercice 3 : Attaque sur le mode CBC avec le processus de bourrage RFC2040

Considérons un attaquant qui a intercepté un chiffré $C = (C_1, C_2, \dots, C_n)$ produit par un système de chiffrement à blocs en mode CBC avec le processus de bourrage RFC2040. Nous supposons que l'attaquant connaît également le vecteur d'initialisation v correspondant.

1. Nous utiliserons cette fonte de type « machine à écrire » pour représenter la valeur d'un octet avec deux chiffres hexadécimaux : `00` = 0, `01` = 1, …, `0A` = 10, …, `10` = 15, …, `FF` = 255.

3.a] Montrer que si l'attaquant dispose d'un oracle qui détermine si le message clair associé à un chiffré arbitraire est bien formé pour l'encodage RFC2040, alors il peut déterminer l'encodage effectivement utilisé pour le chiffré C.

3.b] Modifier l'attaque pour qu'il détermine le dernier octet du dernier bloc de clair.

3.c] En itérant le processus, montrer que l'attaquant peut obtenir ainsi le message clair $M_1, M_2, \ldots, M_n$ en intégralité.

Exercice 4 : Schéma de FEISTEL à un, deux ou trois tours

4.a] Décrire un moyen pour distinguer un schéma de Feistel à un tour d'une permutation aléatoire (par une attaque à clairs connus).

4.b] Décrire un moyen pour distinguer un schéma de Feistel à deux tours d'une permutation aléatoire (par une attaque à clairs choisis).

4.c] ($\star$) Décrire un moyen pour distinguer un schéma de Feistel à trois tours d'une permutation aléatoire (par une attaque à chiffrés choisis).

Indication : On pourra demander à l'oracle de chiffrement le chiffré de deux messages X_0 et Y_0 avec $Y_0^L = X_0^L \oplus \delta$ et $Y_0^R = X_0^R$ (avec $\delta \neq 0$) puis à l'oracle de déchiffrement le clair associé au chiffré Z_3 avec $Z_3^L = Y_3^L \oplus \delta$ et $Z_3^R = Y_3^R$.