Université Bordeaux 1 — **MHT633 – Arithmétique et Cryptologie**
Mathématiques — Année 2008–2009

FEUILLE D'EXERCICES n° 4
Chiffrements symétriques

Exercice 1 – [Modes de chiffrement]
Soit $e_K : \mathbb{F}_2^n \to \mathbb{F}_2^n$ une fonction de chiffrement symétrique, et d_K sa fonction de déchiffrement. On découpe le message clair en k blocs de taille n : $P = P_1P_2 \dots P_k$.

1) Mode ECB (*electronic code book*) : on chiffre $C_i = e_K(P_i)$ et le chiffré de P est $C = C_1C_2 \dots C_k$. Dans ce mode de chiffrement, un même bloc est toujours chiffré de la même façon. Pourquoi est-ce un problème ? Explicitez le déchiffrement.

2) Mode CBC (*cipher block chaining*) : on choisit un vecteur d'initialisation $C_0 = v \in \mathbb{F}_2^n$; puis on chiffre $C_i = e_K(C_{i-1} + P_i)$, pour $i = 1, \dots, k$. Le chiffré de P est $C = C_0C_1C_2 \dots C_k$.

En quoi ce mode de chiffrement corrige-t'il le défaut du précédent ? Explicitez le déchiffrement. Montrez qu'il a l'inconvénient de propager une erreur éventuelle apparaissant dans un bloc de clair (effet d'avalanche).

3) Modes CFB (*cipher feedback*) et OFB (*output feedback*) : ils ont en commun de construire un masque $Z_0Z_1Z_2 \dots Z_k$ et de chiffrer par $C_i = P_i + Z_i$, le chiffré de P étant $C = C_0C_1C_2 \dots C_k$. Le déchiffrement est donc réalisé par la même procédure que le chiffrement.

Dans cette question, les blocs P_i sont maintenant de taille s, où $1 \leqslant s \leqslant n$. On choisit un vecteur d'initialisation $v \in \mathbb{F}_2^n$.

a) CFB avec $s = n$: $C_0 = v$, $Z_0 = 0$, $Z_i = e_K(C_{i-1})$ pour $i \geqslant 1$.

b) OFB avec $s = n$: $C_0 = 0$, $Z_0 = v$, $Z_i = e_K(Z_{i-1})$ pour $i \geqslant 1$.

c) CFB avec $s < n$: $C_0 = v$, P est découpé en blocs P_i de taille s, $Z_0 = 0$, $Z_1 = \mathrm{MSB}_s(e_K(C_0))$, et pour $i \geqslant 2$,

$$Z_i = \mathrm{MSB}_s(e_K(\mathrm{LSB}_{n-s}(C_{i-2}) \,||\, C_{i-1})).$$

d) OFB avec $s < n$: $C_0 = 0$, $Y_0 = v$, $Y_1 = e_K(Y_0))$, et pour $i \geqslant 2$,

$$Y_i = e_K(\mathrm{LSB}_{n-s}(Y_{i-2}) \,||\, \mathrm{MSB}_s(Y_{i-1})),$$

$$Z_i = \mathrm{MSB}_s(Y_i)$$

4) Mode CTR (*counter*) : encore un masque. On initialise un compteur c puis on calcule $Z_i = e_K(c)$ et on incrémente $c \leftarrow c + 1 \mod 2^n$. Le chiffré est $C_0 = c$, $C_i = P_i + Z_i$ et toujours $C = C_0C_1C_2 \dots C_k$.

Exercice 2 – [AES – Chiffrement]
Le principe d'AES a été vu en cours et une feuille complémentaire a été donnée qui résume les différentes étapes du chiffrement.

1) Appliquer `SubBytes` à l'octet (01010011).

2) Appliquer `MixColumns` au tableau

$$\begin{pmatrix} (11000001) & (00000111) & (00000000) & (11111111) \\ (11000000) & (00001000) & (00011110) & (11111100) \\ (11000011) & (00000100) & (00000001) & (11100000) \\ (10001001) & (00000110) & (11000000) & (00010111) \end{pmatrix}.$$

On se contentera de calculer quelques nouveaux octets.

Exercice 3 – [AES – Déchiffrement]

1) Définir la procédure inverse de chacune des procédures `SubBytes`, `ShiftRows`, `MixColumns` et `AddRoundKey`.

2) Montrer que l'on peut permuter `InvShiftRows` et `InvSubBytes`.

3) Indiquer comment modifier la clé de tour correspondante pour pouvoir permuter les procédures `AddRoundKey` et `InvMixColumns`.

4) Donner un découpage en tours pour le déchiffrement qui applique à chaque tour les procédures inverses des procédures de chiffrement *dans le même ordre que les procédures initiales*, en indiquant comment modifier l'expansion de clé (concaténation des clés de tour).

Exercice 4 – [Chiffrement de Pohlig–Hellman, Log discret]
Soit $\mathcal{P}$ l'ensemble des symboles (lettres, blocs) à chiffrer que l'on a par commodité identifiés à des entiers inférieurs ou égaux à un entier donné n. On identifie donc $\mathcal{P}$ avec un sous-ensemble de $\{0, 1, \ldots, n\}$. Par exemple si les symboles sont les blocs de deux lettres AB, ZC, ... codés 0001, 2502, ... on peut prendre $n = 2525$.

Soit $p > n$ un premier impair et soit e une clé secrète où e est un entier vérifiant

$$2 \leqslant e \leqslant p-1 \quad \text{et} \quad \operatorname{pgcd}(p-1, e) = 1.$$

Soit $m \in \mathcal{P}$. Le chiffrement est défini par $m \mapsto m^e \bmod p \in \mathbb{Z}/p\mathbb{Z}$.

1) Rappeler pourquoi e est inversible modulo $p-1$. On note d un entier qui représente e^{-1} modulo $p-1$. Monter que le déchiffrement est donné par $c \mapsto c^d \bmod p$.

2) Soit $p = 11$. On a chiffré 2 en 7. Quelles sont les clés de chiffrement et de déchiffrement ?

3) Soit $p = 31$. On sait que l'on a chiffré 2 en 4. Peut-on trouver la clé de chiffrement ?

4) Soit p un nombre premier, soit α un générateur de $(\mathbb{Z}/p\mathbb{Z})^*$ et soit $\beta \in (\mathbb{Z}/p\mathbb{Z})^*$. Le problème qui consiste à résoudre l'équation

$$\alpha^a = \beta \bmod p$$

d'inconnue a est appelé *problème du logarithme discret*. Montrer que ce problème a une solution définie de façon unique modulo $p-1$. On note alors

$$\log_\alpha \beta := a \in \mathbb{Z}/(p-1)\mathbb{Z}.$$

En général, le problème du logarithme discret est difficile à résoudre, mais il existe des algorithmes performants quand $p-1$ est friable, en particulier l'algorithme de Pohlig-Hellman qui fait l'objet de la question suivante.

5) On suppose que l'on connaît la décomposition de $p-1$ en produit de facteurs premiers :

$$p - 1 = \prod_{i=1}^{r} q_i^{c_i}.$$

La première idée consiste à calculer $a \bmod q_i^{c_i}$ pour chaque i puis à se servir du théorème chinois pour déterminer a.

a) Pour simplifier on pose $q_i = q$, $c_i = c$ et l'on cherche donc $x = a \bmod q^c$, avec $0 \leqslant x \leqslant q^c - 1$. On décompose x sous la forme

$$x = \sum_{i=0}^{c-1} a_i q^i \quad (0 \leqslant a_i \leqslant q-1)$$

et on se propose de calculer les a_i. Montrer qu'il existe un entier s tel que

$$a = \sum_{i=0}^{c-1} a_i q^i + s q^c.$$

b) Montrer que $\beta^{(p-1)/q} = \alpha^{a_0(p-1)/q}$.

c) En déduire une procédure de calcul de a_0.

d) Si $c = 1$ c'est terminé. On suppose donc maintenant que $c > 1$ et que l'on a calculé $a_0, a_1, \ldots, a_{j-1}$ où $1 \leqslant j \leqslant c-1$. On désire calculer a_j. On pose

$$\beta_j = \beta\alpha^{-(a_0+a_1q+\cdots+a_{j-1}q^{j-1})} \pmod{p}.$$

Montrer que $\beta_j^{(p-1)/q^{j+1}} = \alpha^{a_j(p-1)/q} \pmod{p}$ et en déduire une procédure pour calculer a_j.

e) En déduire un algorithme de calcul de x qui s'appuie sur les calculs successifs de $a_0, \beta_1, a_1, \ldots, \beta_{c-1}, a_{c-1}$.

6) Se servir de cet algorithme pour trouver les clés de chiffrement et de déchiffrement sachant que $p = 37$ et que 2 se code 19.

7) Même question avec $p = 3001$, sachant que 14 se code 1873.