Partiel ISEC 2012-2013

Version du 26 février 2013

Calculatrice et documents **interdits**. Le barème est donné à titre indicatif.

Exercice 1 – Cours (4 points)

1. Donner la définition d'une fonction à sens unique avec trappe.
2. Expliquer comment construire une algorithme de signature avec une fonction à sens unique avec trappe.
3. Calculer pgcd(434,128) et une relation de Bézout entre 434 et 128.
4. Soient $p = 251$ et $g = 11$ un générateur de $\mathbb{Z}_p^*$. On note $n_A = 3$ le secret d'Alice et $n_B = 4$ le secret de Bob. Déterminer la clef commune d'Alice et Bob construite avec le protocole de Diffie-Hellman (on donne $11^2 \equiv 121 \bmod 251$, et $11^6 \equiv 3 \bmod 251$).

Exercice 2 – Maple (4 points)

On se donne un algorithme de chiffrement par blocs :

$$\begin{array}{ccccccc} E & : & \{0,1\}^{128} & \times & \{0,1\}^{128} & \to & \{0,1\}^{128} \\ & & k & , & x & \mapsto & E_k(x) \end{array}$$

Ainsi, E_k désigne le chiffrement. On note D_k la fonction de déchiffrement.

On suppose que vous avez une procédure `Chiffrement:=proc(M, k)` qui prend comme paramètres un message `M` (sous forme d'une liste de 128 bits), une clef `k` (sous forme d'une liste de 128 bits) et qui retourne $E_k(M)$ (sous forme aussi d'une liste de 128 bits). La procédure `Dechiffrement:=proc(C, k)` retourne $D_k(C)$ (sous forme aussi d'une liste de 128 bits).

- Écrire une fonction Maple `Cut:=proc(LongM, k)` qui prend comme paramètres un message `LongM` sous forme d'une liste (dont la taille est un mutliple de 128) et découpe `LongM` en des blocs de taille 128. La fonction retourne une liste de listes (la liste des blocs qui composent `LongM`).
- Écrire une fonction Maple `EncryptECB:=proc(LongM,k)` qui chiffre `LongM` en mode ECB avec le clef `k` (vous utiliserez `Chiffrement` et `Cut`). La fonction retourne le chiffrement de `LongM` comme une liste de listes.
- Écrire une fonction Maple `DecryptECB:=proc(C,k)` qui déchiffre – en mode ECB – un chiffré `C` avec la clef `k` (vous utiliserez `Dechiffrement` et `Cut`). La fonction retourne une liste de listes.

Exercice 3 – Schéma de Feistel sur 3 tours (5 points)

Considérons un schéma de Feistel sur 3 tours défini par les fonctions de tour F_1, F_2 et F_3 avec $F_i : \mathbb{F}_2^n \to \mathbb{F}_2^n$ pour chaque $i, 1 \leq i \leq 3$. Soit $X_0 = (X_0^L, X_0^R)$ une entrée du schéma. Pour tout $i, 1 \leq i \leq 3$, nous avons :

$$\left\{ \begin{array}{rcl} X_i^L & = & X_{i-1}^R, \\ X_i^R & = & X_{i-1}^L \oplus F_i(X_{i-1}^R). \end{array} \right.$$

- Exprimer (X_3^L, X_3^R) (la sortie) en fonction de (X_0^L, X_0^R) entrée.
- Inversement, exprimer (X_0^L, X_0^R) en fonction de (X_3^L, X_3^R).

On considére deux messages $X_0 = (X_0^L, X_0^R)$ et $Y_0 = (X_0^L \oplus \delta, X_0^R)$, avec $\delta \neq 0 \in \mathbb{F}_2^n$. On note par $Y_3 = (Y_3^L, Y_3^R)$ le chiffrement de Y_0 après trois tours de Feistel.

- Exprimer (Y_3^L, Y_3^R) en fonction de (X_0^L, X_0^R).
- En déduire une relation entre $Y_3^L \oplus X_3^L$ et (X_0^L, X_0^R).

Exercice 4 – Mode CBC dégradé (7 points)
On se donne un algorithme de chiffrement par blocs :

$$\begin{array}{ccccccc} E & : & \{0,1\}^s & \times & \{0,1\}^n & \rightarrow & \{0,1\}^n \\ & & k & , & x & \mapsto & E_k(x) \end{array}$$

En mode CBC, on chiffre un message $m = m_1, \ldots, m_t$:

$$c_0 = \mathrm{IV},\ c_i = E_k(m_i \oplus c_{i-1}),\ 1 \leq i \leq t,$$

avec IV un vecteur d'initialisation de n bits.

– Comment se propage une erreur (lors de la transmission d'un bloc) sur le déchiffrement en mode CBC ?

On considère une version **dégradé** du mode CBC dans lequel le premier message est chiffré avec un IV aléatoire. Le dernier bloc du chiffré d'un message est utilisé comme vecteur initialisation pour le chiffrement du message suivant.

Soient $c_1, \ldots, c_t$ le chiffrement en mode CBC dégradé de $m = m_1, \ldots, m_t$ (avec IV comme vecteur d'initialisation), i.e.

$$c_1, \ldots, c_t \leftarrow \mathrm{CBC\_}E_k(m_1, \ldots m_t).$$

On pose $m' = m^* \oplus c_0 \oplus c_t, m'_2, \ldots, m'_t$, avec $m^* \in \{0,1\}^n$. On note :

$$c'_1, \ldots, c'_t \leftarrow \mathrm{CBC\_}E_k(m'_1, \ldots, m'_t).$$

– Si m' est le message chiffré juste après le premier message m, comment est chiffré m' en mode CBC dégradé ?
– Donner le chiffrement c'_1 qui correspond au chiffrement du premier bloc $m'_1 = m^* \oplus c_0 \oplus c_t$ de m'.
– Montrer que :

$$c'_1 = c_1 \iff m^* = m_1.$$

– Comment généraliser cette propriété pour un bloc m_j en position quelconque, c'est à dire donner un message m' tel que :

$$c'_1 = c_j \iff m^* = m_j.$$