Exercices
Semaine 8

Cours Turing

1 Représentation binaire

a) Ecrire un programme qui transforme un nombre entier en une chaîne de caractères qui soit la représentation binaire de celui-ci.

Par exemple, si l'entrée est 1984, alors la sortie doit être "11111000000".

Indication : Il y a deux façons de faire :

- soit "partir depuis la droite", en cherchant d'abord le bit de poids le plus faible. La première question à se poser est alors : "Pour un nombre donné N, quand-est-ce que ce bit vaut 1 ou 0 ?"

- soit "partir depuis la gauche", c'est-à-dire du bit de poids le plus fort (qui par définition vaut toujours 1). La première question à se poser est alors : "Pour un nombre donné N, quand est-ce que le *deuxième* bit (depuis la gauche) vaut 1 ou 0 ?"

b) Ecrire un programme qui transforme une chaîne de caractères en une autre chaîne de caractères qui soit la représentation binaire de celle-ci.

Ici, chaque caractère de la chaîne d'origine est encodé par un nombre entier compris entre 0 et 255 (code ASCII étendu) : il faut donc utiliser la partie a) de l'exercice, mais modifier légèrement celle-ci pour étendre la représentation binaire de chaque caractère à 8 bits exactement.

2 Clé à usage unique

a) Comme nous l'avons vu dans le cours (et encore vérifié ci-dessus...), la clé à usage unique est un système à 100% sûr pour chiffrer un message. Ci-dessous, nous vous proposons plusieurs versions légèrement modifiées de ce système. A vous d'identifier le(s)quel(s) restent 100% sûrs ! (pour rappel, le message d'origine M et le message chiffré C sont des séquences de n bits)

i) soit K un bit tiré au hasard ; pour chaque valeur de i entre 0 et $n-1$, $C_i = M_i \oplus K$.

ii) soit K un clé de n bits tirée au hasard : pour chaque valeur de i entre 0 et $n-1$,

- $C_i = M_i$ avec probabilité $1/2$
- $C_i = M_i \oplus K_i$ avec probabilité $1/2$

iii) pour chaque valeur de i entre 0 et $n-1$,

- $C_i = M_i$ avec une probabilité 1/2
- $C_i = M_i \oplus 1$ avec probabilité 1/2

b) Transposons maintenant le système de clé à usage unique sur une séquence de lettres :

Chaque lettre M_i du message d'origine est chiffrée comme suit : $C_i = M_i + K_i \,(\text{mod } 26)$; le XOR a été remplacé ici par l'addition modulo 26 vue la dernière fois, et la clé K est maintenant une séquence de n lettres. A nouveau, parmi les systèmes suivants, lequel est 100% sûr ?

i) Chaque lettre K_i est tirée uniformément au hasard parmi les 26 lettres de l'alphabet.

ii) Même clé K, mais cette fois :

- $C_i = M_i$ avec probabilité 1/2
- $C_i = M_i + K_i \,(\text{mod } 26)$ avec probabilité 1/2.

iii)

- $C_i = M_i$ avec probabilité 1/2
- $C_i = M_i + 13 \,(\text{mod } 26)$ avec probabilité 1/2.

Indication : Pour répondre à cette denière question, vous pouvez (vous devez, même :-) programmer ce système de chiffrement et essayer de voir s'il est possible de le décrypter sans connaître la clé K.

3 Système DES

Supposons qu'Alice et Bob, qui partagent une clé secrète $K = (K_a, K_b)$ d'une longueur de $2n$ bits, utilisent le "premier round" du système DES suivant pour communiquer :

1. Pour envoyer le message $M = (M_a, M_b)$, Alice effectue successivement les deux opérations suivantes :

$$C_a = M_a \oplus f(K_a, M_b) \quad \text{et} \quad C_b = M_b \oplus f(K_b, C_a)$$

où $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ est une fonction non-linéaire donnée (connue de tout le monde). Puis Alice envoie le message chiffré $C = (C_a, C_b)$ à Bob.

2. Pour déchiffrer le message provenant d'Alice, Bob effectue successivement les deux opérations suivantes :

$$D_b = C_b \oplus f(K_b, C_a) \quad \text{et} \quad D_a = C_a \oplus f(K_a, D_b)$$

Supposons maintenant qu'Eve intercepte le message chiffré $C = (C_a, C_b)$, et qu'elle ait également eu accès par un moyen détourné à la première partie de la clé secrète K_a. Laquelle des affirmations suivantes est vraie ?

(A) Eve peut décrypter sans problème le message entier M.

(B) Eve peut décrypter sans problème la première partie M_a, mais pas la seconde partie M_b.

(C) Eve peut décrypter sans problème la seconde partie M_b, mais pas la première partie M_a.

(D) Eve ne peut décrypter ni la première partie M_a, ni la seconde partie M_b.