Exercices
Semaine 8

Cours Turing

1 Système DES simplifié

Dans cette exercice, nous vous proposons d'implémenter une version simplifée du système DES. En entrée, votre programme demandera une clé K (au format str, de longueur paire), ainsi qu'un texte T (au format str), dont la longueur soit un multiple de la longueur de la clé K.

a) Chiffrement : La première étape consiste à couper le texte T en une suite de messages M de même longueur que la clé K, puis pour chaque message M :

- convertir $M = (M_a, M_b)$ et $K = (K_a, K_b)$ en deux suites de nombres compris entre 0 et 25 ;

- calculer le message chiffré $C = (C_a, C_b)$ selon le schéma suivant :

pour la première moitié : $C_a[i] = (M_a[i] + f(K_a[i], M_b[i]))$ (mod 26)

pour la seconde moitié : $C_b[i] = (M_b[i] + f(K_b[i], C_a[i]))$ (mod 26)

- finalement, convertir le message $C = (C_a, C_b)$ en string, et ajouter ce message chiffré au texte chiffré final.

- et répéter la même suite d'opérations pour toutes les parties du texte.

Plusieurs **remarques** s'imposent :

- pour simplifier, nous travaillons ici avec des nombres entre 0 et 25 plutôt que des bits.

- vous avez le choix de la fonction f ! Mais à nouveau, pour simplifier, nous vous proposons ici de choisir une fonction f non-linéaire qui agit sur deux nombres seulement, et pas sur les deux suites de nombres.

b) Déchiffrement :

Refaire la même chose, mais en utilisant successivement :

pour la seconde moitié : $D_b[i] = (C_b[i] - f(K_b[i], C_a[i]))$ (mod 26)

pour la première moitié : $D_a[i] = (C_a[i] - f(K_a[i], D_b[i]))$ (mod 26)

(comme nous utilisons des nombres entre 0 et 25 plutôt que des bits, il faut faire des soustractions ici).

2 Clé à usage unique

a) Comme nous l'avons vu dans le cours, la clé à usage unique est un système sûr à 100% 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 : $C_i = M_i + K_i \pmod{26}$; 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
- $C_i = M_i + K_i \pmod{26}$ avec probabilité 1/2.

iii)

- $C_i = M_i$ avec probabilité 1/2
- $C_i = M_i + 13 \pmod{26}$ 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.