info505 : mathématiques pour l'informatique
TD 2 : cryptographie I

Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 22, poste : 94 22
email : Pierre.Hyvernat@univ-savoie.fr
www : http://www.lama.univ-savoie.fr/~hyvernat/
wiki : http://www.lama.univ-savoie.fr/wiki/

Exercice 1 : Une cryptanalyse "facile"

Le système de César ($n \mapsto n+3 \pmod{26}$) ainsi que ces variantes (décalage de lettres) est assez facile à analyser : une fois qu'on a trouvé une seule lettre, les autres suivent directement.

Le texte suivant a été codé en utilisant une permutation des lettres de l'alphabet :

```
qb jno aboa bai qx zunab yh pnoyb qx ptbhe mxkixvbb; zxk zuxzho mboab bo
bikb at jtbo mnhkgh, lhb zbhe pbpb lht anoi qba mqha ytwwtztqba x
znoiboibk bo inhib xhikb zunab o'noi mntoi znhihpb y'bo ybatkbk mqha
lh'tqa bo noi. bo lhnt tq o'bai mxa gkxtabpjqxjqb lhb inha ab iknpmboi:
pxta mqhini zbqx ibpntvob lhb qx mhtaaxozb yb jtbo shvbk bi ytaitovhbk
qb gkxt y'xgbz qb wxhe, lht bai mknmkbpboi zb lh'no onppb qb jno aboa nh
qx kxtano, bai oxihkbqqbpboi bvxqb bo inha qba unppba; bi xtoat lhb qx
ytgbkatib yb ona nmtotnoa ob gtboi mxa yb zb lhb qba hoa anoi mqha
kxtanooxjqba lhb qba xhikba, pxta abhqbpboi yb zb lhb onha znoyhtanoa
ona mboabba mxk ytgbkaba gntba, bi ob znoatybknoa mxa qba pbpba zunaba.
zxk zb o'bai mxa xaabc y'xgntk q'bamkti jno, pxta qb mktoztmxq bai yb
q'xmmqtlhbk jtbo. qba mqha vkxoyba xpba anoi zxmxjqba yba mqha vkxoya
gtzba xhaat jtbo lhb yba mqha vkxoyba gbkiha; bi zbhe lht ob pxkzuboi
lhb wnki qboibpboi mbhgboi xgxozbk jbxhznhm yxgxoixvb, a'tqa ahtgboi
inhsnhka qb yknti zubpto, lhb ob wnoi zbhe lht znhkboi bi lht a'bo
bqntvoboi.
```

Question 1. Essayez de faire une cryptanalyse de ce code, et envoyez moi le résultat par email. (Et une sucette pour les premiers qui me donnent une référence du texte...)

- un fichier texte est disponible sur ma page web
- la fréquence des lettres en français est disponible sur wikipedia
- pour vous faciliter la vie :
  `http ://cryptoclub.math.uic.edu/substitutioncipher/frequency_txt.htm`

Exercice 2 : Un DES "simplifié"

Nous allons utiliser une version simplifiée de DES : "simplified DES". Une description succincte suit, et les détails seront donnés à l'oral.

Une clé secrète partagée de 10 bits permet de générer deux sous clés de 8 bits :

- on applique la permutation $(3,5,2,7,4,10,1,9,8,6)$ sur la clé
- le résultat est divisé en deux parties de 5 bits, et chaque partie est "décalée" (permutation circulaire) vers la gauche. On obtient (A_1, A_2)
- on applique la fonction $P = (6,3,7,4,8,5,10,9)$ sur (A_1, A_2) pour obtenir la première sous-clé : K_1
- on décale A_1 et A_2 de deux bits à gauches pour obtenir (B_1, B_2) et on applique la fonction P pour obtenir la deuxième sous-clé K_2

Pour l'encodage proprement dit, on utilise des blocs de 8 bits. On procède comme suit :
- on commence par permuter le bloc avec $(2, 6, 3, 1, 4, 8, 5, 7)$
- le bloc est divisé en deux : (L_0, R_0) et on calcule $\big(L_1 = L_0 \oplus F(R_0, K_1), R_1 = R_0\big)$
- on inverse les blocs $(L'_1 = R_1, R'_1 = L_1)$
- on calcule $\big(L_2 = L'_1 \oplus F(R'_1, K_2), R_2 = R'_1\big)$
- on termine en utilisant la permutation inverse de la première étape : $(4, 1, 3, 5, 7, 2, 8, 6)$

Il reste maintenant à définir la fonction F : elle prend en entrée un bloc de 4 bits et une sous clé K_i. Ensuite,
- elle l'étend en un bloc de 8 bits avec la fonction $(4, 1, 2, 3, 2, 3, 4, 1)$
- elle ajoute $(\oplus)$ la sous-clé K_i
- les 4 premiers bits sont passés dans S_1 et les 4 derniers dans S_2 (voir plus bas)
- le résultat (4 bits) est finalement permuté avec $(2, 4, 3, 1)$; c'est le résultat.

Les matrices S_1 et S_2 sont définies par

$$S_1 = \begin{pmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{pmatrix} \qquad \text{et} \qquad S_2 = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{pmatrix}$$

et sont utilisées comme suit : étant donnés 4 bits (b_1, b_2, b_3, b_4)
- on utilise les bits (b_1, b_4) pour former un entier en base 2, qui nous donne la ligne
- les bits (b_2, b_3) forment un entier en base 2 qui donne la colonne
- on renvoie les deux bits constituant l'entrée correspondante dans la matrice.

Le véritable DES est très similaire, mais utilise
- des clés de 58 bits
- des blocs de 64 bits
- 16 itérations au lieu de 2.

Question 1. On veut encoder, avec la clé (0110011100), le texte "`OK`" qui correspond, en ASCII, à la chaîne de bits (1001111 1001011). Comment l'encodez-vous ?

Question 2. On veut décrypter, avec la clé (1101111001) la chaîne : (00010101 00000110). Quel est le texte clair ?

Question 3. Programmez (chez vous) le DES simplifié dans le langage de votre choix.

Question 4. Le DES double n'est pas considéré comme sûr. La raison est que le temps nécessaire pour casser DES double n'est pas très différent du le temps pour casser DES. Le seul problème est qu'il faut disposer d'une quantité de mémoire énorme...

Supposons qu'Alice envoie un message à Bob DES double : $c = \mathrm{DES}\big(\mathrm{DES}(m, k_1), k_2\big)$. Si un observateur malveillant (Eve) peut essayer de casser le code par une attaque "meet in the middle". Pour faire ça, Eve doit disposer d'au moins un message clair m et de son code c ; Eve peut ensuite calculer $\mathrm{DES}(m, k)$ pour toutes les clés k possible, et décoder c avec $\mathrm{DES}^{-1}(c, k)$ pour toutes les clés possibles k.

Comment peut-on se servir de ces résultats pour casser l'encryption de DES double ? Est-ce que c'est facilement faisable en pratique ?