# INFO-F-405 : Sécurité Informatique
# Travaux Pratiques:
# Séance 1 - Chiffrement et cryptanalyse statistique

Assistant: Qachri Naïm

Année académique 2009-2010

## 1 Introduction

Le but de ce TP est de vous familiariser avec le chiffrement, le déchiffrement et les premiers outils dans l'histoire de la cryptographie et de la cryptanalyse. Nous partirons d'un cas d'école : *le chiffrement par décalage*. Nous avancerons dans le temps vers l'une ou l'autre forme de chiffrement alphabétique un peu plus évolué. Nous montrerons pourquoi, dans le monde dans lequel nous vivons, ces méthodes de chiffrement sont plus que dépassés.

## 2 Chiffrement monoalphabétique

Le chiffrement *monoalphabétique* a été une des premières formes de cryptographie. La forme la plus connue est *le chiffrement par décalage*. À l'époque de César, ce chiffrement était utilisé pour la valeur particulière de clé $k = 3$. Ce chiffrement était bien suffisant, puisque la majorité des gens ne savaient pas lire. Toutefois avec le temps, cette méthode de chiffrement devint très vite obsolète par sa faiblesse aux attaques d'analyse de fréquences.

### 2.1 Le chiffrement et déchiffrement

#### 2.1.1 Le chiffrement

Le chiffrement est assez simple à représenter. Si nous représentons chaque lettre par sa valeur numérique comprise en 0 et 25. C'est à dire que nous projetons chaque lettre avec la correspondance suivante :

$$A \rightarrow 0\ ,\ B \rightarrow 1,\ ..., Z \rightarrow 25$$

Nous pouvons donc ainsi exprimer la fonction $e_k(x)$ du *chiffrement par décalage* dans $\mathbb{Z}_{26}$ avec la clé « secrète » de $k$, pour $0 \leq k \leq 25$, qui représente l'amplitude du décalage. Voici la définition de la fonction de décalage :

$$e_k(x) = (x + k) \bmod 26$$

### 2.1.2 Le déchiffrement

La fonction de déchiffrement est la fonction inverse $d_k(x)$, qui s'interprète de façon inverse de la fonction de chiffrement :

$$d_k(y) = (y - k) \bmod 26$$

**Exercice :** Écrivez le code qui déchiffre le texte suivant de longueur 88 avec k = 'i' : "wp! dwcavmaiczqmhniqzmicbzmumvb,lqbtmkpib :qkq,bwcbtmuwvlmmabnwc.rmacqanwc. dwcambmanwttm."

## 2.2 La cryptanalyse

### 2.2.1 Force Brute

Il n'est pas difficile de se dire qu'une attaque par force brute est très efficace contre ce genre de chiffrement. Tester 26 possibilités est le genre de calcul qui ne prend aucun temps à effectuer.... ni à écrire en `C++`. Là encore, un regard humain ou une analyse automatisée du texte partiellement chiffré permet de retrouver des mots courts comme « de » ou « le », permettant ainsi de retrouver la lettre clé plus rapidement encore.

**Exercice :** Écrivez le code permettant de casser par force brute le message suivant :

```
"zogvhjmvgzyzxzxd,x'zno:njtzuxzlpzqjpnqjpymdzuvqjdmg'vdmy'zomz;jp,kjpmkvmgzmk
gpnndhkgzhzio:izqjpndhvbdizukvnzomzydaazmziozyzxzlp'dgzpokpnzhwgzmvvpompdlpz
qjpnapnndzujpzpnndzukpzomzzimznoviodyziodlpzvxzlpzqjpnapoznnvinevhvdnkvmvdom
zvpomzlpzqjpni'zodzuvqvioy'zomzyzqzipzxzlpzqjpnzozn."
```

### 2.2.2 Analyse

La cryptanalyse est assez simple. Nous allons utiliser ce qu'on appelle l'analyse de fréquences pour retrouver les lettres chiffrées par la substitution. Cette méthode appartient à la famille des attaques *statistiques*.

L'idée est de tester la fréquence d'apparition des lettres du texte chiffré et de comparer cette fréquence à un tableau de référence de la langue dans laquelle le message chiffré a été écrit. Il y a de fortes chances pour trouver des correspondances entre les lettres. Après plusieurs essais/erreurs, il n'est pas difficile de retrouver un mot (c'est ici qu'intervient l'humain), d'en étudier le décalage et

donc de retrouver la lettre de référence. Voici d'ailleurs un exemple de table pour la langue française :

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 9,42 | 1,02 | 2,64 | 3,39 | 15,87 | 0,95 | 1,04 | 0,77 | 8,41 | 0,89 | 0,00 | 5,34 | 3,24 |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 7,15 | 5,14 | 2,86 | 1,06 | 6,46 | 7,90 | 7,26 | 6,24 | 2,15 | 0,00 | 0,30 | 0,24 | 0,32 |

**Exercice :** Écrivez le code permettant sur base du tableau donné ci-dessus, de faire une analyse de fréquences du texte chiffré de l'exercice précédent. Une fois que vous êtes sûrs de la clé de décalage que vous avez trouvée par comparaison entre le tableau donné ci-dessus et celui calculé, vérifiez la valeur avec celle trouvée par force brute.

### 2.3 Autres chiffrement classiques

D'autres chiffres existent en monoalphabétique, comme *le chiffrement par substitution*, dont le chiffrement par décalage n'est qu'un cas particulier. Identiquement, il existe une autre méthode particulière de chiffrement par substitution appelée *chiffrement affine*. Je vous renvoie à [1] pour plus de détails.

## 3 Le chiffrement de Vigenère

Nous quittons ici le domaine du chiffrement *monoalphabétique*, qui consiste en une permutation de lettres, pour le chiffrement dit *polyalphabétique* avec le chiffrement de Vigenère. Le chiffrement de Vigenère est une réussite cryptographique importante dans l'histoire puisqu'elle a tenu en échec les cryptanalystes pendant une période entre 200 et 300 ans.

### 3.1 Le chiffrement et déchiffrement

#### 3.1.1 Chiffrement

Le chiffrement de Vigenère n'est pas très compliqué. Pour chiffrer, il faut utiliser le carré de Vigenère. Ce carré contient les 26 décalages possibles du chiffrement par décalage. Il suffit alors de choisir un mot clé, de prendre votre texte clair et d'y copier en-dessous le mot clé autant de fois qu'il le faudra pour avoir un texte composé des mots clés aussi grand que le texte clair. Ensuite, il suffit pour chaque caractère du texte clair de lui appliquer le chiffrement par décalage de la lettre du mot clé qui lui est associé. Voici le carré de Vigenère :

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
B C D E F G H I J K L M N O P Q R S T U V W X Y Z A
C D E F G H I J K L M N O P Q R S T U V W X Y Z A B
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

E F G H I J K L M N O P Q R S T U V W X Y Z A B C D
F G H I J K L M N O P Q R S T U V W X Y Z A B C D E
G H I J K L M N O P Q R S T U V W X Y Z A B C D E F
H I J K L M N O P Q R S T U V W X Y Z A B C D E F G
I J K L M N O P Q R S T U V W X Y Z A B C D E F G H
J K L M N O P Q R S T U V W X Y Z A B C D E F G H I
K L M N O P Q R S T U V W X Y Z A B C D E F G H I J
L M N O P Q R S T U V W X Y Z A B C D E F G H I J K
M N O P Q R S T U V W X Y Z A B C D E F G H I J K L
N O P Q R S T U V W X Y Z A B C D E F G H I J K L M
O P Q R S T U V W X Y Z A B C D E F G H I J K L M N
P Q R S T U V W X Y Z A B C D E F G H I J K L M N O
Q R S T U V W X Y Z A B C D E F G H I J K L M N O P
R S T U V W X Y Z A B C D E F G H I J K L M N O P Q
S T U V W X Y Z A B C D E F G H I J K L M N O P Q R
T U V W X Y Z A B C D E F G H I J K L M N O P Q R S
U V W X Y Z A B C D E F G H I J K L M N O P Q R S T
V W X Y Z A B C D E F G H I J K L M N O P Q R S T U
W X Y Z A B C D E F G H I J K L M N O P Q R S T U V
X Y Z A B C D E F G H I J K L M N O P Q R S T U V W
Y Z A B C D E F G H I J K L M N O P Q R S T U V W X
Z A B C D E F G H I J K L M N O P Q R S T U V W X Y

Montrons un exemple de chiffrement. Si la phrase à chiffrer est "Tout a une morale si l'on cherche bien." avec la clé "lewis", alors nous procédons comme suit :

| Texte clair | T | O | U | T | A | U | N | E | M | O | R | A | L | E | ..... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| clé | L | E | W | I | S | L | E | W | I | S | L | E | W | I | ..... |
| Text chiffré | E | S | Q | B | S | F | R | A | U | .... | | | | | |

### 3.1.2 Déchiffrement

Pour déchiffrer, ça revient tout simplement à ajouter, en-dessous du texte, la clé copiée autant de fois que le texte le demande, et d'effectuer les opérations de déchiffrement lettre par lettre par la même méthode que dans le chiffrement par décalage.

**Exercice :** Déchiffrez le texte suivant avec la clé "edgar" :
" iwiegiqjaexgotcijxaehogsfpxzifrqkskghxtrmqkmvrwvajexyszhllfzglrehyrtsvplsaxmqkrrmw
gugvhsivvfuughrkicghycrvditvvhycfqpkcyefangsxxrrmwreuiyonvvigczphseexiurdiqzuegkofw
vhiejxdjiiitaicwsxejiqzeexxtsvrvsazwggpiiviehyhtolwvgvfrvjebmgjjvrhjemelyprwoksltsusvvf
gprfojdvjdhrzuxkrlrhihrrwolcsqjetvbvtfkugpymhhivrdhskvxyeaimagveljoegwuukhdhoihtaet
ioaitmhzazxganvivvetivomgphzecghveehdtthydriexhrlzkhtcvkuusjmhxeuypgrzrlrdlxsgrrmw
xerfvullqhttzrvullfoksrrvratphl "

## 3.2 Test de coïncidence

Le test de coïncidence est un test statistique préliminaire qui permet de déterminer la langue utilisée pour le texte clair et ainsi utiliser les statistiques adaptées pour le reste de la cryptanalyse. Ce test se base sur l'idée que, peu importe le décalage, la fréquence probable d'apparition globale des lettres reste inchangée. Nous avons donc, pour $p_i$ la fréquence d'apparition de la lettre $i$ dans une langue précise, la quantité suivante :

$$I_c(x) = \sum_{i=0}^{25} {p_i}^2$$

**Exercice :** Sur base du tableau des fréquences des lettres de l'alphabet donné plus haut, calculez l'indice de coïncidence pour la langue française. En quoi ce test peut vous aider à accélérer la cryptanalyse de la méthode de chiffrement par décalage ?

## 3.3 Test de Kasiski

Le test de Kasiski n'est pas très compliqué et permet d'avoir une bonne approximation de la taille de la clé. En fait, il revient à chercher des *motifs* de 2, 3 ou 4 lettres qui reviennent régulièrement. Une fois trouvé, nous mesurons la distance entre ces *motifs*. Une fois fait, nous déterminons le *pgcd* de ces distances, et ainsi nous déterminons approximativement la taille de la clé. La section suivante vous donne un test pour confirmer la taille de cette dernière.

**Exercice :** Sur base de la méthode donnée, retrouvez des *motifs*, et déterminez la taille possible de la clé.

## 3.4 Le test automatisé sur l'indice de coïncidence

Le test automatisé sur l'indice de coïncidence est un test statistique qui a pour but de tester un texte chiffré et d'en déterminer, selon la langue, une statistique assez "fiable", sur base du test de Kasiski, pour trouver/confirmer la longueur du mot formant la clé de chiffrement.

Ce test commence par diviser le texte en $m$ sous-chaînes $z_1, z_2, ....., z_m$ de y, le texte chiffré, en l'écrivant colonne par colonne. C'est à dire que l'on va construire, dans un tableau $m \times (\frac{n}{m})$ (où $n$ est la longueur du texte chiffré), chaque ligne $k$ comme une des sous-chaînes. Chaque valeur représente en position $i$ de la ligne $k$, la valeur $y_{i+k}$ du texte chiffré. Les lignes de ce tableau, une fois construit, représentent les sous-chaînes $z_i$, $1 \leq i \leq m$. On obtient $z_i$, qui représente la ligne $i$ de la matrice, comme suit :

$$z_i = y_i \ y_{m+i} \ y_{2m+i} \ ....$$

Un chiffrement *monoalphabétique* n'altère en rien la valeur de l'*indice de coïncidence*. Dans Vigenère, chaque sous-chaîne $z_i$, si $m$ est la longueur de la clé, possède un indice de coïncidence proche de la langue du texte clair, qui pour le français est $I_c = 0,0778$. L'indice de coïncidence d'un contenu parfaitement uniformisé est $I_c = 0,038$, hors l'indice pour la langue française est assez éloigné pour que l'on puisse distinguer un texte d'une langue précise du bruit. C'est ce qui nous amène à l'attaque globale.

**Exercice :** À partir du programme précédent, ajoutez le test pour vérifier la longueur du mot déterminé par le test de Kasiski et pour confirmer la taille de la clé du chiffré donné ci-dessus. Le test revient à calculer pour chaque $m$,l'indice de coïncidence $z_1$. Si cet indice est en accord avec le test de Kasiski donné plus haut, alors l'indice de coïncidence pour $m =$*taille de la clé* sera proche de la valeur pour le français.

### 3.5 De la combinaison des attaques à la cryptanalyse de Vigenère

Nous avons vu comment déterminer la langue et comment déterminer la longueur du mot clé de manière efficace. Il nous reste plus qu'à déchiffrer le texte. Comme nous l'avons vu plus haut, chiffrer avec Vigenère revient à chiffrer la sous-chaîne $z_i$, comme dans *un chiffrement par décalage*.

La clé de la cryptanalyse est là. Il suffit de cryptanalyser par analyse de fréquences chaque sous-chaîne $z_i$ pour obtenir les décalages. Un test basé sur le test de coïncidence peut nous aider à accélérer le calcul. En utilisant le test suivant, nous pouvons déterminer un test qui va analyser le décalage de chaque sous chaînes, sur base de *l'indice de coïncidence*, en utilisant, pour $0 \leq g \leq 25$, la quantité suivante :

$$\sum_{i=0}^{25} p_i \cdot \frac{f_{i+g}}{n/m}$$

où $p_i$ représente la fréquence d'apparition de la lettre en $i$ème position dans la langue française, $f_{i+g}$ le nombre de fois que, dans la sous-chaîne $z_i$, apparaît la lettre en $(i+g)$ème position dans l'alphabet, et $n/m$ représente la longueur de la chaîne $z_i$.

Si $g$ correspond au décalage de la lettre clé $k_i$ pour la sous-chaîne $z_i$, alors la quantité ci-dessus vaudra $\approx 0,0778$.

**Exercice :** Ajoutez à votre code le test pour déterminer le décalage de chaque sous-chaîne et trouver ainsi la clé du texte chiffré précédent.

## Références

[1] Wikipedia - le portail de la cryptographie.

[2] Henry Beker et Fred Piper. *Cipher Systems - The protection of communications.* Northwood Publications, 1982.

[3] Simon Singh. *L'histoire des codes secrets.* Number 15097. Le livre de poche, 1999.

[4] Douglas Stinson. *Cryptographie - Théorie et pratique.* Vuibert, second edition, 2003.