Travaux pratiques

informatique

Cryptographie à clef secrète

Le principe de la cryptographie à clef secrète est probablement le principe le plus naturel auquel on pense lorsqu'on envisage de cacher une information : l'émetteur et le récepteur partagent un secret commun qui permet de chiffrer et de déchiffrer un texte. Les opérations pour le codage et pour le décodage sont alors essentiellement les mêmes, d'où le qualificatif « symétrique » pour de tels cryptosystèmes.

Nous allons nous intéresser à un procédé de chiffrement à clef secrète apparu au XVI^{e} siècle : le chiffrement de Vigenère et nous verrons qu'avec les moyens de calcul actuels ce type de cryptosystème est facile à casser.

Chiffrement de Vigenère

Dans un but de simplification, nous ne considérerons que des messages non accentués écrits en lettres majuscules, sans espace ni ponctuation. Par exemple, le message à coder « *Les sanglots longs des violons de l'automne* » sera transmis sous la forme : `LESSANGLOTSLONGSDESVIOLONSDELAUTOMNE`.
Commencez par définir l'alphabet :

```
alph = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

Cet alphabet est identifié à $[\![0, 25]\!]$; ainsi à la lettre `A` correspond la valeur 0, à la lettre `B` la valeur 1, etc. Cette identification induit en outre une opération d'addition (ou de décalage si on préfère) sur les lettres : par exemple, `D + K = N` puisque $(3 + 10) \bmod 26 = 13$, ou encore `J + T = C` puisque $(9 + 19) \bmod 26 = 3$.
Rappel. La méthode `index` appliquée à une chaîne de caractères permet d'obtenir la valeur associée à un caractère présent dans la chaîne de caractères `alph` :

```
In [1]: alph.index('N')
Out[1]: 13

In [2]: alph[13]
Out[2]: 'N'
```

La clef secrète de la méthode de Vigenère est un mot c de longueur ℓ sur cet alphabet. Le message à coder est découpé en blocs b de longueur ℓ et chaque lettre de chacun de ces blocs est décalée de la valeur associée à la lettre de même rang dans la clef secrète.
Par exemple, le chiffrement du message ci-dessus à partir de la clef secrète `VERLAINE` se réalise ainsi :

	L	E	S	S	A	N	G	L	O	T	S	L	O	N	G	S	D	E	S	V	I	O	L	O	N	S	D	E	L	A	U	T	O	M	N	E
+	V	E	R	L	A	I	N	E	V	E	R	L	A	I	N	E	V	E	R	L	A	I	N	E	V	E	R	L	A	I	N	E	V	E	R	L
=	G	I	J	D	A	V	T	P	J	X	J	W	O	V	T	W	Y	I	J	G	I	W	Y	S	I	W	U	P	L	I	H	X	J	Q	E	P

Le message chiffré est donc : `GIJDAVTPJXJWOVTWYIJGIWYSIWUPLIHXJQEP`.

Question 1.

a) Rédiger deux fonctions `add(x, y)` et `sub(x, y)` qui prennent pour arguments deux caractères x et y présents dans la chaîne `alph` et qui renvoient respectivement les caractères $x + y$ et $x - y$.

b) En déduire deux fonctions `chiffre(clef, message)` et `dechiffre(clef, message)` réalisant respectivement le chiffrement et le déchiffrement d'un message à partir d'une clef secrète en utilisant la méthode de Vigenère.

c) Dans le fichier `vigenere.txt` que vous pouvez récupérer sur le site `http://pc-etoile.schola.fr` à la rubrique *informatique - travaux pratiques* se trouvent plusieurs textes qui ont été chiffrés à l'aide de la méthode de Vigenère. Déchiffrer le premier sachant qu'il a été codé à l'aide de la clef `VERLAINE`.

Attaque du chiffrement de Vigenere

Nous allons maintenant nous intéresser aux méthodes qui permettent de « casser » un texte chiffré par la méthode de Vigenère, c'est-à-dire qui permettent de reconstituer le texte initial sans en posséder la clef.
Dans un premier temps, nous allons supposer connue la longueur ℓ de la clef. On sait alors que pour tout $k \in [\![0, \ell-1]\!]$, toutes les lettres dont les indices sont congrus à k modulo ℓ sont décalées d'une même valeur $d_k \in [\![0, 25]\!]$. Notons M_k l'ensemble de ces lettres. Si le message est suffisamment long il y a de fortes chances pour que la valeur de d_k soit celle pour laquelle la fréquence d'apparition des lettres de l'ensemble $\left\{x - d_k \;\middle|\; x \in M_k\right\}$ soit la plus proche possible de la fréquence d'apparition de ces mêmes lettres dans la langue française.

Nous avons donc besoin des fréquences d'apparition des différentes lettres de la langue française.
Le fichier `hugo.txt` à récupérer sur `http://pc-etoile.schola.fr` contient la reproduction du roman « Quatrevingt-Treize » de Victor Hugo. Outre les 26 lettres de l'alphabet ce texte comporte des espaces, des symboles de ponctuation (points, virgules, etc) et des passages à la lignes (le caractère `'\n'`).

Question 2. Utiliser ce document pour créer un tableau `francais` de 26 cases contenant la fréquence d'apparition de chacune des 26 lettres de l'alphabet dans ce roman (consulter la figure 1 pour savoir comment travailler sur le contenu d'un fichier texte en Python).

Pour accéder au contenu d'un fichier `exemple.txt` il faut commencer par l'ouvrir en mode lecture :

```
f = open('exemple.txt', 'r')
```

On crée ainsi un objet que l'on peut énumérer ligne par ligne :

```
for l in f:
    ...
```

donne à `l` la valeur de chacune des lignes de ce document (y compris le caractère `'\n'` en fin de ligne).

On met fin à l'ouverture de ce fichier à l'aide de la méthode `close` :

```
f.close()
```

Figure 1 – Lecture d'un fichier texte.

Nous considérerons désormais que ce tableau correspond à la fréquence d'apparition des lettres dans la langue française.

Question 3. Rédiger une fonction `frequence(texte)` qui prend en argument une chaîne de caractère et retourne un tableau de 26 cases contenant la fréquence d'apparition de chacune des 26 lettres de l'alphabet dans ce texte.

Les tableaux renvoyés par la fonction `frequence` sont destinés à être comparés au tableau `francais` calculé à la question 2. Pour cela, nous allons calculer le *coefficient de corrélation* qui lie ces deux tableaux. Ce dernier se définit de la manière suivante : si $X = [x_0, x_1, \dots, x_{n-1}]$ et $Y = [y_0, y_1, \dots, y_{n-1}]$ sont deux tableaux de valeurs de même longueur alors

$$\text{cor}(X, Y) = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2}\sqrt{\sum_i (y_i - \overline{y})^2}}$$

où $\overline{x}$ et $\overline{y}$ désignent respectivement les moyennes des valeurs contenues dans les tableaux X et Y.
Cette quantité est comprise entre -1 et 1, et plus $\text{cor}(X, Y)$ est proche de 1, plus on peut considérer que les deux tableaux X et Y sont corrélés.

Question 4. Rédiger une fonction `correlation(X, Y)` qui calcule le coefficient de corrélation de deux listes de valeurs X et Y de même taille. On s'efforcera de réaliser ce calcul en n'effectuant qu'un seul parcours en parallèle de ces deux tableaux (penser à la formule de König).

On considère maintenant un message M chiffré par la méthode de Vigenère à l'aide d'une clé de longueur ℓ. Ce message est découpé en ℓ messages $M_0, \ldots, M_{\ell-1}$: chacun de ces messages est l'ensemble des lettres qui ont été codées à l'aide de la même lettre de la clef secrète. En d'autres termes, si `message` est la chaîne de caractère représentant en Python le message M, alors pour tout $k \in [\![0, \ell-1]\!]$ le message M_k est représenté en Python par la chaîne `message[k::l]`.

Chacun de ces messages M_k a été codé par la méthode de Vigenère à l'aide d'une clef de longueur 1. Il est raisonnable de penser que le décalage conduisant à une corrélation maximale avec la fréquence des lettres en français correspond à cette clef, ce qui ne laisse que 26 possibilités à examiner pour chacun des ℓ messages M_k.

Question 5.

a) Rédiger une fonction `clefL(message, l)` qui prend en arguments un message crypté à l'aide d'une clé de longueur ℓ et qui renvoie la clef utilisée pour chiffrer ce message, en suivant la démarche décrite ci-dessus.

b) Déchiffrer le second message du fichier `vigenere.txt` sachant qu'il a été chiffré à l'aide d'une clé de longueur 6.

Enfin, lorsqu'on ne connait pas la longueur ℓ de la clef, on fait l'hypothèse que celle-ci est comprise entre deux valeurs raisonnables, par exemple 4 et 20. Pour chacune de ces valeurs on déchiffre le message de la manière précédente en faisant l'hypothèse que la clef est de longueur ℓ, puis on calcule le coefficient de corrélation avec la langue française du message obtenu. La longueur vraisemblable de la clef correspondra à la valeur maximale de ce coefficient de corrélation.

Question 6.

a) Rédiger une fonction `clef(message)` qui prend en argument un message crypté pour lequel la clef a une longueur comprise entre 4 et 20 et qui renvoie la clef utilisée pour chiffrer ce message.

b) Déchiffrer enfin le troisième message du fichier `vigenere.txt`.