Semestre 2 **Exercices pour TP 4** 2010–2011

Cryptosystèmes classiques

Charger

```
MAT16 = "http://iml.univ-mrs.fr/~kohel/tch/MAT16/"
exec(open(get_remote_file(MAT16 + "MAT16.sage")).read())
```

L'ensemble S et les textes standards en français et en anglais, encodés dans S, se créent avec les commandes suivantes :

```
S = AlphabeticStrings()
fr = S.encoding(open(get_remote_file(MAT16 + "chapitre.1.txt")).read())
en = S.encoding(open(get_remote_file(MAT16 + "blackcat.txt")).read())
```

1. Trouver le chiffrement des textes standards français et anglais avec le chiffrement à substitution de clé `ULOIDTGKXYCRHBPMZJQVWNFSAE`. Par exemple, pour le chiffrement du texte m ci-dessous :

```
Subs = SubstitutionCryptosystem(S)
m = S("LAMAISONBLANCHE")
K = S("ULOIDTGKXYCRHBPMZJQVWNFSAE")
E_K = Subs(K)
c = E_K(m)
print c
```

Comparer les fréquences des caractères dans les textes clairs et chiffrés (en utilisant les textes standards français et anglais pour m). D'abord, définissons des couleurs :

```
(rouge, vert, bleu) = ((0.8,0,0),(0.2,0.5,0.2),(0,0,0.8))
```

puis faisons la comparaison des fréquences des textes :

```
F_m = m.frequency_distribution()
F_m.plot(rgbcolor = bleu).show(aspect_ratio=100)
F_c = c.frequency_distribution()
F_c.plot(rgbcolor = rouge).show(aspect_ratio=100)
```

Quelles proprietés restent inchangées par un chiffrement à substitution ?

2. Trouver le chiffrement du texte standard français avec un chiffrement par transposition avec la clé `[3,5,2,6,1,4]`. Le chiffrement par transpostion découpe le texte en blocs de longueur égale, ici 6, et applique la permutation des lettres données par la clé. Le chiffrement par transposition se crée en Sage avec les commandes suivantes :

```
Trans = TranspositionCryptosystem(S,6)
K = Trans.key_space()([3,5,2,6,1,4])
E_K = Trans(K)
```

Quelles proprietés restent inchangées par un chiffrement par transposition ?

3. Trouver le chiffrement des textes standards avec un chiffrement de Vigenère avec la clé `SECURITE`. Le chiffrement par transpostiion se crée en Sage avec les commandes suivantes :

```
Vigen = VigenereCryptosystem(S,8)
K = S("SECURITE")
E_K = Vigen(K)
```

Que se passe-t-il pour les fréquences des lettres avec un chiffrement de Vigenère ?

Cryptanalyse des textes chiffrés

Pour les exercices ci-dessous, nous prendrons les textes chiffrés suivants :

```
ct1 = S(open(get_remote_file(MAT16 + "cipher_text.1.txt")).read())
ct2 = S(open(get_remote_file(MAT16 + "cipher_text.2.txt")).read())
ct3 = S(open(get_remote_file(MAT16 + "cipher_text.3.txt")).read())
```

Les graphes des fréquences des lettres de ces trois chiffrés et la comparaison avec ceux en français et en anglais sont donnés ci-dessous :

```
en.frequency_distribution().plot(rgbcolor=rouge,aspect_ratio=100)
fr.frequency_distribution().plot(rgbcolor=bleu,aspect_ratio=100)
ct1.frequency_distribution().plot(rgbcolor=vert,aspect_ratio=100)
ct2.frequency_distribution().plot(rgbcolor=vert,aspect_ratio=100)
ct3.frequency_distribution().plot(rgbcolor=vert,aspect_ratio=100)
```

4. Identifier les textes chiffrés par un chiffrement de Vigenère, à substitution et par transposition.

5. L'indice de coïncidence est la probabilité de répétition de lettres choisis au hasard dans un texte. Pour un texte aléatoire sans biais, l'indice de coïncidence est $1/26 = 0,03846\ldots$.

```
ct1.coincidence_index()
ct2.coincidence_index()
ct3.coincidence_index()
```

Quels chifferements sont compatibles avec ses indices de coïncidence ? Est-ce c'est plus probable que leur texte clair soit en anglais ou en français ?

Cryptanalyse des chiffrements de Vigenère

Supposons que $\texttt{ct} = c_0c_1\ldots$ est un texte chiffré par un chiffrement de Vigènere. Pour chaque $0 \leq i < r$, on peut calculer les partitions

$$c_ic_{i+r}c_{i+2r}\ldots$$

avec la syntaxe `ct[i::r]` en Python/Sage. Alors l'indice de coïncidence moyen de toutes les partitions se trouve facilement :

```
ct = ct1
for r in range(1,20):
    print "%2s: %s"%(r, sum([ ct[0::r].coincidence_index() ])/r)
```

Pour la bonne longueur r, l'indice de coïncidence sera égal à celui du français ou de l'anglais.

6. Trouver la période pour le texte chiffré par un chiffrement de Vigenère.
Indication.
```
AZ = S.gens() # l'alphabet {A,B,..,Z}
ct1[0::13].frequency_distribution().plot(labels=AZ)
```

Cryptanalyse des chiffrements par transpositions

Supposons que $\texttt{ct} = c_0c_1\ldots$ est un texte chiffré par un chiffrement par transposition. Pour $I = (i, j)$, nous formons la suite de couples de lettres

$$c_ic_j, c_{i+r}c_{j+r}, c_{i+2r}c_{j+2r}, \ldots$$

avec la fonction `decimation(ct,[i,j],2)`. Les positions (i, j) tel que $c_{i+rk}c_{j+rk}$ semble suivre les probabilités d'un texte clair peut être trouver avec une analyse de la correlation :

```
ct = ct3
F2 = fr.frequency_distribution(2)
i = 0
for r in range(2,20):
    max_corr = -1
    for j in range(r):
        ct_ij = decimation(ct,[i,j],r)
        F2_ij = frequency_distribution(ct_ij)
        X2_ij = DiscreteRandomVariable(F2, F2_ij.function())
        max_corr = max(max_corr, F2.correlation(X2_ij))
    print "%2s: %s" % (r, max_corr)
```

Alors pour la bonne période, les positions (i, j) tel que $c_{i+rk}c_{j+rk}$ semble suivre les probabilités d'un texte clair peuvent être trouver avec une analyse des corrélations :

```
for i in range(r):
    for j in range(r):
        ct_ij = decimation(ct,[i,j],r)
        F2_ij = frequency_distribution(ct_ij)
        X2_ij = DiscreteRandomVariable(F2, F2_ij.function())
        corr = F2.correlation(X2_ij)
        if corr > 0.65:
            print "(%2s, %2s): %s" % (i,j,F2.correlation(X2_ij))
```

7. Trouver la période pour le texte chiffré par un chiffrement par transposition.

8. Pour le chiffrement par transposition, trouver la suite des indices (i, j) qui étaient adjacents dans le texte clair.