Cryptographie : laboratoire 1

Auteur : Rayane Annen

Réponses aux questions

1 - Quel est l'avantage d'utiliser le test du χ^2 plutôt que de comparer simplement la lettre la plus fréquente dans le texte chiffré par rapport aux statistiques du langage de base ?

Le test nous permet de déterminer si la distribution des lettres dans le chiffré est significativement différente de la distribution de référence. On identifie alors les lettres qui sont utilisées de manière disproportionnée dans le chiffré par rapport à la distribution de référence.

Utiliser seulement la lettre la plus fréquente du chiffré est trompeur puisque celle-ci peut être différente de la langue de base étant donnée qu'on procède à un décalage des lettres.

Le test du χ^2 nous permet de prendre en compte l'ensemble des observations.

2 - Pourquoi est-ce que le test du χ^2 ne fonctionne-t-il pas directement sur un texte chiffré à l'aide du chiffre de Vigenère ?

Contrairement au chiffre de César, le chiffre de Vigenère change la fréquence d'apparition des lettres (du moins ne fait pas qu'un décalage de la distribution c.f. questions 4), les fréquences étant différentes de celles du texte de base effectuer le test ne nous apprendra pas grand chose.

3 - Que mesure l'indice de coïncidence ?

L'indice de coïncidence permet de déterminer la probabilité que si nous tirons 2 lettres au hasard dans un texte que celles-ci soit les mêmes. Il permet de savoir si un texte a été chiffré par un chiffre monoalphabétique (la clef est une seule lettre comme César) ou bien par un chiffre polyalphabétique (comme Vigenère, clef formée de plusieurs lettres)

4 - Pourquoi est-ce que l'indice de coïncidence n'est-il pas modifié lorsque l'on applique le chiffre de César généralisé sur un texte ?

On ne fait que de décaler toutes les lettres d'un même indice. La distribution (fréquences d'apparition) de nos lettres reste la même, c'est juste qu'on décale (cycliquement) de *indice* la distribution, e.g. la fréquence d'apparition de a devient celle de b, celle de b devient celle de c, ainsi de suite. Par conséquent la probabilité de tirer deux fois la même lettre d'affilé reste la même, la probabilité étant indépendante de la lettre en elle même, c'est pourquoi l'IC reste le même.

5 - Est-il possible de coder un logiciel permettant de décrypter un document chiffré avec le chiffre de Vigenère et une clef ayant la même taille que le texte clair ? Justifiez.

Non sous la condition que la clef soit utilisée qu'une seule fois. Si la clef tirée est uniformément aléatoire, on est face à un chiffre de Vernam, qui est considéré comme parfaitement sûr. La probabilité de voir apparaître l'une ou l'autre lettre dans le chiffré est de $1/26$, en sachant que ces lettres sont indépendantes les unes des autres, il est irréaliste de penser pouvoir décrypter un tel texte.

6 - Expliquez votre attaque sur la version améliorée du chiffre de Vigenère.

En résumé, sur la version améliorée, le but est de renverser les chiffrements de César (toujours généralisé) successifs puis finalement appliquer l'algorithme pour casser le chiffre de Vigenère. La difficulté est qu'il faut faire cela pour toutes les tailles de clefs possibles de Vigenère et toutes les clefs de César. On doit alors chercher le candidat le plus probable avec l'indice de coïncidence.

Pour une la longueur de clef de Vigenère $l \in \{1,\ 2,\ \ldots,\ \text{max\_length}\}$ on fait une partition de notre chiffré en l segments, on va appliquer sur tous les segments un déchiffrement de César de clef $k \in \{0,\ 1,\ \ldots,\ 25\}$, ainsi pour chaque couple de clefs (l, k) on obtient un texte qui est uniquement chiffré avec Vigenère. On annule donc la complexité supplémentaire des chiffrements successifs de César.

Ensuite pour chaque couple de clefs, on va effectuer, ce que j'appelle une partition en colonne de notre texte chiffré uniquement avec Vigenère, c'est à dire les segments ayant leurs lettres aux positions $0, l, 2l, 3l$ puis $1, l+1, 2l+1$, etc.

Visuellement pour le texte "ABCDEFG", si $l = 2$, on aurait :

```
AB
CD
EF
G
partition = [ACEG, BDF]
```

Ensuite similairement au cassage de Vigenere, nous allons calculer la moyenne des indices de coïncidences appliqués à chaque segments.

Cette valeur est sauvegardée afin de déterminer quel texte possède l'indice le plus proche à celui de notre texte de référence.

Finalement au bout de max_length $\times$ 25 itérations nous avons trouvé le meilleur candidat pour notre clef de Vigenère.

Pour la trouver, on applique simplement la dernière étape de notre cassage de l'algorithme de Vigenère (la partie recherche du meilleur texte candidat étant déjà faite)

On reconstruit notre clef de Vigenère en appelant `caesar_break` sur chaque colonnes de la partition faite précédemment (dans l'exemple d'avant ce serait `ACEG` et `BDF`).

Finalement, on retourne notre clef de Vigenère reconstruite et la clef de César récupérée dans l'étape de la recherche du candidat.

7 - D'où proviennent les textes clairs correspondants aux fichiers `vigenere.txt` et `vigenereAmeliore.txt`

Le premier vient directement du site suivant : https://heig-vd.ch/campus/vie-sur-le-campus/manifestations/innovation-crunch-time. C'est le texte introduction à l'innovation CRUNCH.

Le second est une citation tirée d'une scène culte du film *Astérix et Obélix : Mission Cléopâtre* quand Astérix, Obélix et Panoramix rencontrent le scribe Otis et lui posent la question : *C'est une bonne situation ça scribe ?*