

TP N° 1 Cryptographie Artisanale

N.B Les solutions proposées doivent être écrites en langage Python.

Exercice 1: Chiffre de César

1. Écrire une fonction qui prend en paramètre une chaîne de caractère `ch` ainsi qu'un nombre de décalage `k` (entre 1 et 25) et retourne la chaîne `ch` cryptée par le chiffre de César de décalage `k`.
2. En déduire un programme qui lit une chaîne de caractères `ch` et lui applique le cryptage et le décryptage en affichant à chaque étape le résultat.
3. Écrire un programme qui cryptanalyse un message intercepté supposé chiffré par César :
 - a. En testant tous les décalages possibles (attaque par force brute),
 - b. En utilisant une attaque statistique.

Exercice 2: Chiffrement par Substitution Mono-Alphabétique

1. Écrire une fonction qui génère aléatoirement une table de substitution mono-alphabétique `Tab` (Attention : éviter les répétitions)

Ex :

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

2. Écrire un programme qui lit un fichier texte (`.txt`) et crypte son contenu par un chiffrement par substitution mono-alphabétique via la table `Tab` et range le résultat dans un autre fichier text.
3. Écrire une fonction qui calcule la fréquence d'apparition des lettres dans un texte et range le résultat pour un traitement ultérieur. (Ne pas compter les espaces)
4. En se basant sur la table de fréquences de la langue utilisée pour la rédaction du texte, tenter de cryptanalyser un texte intercepté (chaîne de plus de 500 caractères) en supposant qu'il a été crypté par substitution mono-alphabétique.

Exemple approximatif pour la langue française :

E	S	A	I	N	T	R	U	L	O	D
14.69%	8.01%	7.54%	7.18%	6.89%	6.88%	6.49%	6.12%	5.63%	5.29%	3.66%

N.B Vous pouvez utiliser un texte en anglais, en se référant à la table de fréquences appropriées.

Exercice 3: Chiffre Homophone

1. Écrire une fonction qui génère aléatoirement un tableau de substitution `Tab` de 0 à 99 (voir cours). Attention : éviter les répétitions.
2. Écrire une fonction qui crypte un texte par un chiffre homophone en se basant sur la table `Tab`. Cette fonction doit choisir d'une manière uniforme les choix de substitution.
 i.e : ne pas choisir un nombre une deuxième fois avant que tous les autres nombres soient choisis au moins une fois.
 Dans l'exemple du cours pour la lettre D, les choix possibles sont 01, 03, 45, 79
3. Refaire les questions 1. et 2. :
 - a. En utilisant un carré de Polybe (initialisé manuellement ou aléatoirement)
 - b. En utilisant l'alternative 2 (voir cours)

Exercice 4 : Chiffrement de Vigenère

1. Implémentation du chiffrement et du déchiffrement

- Écrire une fonction qui chiffre un texte clair à l'aide du chiffre de Vigenère en prenant comme paramètres le texte clair et la clé alphabétique.
- Écrire une fonction permettant de déchiffrer un texte chiffré à partir de la clé.

2. Cryptanalyse du chiffre de Vigenère

a. Méthode de Kasiski

- La méthode de Kasiski permet d'estimer la longueur de la clé. Elle consiste à :
 - Repérer des séquences de lettres répétées dans le texte chiffré (LS=3,4,5,...),
 - Mesurer les distances entre leurs occurrences,
 - déterminer les diviseurs communs de ces distances, qui sont de bons candidats pour la longueur de la clé.

Réf: https://fr.wikipedia.org/wiki/Cryptanalyse_du_chiffre_de_Vigen%C3%A8re

b. Affinement avec l'Indice de Coïncidence (IC)

- Pour chaque longueur candidate (L) trouvée par Kasiski :
 1. Découper le texte chiffré en L sous-textes :
 - Sous-texte 1 = lettres aux positions 0, L, 2L, ...
 - Sous-texte 2 = lettres aux positions 1, L+1, 2L+1, ...
 - ... jusqu'au sous-texte L.
 2. Calculer l'IC de chaque sous-texte :
 - IC proche de 0,065 : longueur probable correcte (texte français/anglais),
 - IC ~ 0,038 : longueur improbable (texte proche de l'aléatoire).
 3. Retenir la longueur dont la moyenne des IC est la plus proche de 0,065.

c. Récupération de la clé et déchiffrement

- Appliquer une analyse fréquentielle sur chaque sous-texte pour retrouver la lettre correspondante de la clé.
- Reconstituer la clé complète.
- Déchiffrer le texte intercepté.

3. Étude de l'impact de la taille de la clé sur la sécurité

- Chiffrer le même texte avec plusieurs clés de tailles différentes (par exemple : 3, 6, 10, 15 caractères...).
- Appliquer la méthode de Kasiski et l'analyse fréquentielle dans chaque cas.
- Montrer expérimentalement que plus la clé est longue, plus la cryptanalyse devient difficile, voire imprécise ou inefficace pour des textes de taille limitée.

Exercice 5: Chiffre de Hill (m = 2)

1. Écrire une fonction qui retourne l'inverse d'un nombre n dans $\mathbb{Z}/26\mathbb{Z}$ s'il existe et 0 sinon.
2. Écrire une fonction qui génère une matrice $Q(m,m)$ inversible dans $\mathbb{Z}/26\mathbb{Z}$.
3. Écrire une fonction qui crypte un texte via le chiffre de Hill basé sur la matrice Q .
4. Écrire une fonction qui décrypte un texte via le chiffre de Hill basé sur la matrice Q .
5. Écrire un programme qui illustre les questions de 1. à 4.
6. Cryptanalyse (attaque par clair connu)

En supposant que l'attaquant connaît :

- Au moins deux digrammes en clair et leurs chiffrés correspondants,
- Et que le chiffrement a été effectué avec une matrice 2×2 ,

Écrire un programme qui :

- Retrouve la matrice de chiffrement Q ,
- Vérifie sa validité (inversible modulo 26),
- Et déchiffre un message intercepté.