


TP 1 : Cryptographie classique : César

Table of Contents

- 1. Chiffrement et déchiffrement
 - 1.1. Éléments techniques
- 2. Exercice 1 : chiffrer et déchiffrer
- 3. Exercice 2 : décrypter

Les objectifs de ce TP sont de :

- Étudier le chiffre le plus connu de la cryptographie classique
- Réaliser une attaque simple sur un chiffré
- Renforcer vos compétences en python et jupyter



JuPyTeR

Tous les exercices sont à faire sur le serveur JuPyteR :
<https://jupyterhub.iut-blagnac.fr/>

Votre login est de la forme `prenom.nom1` (avec **1 à la fin**) et le mot de passe (à changer et mémoriser !) est `Etud14nt`

Il vous est vivement recommandé d'utiliser un gestionnaire de mot de passe.

1. Chiffrement et déchiffrement

Jules César utilisait le système suivant pour communiquer secrètement : chaque lettre de l'alphabet était décalée de 3 unités (3 est donc la **clé privée**).

C'est un chiffre par **substitution monoalphabétique** (chaque lettre est remplacée par une autre mais cela pourrait être n'importe quel symbole).

C'est à dire : a donne d, b donne e, ... , w donne z, x donne a, y donne b et z donne c.

Pour déchiffrer... c'est l'inverse.

1.1. Éléments techniques



Éléments techniques pour le chiffrement : méthode 1

On peut utiliser le code ASCII des lettres et ajouter/soustraire le décalage

Pour parcourir une chaîne de caractères en python :

```
texte="mon texte que je dois analyser"
for lettre in texte:
    print(lettre)
```

```
ord('A') # renvoie 65
chr(65) # renvoie 'A'
```



Éléments techniques pour le chiffrement : méthode 2

On peut aussi utiliser le *slicing* et la concaténation de listes en python :

```
texte="abcdefghij"
texte[2:6] # renvoie "cdef"
texte[:3] # renvoie "abc"
texte[3:]+texte[:3] # renvoie "defghijabc"
```

2. Exercice 1 : chiffrer et déchiffrer

Cahier des charges : le texte clair (copier/coller depuis Internet), sera placé dans un fichier texte. Votre `notebook` devra à la fin afficher le texte chiffré correspondant.

Après avoir lu les éléments techniques ci-dessus, écrire un `notebook` permettant de :


- charger un texte depuis un fichier texte brut
- qui utilise la fonction de "nettoyage" fournie ci-dessous afin d'avoir un texte sans espaces, ponctuations où tout sera mis en majuscule pour simplifier

```
def supprime_accents_minuscules_ponctuations(message):
    """
    Supprime accents, ponctuations et
    met tout en majuscules
    """
    accents = {'a': ['à', 'ä', 'á'],
               'e': ['é', 'è', 'ê'],
               'o': ['ô', 'ó', 'ò', 'ö'],
               'u': ['ù'],
               ' ': [' ', ':', ',', '-', '(', ')', "'", '"', '.'],
               }

    for (lettre_sans_accent, lettres_avec_accents) in accents.items():
        for lettre in lettres_avec_accents:
            message = message.replace(lettre, lettre_sans_accent)
    return message.upper()
```

- Calculer le chiffré de ce texte avec une clé de 3 (modifiable) et le faire afficher sur le `notebook`
- Le déchiffrement est évident. Testez que votre chiffrement soit bien fonctionnel !

3. Exercice 2 : décrypter



<https://chiffrer.info/>

Décrypter consiste à retrouver le texte original à partir d'un message chiffré **sans posséder la clé** de (dé)chiffrement. Décrypter ne peut accepter d'antonyme : il est en effet impossible de créer un message chiffré sans posséder de clé de chiffrement.

Nous allons voir le premier type d'attaque en cryptographie : l'attaque par texte chiffré (*cyphertext-only attack*) qui peut ici se décliner en deux modalités :

- recherche exhaustive (*brute force*) : ici il y a 25 clés possibles, c'est envisageable mais peu élégant même si on finit toujours par y avoir recours...
- attaque par mot probable
- analyse fréquentielle

Sur votre `notebook` , réaliser le nécessaire pour **décrypter** un message secret. On utilisera une attaque par fréquence et on supposera que le texte clair a été chiffré avec César. Pour cela :

- importer votre texte chiffré de l'exercice précédent (depuis le notebook lui-même ou fichier)
- mener une analyse fréquentielle
- déduire la valeur de la clé
- retrouver le texte clair.