

TPIII-4 Sécurisation des communications

I. Chiffrement symétrique

Le message est chiffré à l'aide d'une clé. Il est décodé avec la même clé. Cela implique la communication de la clé de chiffrement entre les deux machines qui échangent des informations cryptées.

I.1. Code César

Dans le principe, on décale l'alphabet d'un certain nombre de lettres (ici 7)

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

Ainsi CESAR ATTAQUERA A TROIS HEURES se code JLZHY HAAHXBLYH H AYPVZ OLBYLZ

Le décodage utilise la même table qui peut aussi être présentée comme ceci :

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

I.1.a. Programmation d'une fonction d'encodage et de décodage avec clé de décalage connue

✎ **Ecrire une fonction code** qui prend en paramètre un message m et un décalage d et renvoie le message codé c. Les espaces seront codés par un espace.

```
def code(m,d) :
    m=m.upper()
    c=""
    ....
    return c
```

Informations :

La méthode upper() d'un objet string le convertir en majuscule.
 Les codes ascii des lettres A à Z sont 65 à 90
 Pour avoir le code ascii d'une lettre : ord("E") renvoie 69
 Pour obtenir la lettre de code ascii : chr(78) renvoie "N"
 N'oubliez pas le rôle du modulo a%b qui renvoie le reste de la division euclidienne de a par b

✎ **Ecrire une fonction decode** qui prend en paramètre un message codé c et le décalage d qui la codé et renvoie le message m en clair. Les espaces seront codés par un espace.

```
def decode(c,d) :
    m=""
    ....
    return m
```

I.1.b. Crackage du code par force brute (clé inconnue)

Il n'y a que 25 décalages possibles donc on peut tester rapidement ces 25 décalages avec la fonction decode et choisir le résultat qui a un sens en français : c'est un algorithme de **force brute**.

✎ **Ecrire une fonction brute1** qui prend en paramètres un message codé c et qui renvoie la liste des 25 décodages possibles

```
def brute1(c) :
    liste=[]
    ....
    return liste
```

On utilisera la fonction decode programmée précédemment.

On dispose d'un ensemble dico de mots courants tirés d'un dictionnaire **francais.text**.

```
#création de l'ensemble des mots courant du français
with open("francais.txt", encoding="utf-8") as file:
    dico = set(line.strip().upper() for line in file)
```

et d'une fonction indexMax qui renvoie l'index du maximum d'une liste

✎ **Ecrire une fonction brute2** qui prend en paramètres la liste des 25 décodages possibles et affiche le message qui a le plus de sens et la clé de décalage correspondante.

```
def brute2(liste) :
```

```
....
```


Information : Pour épeler les mots d'une phrase, on utilise la méthode split()

I.1.c. crackage du code par analyse fréquentielle (clé inconnue)

Le tableau suivant donne la fréquence d'utilisation des lettres dans les mots français par ordre décroissant.
 frequency=['E','S','A','T','N','T','R','U','L','O','D','C','P','M','V','Q','G','F','H','B','X','J','Y','Z','K','W']

Ainsi si le texte est suffisamment long la lettre la plus fréquente sera le E et la moins fréquente le W.

On va ainsi pouvoir décoder un codage César en cherchant la lettre la plus fréquente dans le texte codé. Elle devrait correspondre à un E dans le message en clair.

 **Ecrire une fonction fréquentiel** qui prend en paramètre un message codé c qui affiche la clé de décalage et le message en clair.
 On utilisera la *fonction decode* précédente.

Message crypté pour tester toutes vos fonctions de décodage

X GZ PQE BXGE SDMZPE PQRUE QF X GZQ PQE BXGE SDMZPQE DQMXUEMFUAZE P MXMZ FGDUZS MGDM QFQ
 PQ FDAGHQD GZQ YQFTAPQ PQ PQODKBFMSQ PG PUEBAEUFUR MXXQYMZP QZUSYM OQXGU OU MHMUF QFQ
 UZHQZFQ HUZSF MZE BXGE FAF BMD X UZSQZUQGD QXQOFDUOUQZ MXXQYMZP MDFTGD EOTQDNUGE P
 MBDQE GZ NDQHOF PQ X UZHQZFQGD ZQDXMZPMUE TGSA WAOT PQBAEQ QZ WEWE M XM NMEQ PQBAEQQ
 OAYYQ YMOTUZQ OUHUXQ QXQOFDAYQOMZUCGQ BAGD ODKBFQD XQE YQEEMSQ OAYYQDOUMGJ QXXQ
 RAZOFUAZZMUF QZ QZFDMZF EGD GZ OXMHUQD EQYNXMNXQ M GZQ YMOTUZQ M QODUDQ PQE XQFFDQE
 PQ RMQAZ FAGF M RMUF OXMEEUCGQ GZQ FAGOTQ BQDYQFFMUF QZEGUFQ P QZHAKQD GZ OAGDMZF
 QXQOFDUCGQ BMDOAGDUD XQ OMNXMSQ PQ DAFAD EGD GZQ EQDUQ PQ PUECGQE QZ YQFMX BQDYQFFMZF P
 QZOAPQD XQ YQEEMSQ QZ FDMZERADYMZF XM XQFFDQ UZUFUMXQ QZ GZQ ZAGHQXXQ XQFFDQ

II. Chiffrement asymétrique

Le message est crypté avec une clé publique que tout le monde connaît.

Le message est décrypté avec la clé privée correspondant à la clé publique que seul le destinataire connaît.