Compte Rendu TP 1

Module : cryptographie

YOUSSEF HACHIMI

AHMED YOUSRI

Objectifs du TP

1. Comprendre le fonctionnement des cryptosystèmes classiques (César et Vigenère).

2. Implémenter ces cryptosystèmes en Python pour chiffrer et déchiffrer des

3.Réaliser une cryptanalyse du chiffre de César en utilisant l'analyse fréquentielle

Exercice 1: Chiffre de César

Le chiffre de César est une méthode de substitution où chaque lettre d'un texte clair est

de 3, A→D,B→E,...,Z→C.

décalée d'un certain nombre de positions dans l'alphabet. Par exemple, avec un décalage

- Implémenter le chiffre de César en suivant les étapes suivantes :

1. Demandez à l'utilisateur un message à chiffrer.

2. Demandez une clé (décalage).

3. Chiffrez le message.

4. Déchiffrez le message à partir du texte chiffré et de la clé.

tp1.py

```python
print("########## Chiffre de César ##########")

letters = 'abcdefghijklmnopqrstuvwxyz'


# --- Fonction de CHIFFREMENT ---
def chiffre(message, key):
    ciphertext = ""
    for letter in message:
        letter_lower = letter.lower()

        if letter_lower in letters:
            index = letters.find(letter_lower)
            new_index = (index + key) % 26
            new_letter = letters[new_index]

            if letter.isupper():
                new_letter = new_letter.upper()

            ciphertext += new_letter
        else:
            ciphertext += letter
    return ciphertext
```

tp1.py

```python
# --- Fonction de DÉCHIFFREMENT ---
def dechiffre(ciphertext, key):
    plaintext = ""
    for letter in ciphertext:
        letter_lower = letter.lower()

        if letter_lower in letters:
            index = letters.find(letter_lower)
            new_index = (index - key) % 26
            new_letter = letters[new_index]

            if letter.isupper():
                new_letter = new_letter.upper()

            plaintext += new_letter
        else:
            plaintext += letter
    return plaintext
```

tp1.py

```python
# menu
print(" Que voulez-vous faire ?")
print("1 - Pour Chiffrer un message ")
print("2 - PourDéchiffrer un message")

choice = input("entrer votre choix (1/2) : ")

message = input("Entrer le message : ")
key = int(input("Entrer la clé doit etre entre 1 et 26 : "))

if choice == "1":
    result = chiffre(message, key)
    print("\n Message chiffré :", result)

elif choice == "2":
    result = dechiffre(message, key)
    print("\n Message déchiffré :", result)

elif choice > "26" or choice < "1" :
    print("\n Choix invalide.")
```

```
########## Chiffre de César ##########
 Que voulez-vous faire ?
1 - Pour Chiffrer un message
2 - PourDéchiffrer un message
entrer votre choix (1/2) : 1
Entrer le message : je suis etudiant bachelor a EST de beni mellal
Entrer la clé doit etre entre 1 et 26 : 8

 Message chiffré : rm acqa mbclqivb jikpmtwz i MAB lm jmvq umttit
```

```
########## Chiffre de César ##########
 Que voulez-vous faire ?
1 - Pour Chiffrer un message
2 - PourDéchiffrer un message
entrer votre choix (1/2) : 2
Entrer le message : rm acqa mbclqivb jikpmtwz i MAB lm jmvq umttit
Entrer la clé doit etre entre 1 et 26 : 8

 Message déchiffré : je suis etudiant bachelor a EST de beni mellal
```

Exercice 2: Chiffre de Vigenère

Le chiffre de Vigenère utilise une clé composée de lettres. Chaque lettre de la clé détermine un décalage pour une lettre correspondante dans le texte clair.

- Implémenter le chiffre de Vigenère en suivant les étapes suivantes :

1. Demandez un message à chiffrer.

2. Demandez une clé (sous forme d'un mot).

3. Chiffrez le message avec la clé.

4. Déchiffrez le message avec la même clé.

py vigenere.py

```
print(" ################  EX 2 : Chiffre de Vigenere ################  ")

text = input("Entrer le message à chiffrer : ").lower()
cle = input("Entrer la clé : ").lower()
m = len(cle)

# modifier len de cle
for i in range(len(cle),len(text)):
    cle += cle[i % m]


# chiffrement c = (p+k) % 26
crypte = ""
for i in range(len(text)):
    C = ((ord(text[i]) - ord("a")) + (ord(cle[i]) -ord("a"))) % 26
    crypte += chr(C + ord("a"))
print("Voila le texte chiffrer est : ", crypte.lower())


# dechiffrement P = (c-k) % 26
decrypt = ""
for i in range(len(crypte)):
    C = ((ord(crypte[i]) -ord("a")) - (ord(cle[i]) -ord("a"))) % 26
    decrypt += chr(C + ord("a"))

print(" Voila le texte clair est : ", decrypt.lower())
```

```
 ###############  EX 2 : Chiffre de Vigenere ###############
Entrer le message à chiffrer : il ya un match entre maroc et frence
Entrer la clé : foot
Voila le texte chiffrer est :  nzbrfbigsaomhvbxshfxsaoktqbxybtkjbqx
 Voila le texte clair est :  ilnyanunnmatchnentrenmarocnetnfrence
```

Exercice 3 : Cryptanalyse du chiffre de César par analyse fréquentielle

L'analyse fréquentielle est une technique de cryptanalyse basée sur la fréquence des lettres dans une langue donnée. Par exemple, en français, la lettre "E" est la plus fréquente.

- Réaliser cryptanalyse du chiffre de César en suivant les étapes suivantes :

1. Chiffrez un texte clair avec une clé tirée de façon aléatoire.

2. Effectuez une analyse fréquentielle sur le texte chiffré.ni

Devinez la clé en utilisant la fréquence la plus élevée (correspondant à la lettre "E").

Déchiffrez le texte avec la clé trouvée.

tp1-ex3.py

```python
import random
from collections import Counter

# 1. Chiffrement de text avec une clé aléatoire

def chiffrerA(texte, cle):
    resultat = ""
    for c in texte.upper():
        if c.isalpha():
            # Décalage dans l'alphabet
            resultat += chr((ord(c) - ord("A") + cle) % 26 + ord("A"))
        else:
            resultat += c
    return resultat

# 2 Analyse fréquentielle

def analyse_freq(texte):
    lettres = [c for c in texte if c.isalpha()]
    freq = Counter(lettres)
    return freq
```

tp1-ex3.py

```python
def dechifrement(code ,cle_trouver):
    firstResult = ""
    for c in code.upper():
        if c.isalpha():
            # Décalage dans l'alphabet
            firstResult += chr((ord(c) - ord("A") - cle_trouver[0]) % 26 + ord("A"))
        else:
            firstResult += c
    secendResult = ""
    for c in code.upper():
        if c.isalpha():
            # Décalage dans l'alphabet
            secendResult += chr((ord(c) - ord("A") - cle_trouver[1]) % 26 + ord("A"))
        else:
            secendResult += c
    print(firstResult)
    print(secendResult)
```

tp1-ex3.py

```python
msg = input("Entrer le texte en clair : ")
cle = random.randint(0, 25)

print(cle)

code = chiffrerA(msg,cle)
freq = analyse_freq(code)

# 3. Deviner la clé (on suppose que la lettre la plus fréquente représente "E")
def newCle(freq):
    lettre_freq = freq.most_common(1)[0][0]  # lettre la plus fréquente
    clea = (ord(lettre_freq) - ord('E')) % 26
    cleb = (ord(lettre_freq) - ord('A')) % 26
    return clea, cleb

cle_trouver = newCle(freq)
print(cle_trouver)

dechifrement(code,cle_trouver)
```

Exextion de code :

```
Entrer le texte en clair : comprendre le fonctionnement des cryptosystemes classiques
21
(21, 25)
COMPRENDRE LE FONCTIONNEMENT DES CRYPTOSYSTEMES CLASSIQUES
YKILNAJZNA HA BKJYPEKJJAIAJP ZAO YNULPKOUOPAIAO YHWOOEMQAO
PS C:\Users\Youssef Hachimi\AppData\Local\Programs\Microsoft VS Code>
```