

FDS-UEH



# D3v0!r d3 S3cur!t3

Préparé par:  
**ALTIDOR Jean-Bernard T.**  
**DUBUCHE Kevin J.**  
**THEODORE Barbara G.**  
[  
Section: Génie Electronique  
Année : 3ième  
Professeur: M. Vladimyr MATHIEU  
]

Décembre 2020

## Table des matières

Enoncé du devoir.....	3
Introduction.....	4
Encrypter un texte avec le chiffrement de Vigenère.....	4
Décrypter un texte connaissant la clé.....	7
Décrypter un texte sans la clé.....	8
Server.....	10
Client.....	12

*Devoir de Sécurité*  
**Vigenère Cypher**

**Enoncé :**

*I- Implémenter en python une fonction permettant d'encrypter un texte avec le chiffrement de Vigenère.*

*II- Implémenter en python une fonction permettant de décrypter un texte codé avec le chiffrement de Vigenère en connaissant la clé.*

*III- Implémenter en python une fonction permettant de décrypter un texte codé avec le chiffrement de Vigenère sans avoir la clé au préalable.*

*IV- Implémenter un système peer to peer utilisant le chiffrement de Vigenère pour communiquer.*

## Introduction

Le chiffre de Vigenère est un système de chiffrement par substitution polyalphabétique mais une même lettre du message clair peut, suivant sa position dans celui-ci, être remplacée par des lettres différentes, contrairement à un système de chiffrement mono alphabétique comme le chiffre de César.

## I-Implémenter en python une fonction permettant d'encrypter un texte avec le chiffrement de Vigenère.

C'est une fonction qui prend trois paramètres :

- 1- le texte à coder
- 2- la clé de chiffrement
- 3- un caractère : 'e' pour encryption

Il s'agit d'ajouter le décalage correspondant au caractère se trouvant à la position i de la clé (se répétant de manière cyclique).

```
#Vignere code 2.0 ,can handle all ASCII symbol
import os

def vignere(txt="", key="", typ=""):
    if not txt:
        print ("Veuillez entrer un text.")
        return
    if not key:
        print ("Veuillez fournir une cle.")
        return
    if typ not in ('d', 'e'):
        print ("""Veuillez choisir "d" pour decryption ou "e" pour encryption""")
        return

    key_to_int = [ord(i) for i in key]
    txt_to_int = [ord(i) for i in txt]
    resultat = ""
    for i in range(len(txt_to_int)):
        shift =key_to_int[i % len(key)]
        if typ == 'd':
            shift *= -1

        v = (txt_to_int[i] + shift) % 240

    resultat += chr(v )

    return (resultat)
# Driver code
```

```
if __name__ == "__main__":
    texte = input("Veuillez choisir le fichier a encrypter: ")
    try :
        dirname = os.path.dirname(__file__)
        filename = os.path.join(dirname, texte)
        with open(filename, 'r') as file:
            string = file.read().replace('\n', ' ').replace("''''''", " ")
        except FileNotFoundError:
            print ("Erreur d'ouverture de fichier.")
            quit()
```

```
key = input("Veuillez entrer la cle : ")
q = vignere(string, key, 'e')
try :
    dirname = os.path.dirname(__file__)
    filename = os.path.join(dirname, "Cypher.txt")
    with open(filename, 'w') as file:
        string = file.write(q)
    except FileNotFoundError:
        print ("Erreur d'ouverture de fichier.")
    print("Votre message a ete encrypte avec succes! Veuillez verifier le fichier Cypher.txt")
```

The screenshot shows the Visual Studio Code interface with the file 'VignereCypher\_2.py' open. The code in the editor matches the snippets provided above. The terminal at the bottom shows the execution of the script using Python 3.8.5. The user is prompted to choose a file to encrypt ('Betes.txt') and enter a key ('we4'). The script successfully encrypts the message and prints a confirmation message.

```
Kevin@Kevin-Inspiron-3583:~/Desktop/Security_Lab-master$ /usr/bin/python3 /home/kevin/Desktop/Security_Lab-master/Vignere/VignereCypher_2.py
Veuillez choisir le fichier a encrypter: Betes.txt
Veuillez entrer la cle : we4
Votre message a ete encrypte avec succes! Veuillez verifier le fichier Cypher.txt
Kevin@Kevin-Inspiron-3583:~/Desktop/Security_Lab-master$
```

## II- Implémenter en python une fonction permettant de décrypter un texte codé avec le chiffrement de Vigenère en connaissant la clé.

C'est une fonction qui prend trois paramètres :

- 1- le texte encodé
- 2- la clé de chiffrement
- 3- un caractère : 'd' pour décrypter

```
#Vignere code 2.0 ,can handle all ASCII symbol
import os

def vignere(txt="", key="", typ=""):
if not txt:
print ("Veuillez entrer un text.")
return
if not key:
print ("Veuillez fournir une cle.")
return
if typ not in ('d', 'e'):
print ("Veuillez choisir 'd' pour decryption ou 'e' pour encryption")
```

```

return

key_to_int = [ord(i) for i in key]
txt_to_int = [ord(i) for i in txt]
resultat = ""
for i in range(len(txt_to_int)):
    shift = key_to_int[i % len(key)]
    if typ == 'd':
        shift *= -1

    v = (txt_to_int[i] + shift) % 240

    resultat += chr(v)

return (resultat)
# Driver code
if __name__ == "__main__":
    texte = input("Veuillez choisir le fichier a encrypter: ")
    try :
        dirname = os.path.dirname( __file__ )
        filename = os.path.join(dirname, texte)
        with open(filename, 'r') as file:
            string = file.read().replace('\n', ' ').replace ("'", " ")
        except FileNotFoundError:
            print ("Erreur d'ouverture de fichier.")
            quit()

    key = input("Veuillez entrer la cle : ")
    q = vignere(string, key, 'e')
    try :
        dirname = os.path.dirname( __file__ )
        filename = os.path.join(dirname, "Cypher.txt")
        with open(filename, 'w') as file:
            string = file.write(q)
        except FileNotFoundError:
            print ("Erreur d'ouverture de fichier.")
            print("Votre message a ete encrypte avec succes! Veuillez verifier le fichier Cypher.txt")
  
```

### III- Implémenter en python une fonction permettant de décrypter un texte codé avec le chiffrement de Vigenère sans avoir la clé au préalable.

Pour décrypter le message sans la clé, on se base sur la probabilité d'occurrence de la lettre "e" en français. On commence par découper le texte encrypté en blocs, avec la fonction *TextSlicer*, et analyser

quel élément à la  $i$  ème position de chacun des blocs apparait le plus. "e" étant le caractère le plus fréquent, après "espace", on considère alors que le deuxième plus fréquent est probablement la lettre "e" et on calcule son écart par rapport à cette dernière pour déterminer la clé. Ceci se fait dans la fonction "*CesarKeyFinder*", car on peut considérer les éléments à la même position des différents blocs comme encryptés par la méthode de César puisqu'ils ont subi un déplacement par la même clé. Il ne nous reste donc qu'à itérer sur la longueur de clé choisie et vérifier à chaque fois si le texte décrypté, avec la fonction "*decryption*", est effectivement du français, avec la fonction *Validate*.

Il a lieu de mentionner que cette méthode ne fonctionne que s'il y a assez de "e" dans le texte considéré.

```

#Crack that works for VignereCypher.py
import collections
import gclid3
detector = gclid3.NNetLanguageIdentifier(min_num_bytes=0,
max_num_bytes=1000)
def CesarKeyFinder(cypher):
most_occurring_charater = collections.Counter(cypher).most_common()[1] #Find the most
occurring letter
shift = (ord(most_occurring_charater[0]) - ord('e')) %240 #Maps it to "e" and gets the shift
return chr(shift)

def Validate(plaintext):
result = detector.FindLanguage(text=plaintext)
if result.probability > 0.50 and result.language == 'fr':
print ("Le resultat du dechiffrement est fiable a {:.2f} % ".format(result.probability*100))
return True
else:
print("Le resultat du dechiffrement est peu fiable .")
return False

def TextSlicer(cypher,step):
C={}
for i in range (0,step):
C[i]=cypher[i::step]
return C

def decryption(txt,key):
key_to_int = [ord(i) for i in key]
txt_to_int = [ord(i) for i in txt]
resultat = ""
for i in range(len(txt_to_int)):
shift=key_to_int[i % len(key)]
v = (txt_to_int[i] - shift) % 255

```



```

resultat += chr(v)

return (resultat)
# Driver code
if __name__ == "__main__":
    with open('/home/kevin/Desktop/Security_Lab-master/Vigenere/Cypher.txt', 'r') as file:
        cypher = file.read()
        for size in range(1,10): #Size of the key
            Sliced = TextSlicer(cypher,size)
            key = ""
            for i in Sliced.keys():
                key +=CesarKeyFinder(Sliced[i])
            print ("Possible Key with size",size,"is :",key)
            resultat = decryption(cypher,key)
            if Validate(resultat):
                test=input(">Entrez T pour terminer ou une autre touche pour continuer. : ")
                if (test.upper() == 'T'):
                    print("Resultat de la decryption :")
                    print(resultat)
                    break

```

The screenshot shows the Visual Studio Code interface with the file 'Crack.py' open. The code in the editor matches the snippet provided above. The terminal window at the bottom shows the execution of the script using Python 3.8.5. The output displays possible keys for different sizes (1, 2, 3) and their corresponding decryption results. The user is prompted to enter 'T' to terminate or another key to continue.

```

Crack.py - Security_Lab-master - Visual Studio Code
36     return (resultat)
37
38 # Driver code
39 if __name__ == "__main__":
40     with open('/home/kevin/Desktop/Security_Lab-master/Vigenere/Cypher.txt', 'r') as file:
41         cypher = file.read()
42
43     for size in range(1,10): #Size of the key
44         Sliced = TextSlicer(cypher,size)
45         key = ""
46         for i in Sliced.keys():
47             key +=CesarKeyFinder(Sliced[i])
48         print ("Possible Key with size",size,"is :",key)
49         resultat = decryption(cypher,key)
50         if Validate(resultat):
51             test=input(">Entrez T pour terminer ou une autre touche pour continuer. : ")
52             if (test.upper() == 'T'):
53                 print("Resultat de la decryption :")
54                 print(resultat)
55
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master$ /usr/bin/python3 /home/kevin/Desktop/Security_Lab-master/Vigenere/Crack.py
Possible Key with size 1 is :
Le resultat du dechiffrement est peu fiable .
Possible Key with size 2 is : 2
Le resultat du dechiffrement est peu fiable .
Possible Key with size 3 is : we4
Le resultat du dechiffrement est fiable a 100.00 % .
>Entrez T pour terminer ou une autre touche pour continuer. :

```

#### IV- Implémenter un système peer to peer utilisant le chiffrement de Vigenère pour communiquer.

Server qui attend la connection du client pour déchiffrer le message reçu.

```
import socket
from VignereCypher_2 import vignere
HOST = '127.0.0.1' # Standard loopback interface address (localhost)
PORT = 6332 # Port to listen on (non-privileged ports are > 1023). port should be an integer
from 1-65535

# socket.socket() creates a socket object that supports the context manager type,
# so you can use it in a with statement. There's no need to call s.close():
#AF_INET,specify the address family and socket type (IPv4)
#SOCK_STREAM is the socket type for TCP
#listen()It specifies the number of unaccepted connections that the system will allow before
refusing new connections.
#accept() blocks and waits for an incoming connection.
data=""
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    print('Succes !!')
    print('The server ['+HOST+' ] starts & is listening on port '+str(PORT))
    conn, addr = s.accept()
    with conn: #on s'aassure de fermer la connection apres la reception de la requette
        print('Connected by', addr)
        while True:
            data = conn.recv(1024)
            print(vignere(data.decode(),'we4','d'))
            if not data:
                break
```

Activities Visual Studio Code Dec 4 12:08

server.py - Security\_Lab-master - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

OPEN EDITORS

SECURITY\_LAB-MASTER

.vscode

Ping

ping.png

Ping.py

PingServer.py

PingSweep

TraceRoute

Vigenere

\_\_pycache\_\_

version1

Betes.txt

client.py

coded.txt

Crack.py

Cypher.txt

kant.txt

server.py

Test.py

VignereCypher\_2.py

server.py

```

19 conn, addr = accept()
20 with conn: #on s'aassure de fermer la connection apres la reception de la requette
21     print('Connected by', addr)
22     while True:
23         data = conn.recv(1024)
24         print(vignere(data.decode(), 'we4', 'd'))
25         if not data:
26             break

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

2: Python

kevin@kevin-Inspiron-3583:~/Desktop/Security\_Lab-master\$ /usr/bin/python3 /home/kevin/Desktop/Security\_Lab-master/Vigenere/server.py

Succes !!

The server [127.0.0.1] starts & is listening on port 6332

Python 3.8.5 64-bit

Activities Visual Studio Code Dec 4 12:11

client.py - Security\_Lab-master - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

OPEN EDITORS

SECURITY\_LAB-MASTER

.vscode

Ping

ping.png

Ping.py

PingServer.py

PingSweep

TraceRoute

Vigenere

\_\_pycache\_\_

version1

Betes.txt

client.py

coded.txt

Crack.py

Cypher.txt

kant.txt

server.py

Test.py

VignereCypher\_2.py

client.py

```

6 IP = input("Enter the IP address: ")
7 t1 = datetime.now()
8 ttl = 1
9 PORT = 6332
10
11 def scan(addr):
12     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
13     socket.setdefaulttimeout(ttl)
14     result = s.connect_ex((addr, PORT))
15
16     if result == 0:
17         s.send(vignere('le texte en clair', 'we4', 'e').encode())
18         return 1
19     else:
20         return 0
21
22 def ping(addr):
23     if scan(addr):
24         t2 = datetime.now()
25         print('Ping: %s -> %s' % (addr, t2 - t1))

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

2: Python

kevin@kevin-Inspiron-3583:~/Desktop/Security\_Lab-master\$ /usr/bin/python3 /home/kevin/Desktop/Security\_Lab-master/Vigenere/server.py

Succes !!

The server [127.0.0.1] starts & is listening on port 6332

Connected by ('127.0.0.1', 53136)

le texte en clair

Veuillez entrer un text.

None

kevin@kevin-Inspiron-3583:~/Desktop/Security\_Lab-master\$

Python 3.8.5 64-bit

Client qui se connecte au server pour envoyer un message codé.

```
import socket
from datetime import datetime
from VignereCypher 2 import vignere
#fin des importations

IP = input("Enter the IP address: ")
t1 = datetime.now()
ttl = 1
PORT = 6332

def scan(addr):
    s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    socket.setdefaulttimeout(ttl)
    result = s.connect_ex((addr,PORT))
    if result == 0:
        s.send(vignere('le texte en clair','we4','e').encode())
        return 1
    else :
        return 0

def ping(addr):
    if (scan(addr)):
        t2 = datetime.now()
        total = t2 - t1
        print('Succes !!')
        print ("from {}: ttl={} time={} ".format(addr,ttl,total))
    else:
        print('Port unreachable')

ping(IP)
```

client.py - Security\_Lab-master - Visual Studio Code

```

6 IP = input("Enter the IP address: ")
7 t1 = datetime.now()
8 ttl = 1
9 PORT = 6332
10
11 def scan(addr):
12     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
13     socket.setdefaulttimeout(ttl)
14     result = s.connect_ex((addr, PORT))
15
16     if result == 0:
17         s.send(vignere('le texte en clair', 'we4', 'e').encode())
18         return 1
19     else:
20         return 0
21
22 def ping(addr):
23     if scan(addr):
24         t2 = datetime.now()

```

Terminal:

```

kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master$ cd Vignere
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master/Vignere$ python3 ./client.py
Enter the IP address: 127.0.0.1
Succes !!
from 127.0.0.1: ttl=1 time=0:00:00.000803
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master/Vignere$

```

client.py - Security\_Lab-master - Visual Studio Code

```

6 IP = input("Enter the IP address: ")
7 t1 = datetime.now()
8 ttl = 1
9 PORT = 6332
10
11 def scan(addr):
12     s = socket.socket([socket.AF_INET, socket.SOCK_STREAM])
13     socket.setdefaulttimeout(ttl)
14     result = s.connect_ex((addr, PORT))
15
16     if result == 0:
17         s.send(vignere('le texte en clair', 'we4', 'e').encode())
18         return 1
19     else:
20         return 0
21
22 def ping(addr):
23     if scan(addr):
24         t2 = datetime.now()

```

Terminal:

```

Connected by ('127.0.0.1', 53136)
le texte en clair
Veuillez entrer un text.
None
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master$ /usr/bin/python3 /home/kevin/Desktop/Security_Lab-master/Vignere/client.py
Enter the IP address: "C:\Program Files\Python\python3\python3.exe"
File "/home/kevin/Desktop/Security_Lab-master/Vignere/client.py", line 6, in <module>
IP = input("Enter the IP address: ")
KeyboardInterrupt

kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master$ /usr/bin/python3 /home/kevin/Desktop/Security_Lab-master/Vignere/server.py
Succes !!
The server [127.0.0.1] starts & is listening on port 6332
Connected by ('127.0.0.1', 53166)
*8.G.80.B$=
Veuillez entrer un text.
None
kevin@kevin-Inspiron-3583:~/Desktop/Security_Lab-master$

```