CHALLENGE KRYPTON

Krypton1

Décodage de la clé en Base64 : **KRYPTONISGREAT**

```
flaya@flaya:~$ echo S1JZUFRPTklTR1JFQVQ= | base64 --decode
KRYPTONISGREATflaya@flaya:~$
```

Krypton2

Affichage de la clé pour Krypton2 :

```
krypton1@krypton:/home$ cd /krypton/krypton1
krypton1@krypton:/krypton/krypton1$ ls
README  krypton2
krypton1@krypton:/krypton/krypton1$ ls -lh
total 8.0K
-rw-r----- 1 krypton1 krypton1 882 Jun 15 11:40 README
-rw-r----- 1 krypton1 krypton1  26 Jun 15 11:40 krypton2
krypton1@krypton:/krypton/krypton1$
```

```
krypton1@krypton:/krypton/krypton1$ cat krypton2
YRIRY GJB CNFFJBEQ EBGGRA
krypton1@krypton:/krypton/krypton1$
```

Décodage la clé (ROT13) : **ROTTEN**

```
krypton1@krypton:/krypton/krypton1$ echo "YRIRY GJB CNFFJBEQ EB
-z]' '[N-ZA-Mn-za-m]'
LEVEL TWO PASSWORD ROTTEN
```

Krypton3

Affichage de la clé pour Krypton3 :

```
krypton2@krypton:~$ cd /krypton/krypton2/
krypton2@krypton:/krypton/krypton2$ cat krypton3
OMQEMDUEQMEK
```

Création d'un script python (en local) :

```
root@flaya:~# touch decrypt.py
root@flaya:~# ls
Access-Your-Private-Data.desktop  etat_ports.sh             openssl-
decrypt.py                        halberd-master            README.t
epsi_fr_1485869054.63.csv         ip_sous_domaine_epsi.txt  theHarve
epsi_fr_1485870891.7.csv          master.zip.1              toto.txt
epsi_subdomains.txt               openssl
root@flaya:~# nano decrypt.py
```

```
  GNU nano 2.5.3

#!/usr/bin/python
import sys

ctext = "OMQEMDUEQMEK"
alphabet = list("ABCDEFGHIJKLMNOPQRSTUVWXYZ")
plaintext = ""
shift = 1

while shift <= 26:
 for c in ctext:
  if c in alphabet:
   plaintext += alphabet[(alphabet.index(c)+shift)%(len
 print("Shift used: " + str(shift))
 print("Ciphertext: " + ctext)
 print("Plaintext: " + plaintext)
 shift = shift + 1
 plaintext = ""
```

Récupération du résultat le plus probable : **CAESARISEASY**

```
root@flaya:~# python decrypt.py
Shift used:  1
Ciphertext:  OMQEMDUEQMEK
Plaintext:  PNRFNEVFRNFL
Shift used:  2
Ciphertext:  OMQEMDUEQMEK
Plaintext:  QOSGOFWGSOGM
Shift used:  3
Ciphertext:  OMQEMDUEQMEK
Plaintext:  RPTHPGXHTPHN
Shift used:  4
Ciphertext:  OMQEMDUEQMEK
Plaintext:  SQUIQHYIUQIO
Shift used:  5
Ciphertext:  OMQEMDUEQMEK
Plaintext:  TRVJRIZJVRJP
Shift used:  6
Ciphertext:  OMQEMDUEQMEK
Plaintext:  USWKSJAKWSKQ
Shift used:  7
Ciphertext:  OMQEMDUEQMEK
Plaintext:  VTXLTKBLXTLR
Shift used:  8
Ciphertext:  OMQEMDUEQMEK
Plaintext:  WUYMULCMYUMS
Shift used:  9
Ciphertext:  OMQEMDUEQMEK
Plaintext:  XVZNVMDNZVNT
```

```
Ciphertext:  OMQEMDUEQMEK
Plaintext:  VTXLTKBLXTLR
Shift used:  8
Ciphertext:  OMQEMDUEQMEK
Plaintext:  WUYMULCMYUMS
Shift used:  9
Ciphertext:  OMQEMDUEQMEK
Plaintext:  XVZNVMDNZVNT
Shift used:  10
Ciphertext:  OMQEMDUEQMEK
Plaintext:  YWAOWNEOAWOU
Shift used:  11
Ciphertext:  OMQEMDUEQMEK
Plaintext:  ZXBPXOFPBXPV
Shift used:  12
Ciphertext:  OMQEMDUEQMEK
Plaintext:  AYCQYPGQCYQW
Shift used:  13
Ciphertext:  OMQEMDUEQMEK
Plaintext:  BZDRZQHRDZRX
Shift used:  14
Ciphertext:  OMQEMDUEQMEK
Plaintext:  CAESARISEASY
Shift used:  15
Ciphertext:  OMQEMDUEQMEK
Plaintext:  DBFTBSJTFBTZ
Shift used:  16
Ciphertext:  OMQEMDUEQMEK
Plaintext:  ECGUCTKUGCUA
Shift used:  17
Ciphertext:  OMQEMDUEQMEK
Plaintext:  FDHVDULVHDVB
```

Krypton 4

Affichage de la clé pour Krypton4 :

```
krypton3@krypton:~$ cd /krypton/krypton3/
krypton3@krypton:/krypton/krypton3$ cat krypton4
KSVVW BGSJD SVSIS VXBMN YQUUK BNWCU ANMJS krypton3@krypton:/krypton/krypton3$
```

Analyse de fréquence :

Fréquence des mots anglais

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
8.2	1.5	2.8	4.3	12.7	2.2	2.0	6.1	7.0	0.2
K	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**
0.8	4.0	2.4	6.7	7.5	1.9	0.1	6.0	6.3	9.1
U	**V**	**W**	**X**	**Y**	**Z**				
2.8	1.0	2.4	0.2	2.0	0.1				

On peut les classer par fréquence

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

En utilisant un site d'analyse de fréquence on peut analyser la fréquence des lettres dans les fichiers *found1*, *found2*, et *found3*.

Classement sur *found1*

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

Classement sur *found2*

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

Classement sur *found3*

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

Correspondance

La phrase à décrypter est : **KSVVW BGSJD SVSIS VXBMN YQUUK BNWCU ANMJS**

On cherche les correspondances pour les lettres de la phrase uniquement (classées par fréquence) :

Lettre	Possibilités
S	E
V	D L U
N	O I N S

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

Par substitution, on trouve que les meilleures possibilités pour le premier de mot de 5 lettres sont :
WELLD ou WELLU.

On remplace les lettres :
WELLWBG**E**JD**ELE**I**EL**XBMNYQUUKBNWCUANMJ**E**

En analyse de fréquence de mot en anglais, il est très probable que la combinaison JDS corresponde à THE.
On remplace :
WELLWBG**ETHELE**I**EL**XBMNYQUUKBNWCUANM**TE**
WELLDONETHELEVELX**O**MNYQUUK**O**N**D**CUANM**TE**
WELLDONETHELEVELFOURPASSWORDISBRUTE => Le mot de passe est BRUTE

Krypton 5

Affichage de la clé pour Krypton5 :

```
krypton4@krypton:~$ cd /krypton/krypton4
krypton4@krypton:/krypton/krypton4$ ls
HINT  README  found1  found2  krypton5
krypton4@krypton:/krypton/krypton4$ cat krypton5
HCIKV RJOXkrypton4@krypton:/krypton/krypton4$
```

Nous allons utiliser la technique suivante pour décrypter le Vigenère :

Tout d'abord, il est nécessaire de diviser le texte en 6 sous-texte correspondant à chaque caractère étant crypté par la même lettre :

YYICSJ**I**ZIBAG**Y**YXRIE**W**VIXAF**N**JOOVQ**Q**VHDLC**R**KLBSS**L**YXRIQ**Y**IIOXQ**T**WXRIC**R**VVKPB**H**ZXIYL**Y**ZPDLC**D**IIKG…

Par exemple, la première séquence est :
YIYWNQRLYTRHYDJTWZSLNNHTMJJYFNYIJJSLWNMFXBBKXIMJTBMIYJJNTYBWKWWLFGWISJSZYSYPJNFJQFWTYWKJJMMNSYWKYSAYMTSQZJRFDMKXFJJPKFSTTTJMBMJDSQIJPFJTSJWJPJIKXISJFFYXMQMYIMZYFSJWJNTWGJYGZTMTYSFFJTWJQBSFSJSJIJJNKWSXZYZKXMSSIFTXSSKTJTYWMYKLTISNJFITWIXNBSJHJF

Nous allons utiliser la technique de l'analyse de fréquence pour définir une équivalence entre deux lettres. On obtient la liste de fréquence suivante (uniquement les 8 plus fréquents) :

J => 37
S => 24
Y => 22
T => 20
F => 18
W => 17
M => 16
I => 14

On peut donc en déduire que la lettre la plus fréquente correspond au **E**. Par conséquent le décalage entre **J** et **E** est de 6. La première lettre de la clé est donc **F**.

On répète l'opération pour chacun des 5 sous-textes restants.

On obtient les décalages suivants :

6 17 5 11 5 25 soit la clé **FREKEY**

On utilise le script Python suivant pour décrypter le mot de passe :

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# On met notre message crypté dans
# la variable message et on décode.
message = "HCIKV RJOX"
message = message.decode("utf-8")
# On demande la clé.
key = raw_input("\nQuelle est la clé?\n").decode("utf-8").upper()

lettres = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

keyIndex = 0

decrypted = ""

for car in message:

    num = lettres.find(car)

    if num != -1:
        # Au lieu d'additionner, on soustraie
        # le nombre qui correspond à la lettre
        # de la clé.
        num -= lettres.find(key[keyIndex])
        num %= len(lettres)
        decrypted += lettres[num]
        keyIndex += 1
        if keyIndex == len(key):
            keyIndex = 0

    else:
        decrypted += car

print "\n*** Message décrypté! ***"
print decrypted + "\n"
```

Décryptage du mot de passe avec la clé : **CLEAR TEXT**

```
D:\DOC\Cours_EPSI\Cryptographie\challenge\vigenere>python vigener.py

Quelle est la cle?
FREKEY

*** Message decrypte! ***
CLEAR TEXT
```