Numérique et Sciences Informatiques
Chapitre XI - Sécurisation des communications
Travaux Dirigés 20

Classe de terminale

On rappelle l'existence des fonctions suivantes, natives en Python :

- `ord(carac)` : renvoie le nombre entier représentant le code Unicode (UTF-8) du caractère `carac` mis en paramètre.
  Exemple :

```
>>> ord(';')
59
```

- `chr(nb)` : renvoie la chaîne représentant un caractère dont le code de caractère Unicode est le nombre entier `nb`.

```
>>> chr(60)
'<'
```

Ainsi que de l'opérateur `%` :

- `a % b` : renvoie le reste de la division euclidienne de `a` par `b`.
  Exemple :

```
>>> 17 % 3
2
```

Dans tout le TD, n'oubliez pas d'écrire la spécification (docstrings) et la mise au point (jeu de tests) de vos fonctions.

Chiffrement symétrique

On rappelle que dans le cas d'un chiffrement symétrique, la clé partagée permet de chiffrer un message en clair et de déchiffrer un message chiffré.

Pour cette première partie, nous prendrons le message en clair suivant :
'BONJOUR A TOUS. VIVE LA MATIERE NSI !'

I. Chiffrement par décalage

Un des chiffrement les plus simples est le chiffrement par décalage. Il existe plusieurs types de chiffrement par décalage. Nous allons en voir deux : par le chiffre de César et par le chiffre de Vigenère.

Dans cette section, on ne s'intéresse qu'à un ensemble de caractères composé uniquement de lettres majuscules, que l'on appellera alphabet :
$ABCDEFGHIJKLMNOPQRSTUVWXYZ$

Le chiffre de César

Utilisé par Jules César dans ses correspondances secrètes, le chiffrement de César remplace chaque lettre du message clair par une lettre à distance fixe vers la droite, en revenant éventuellement au début de l'alphabet si besoin. Pour déchiffrer, le principe est le même mais en se décalant vers la gauche, en revenant éventuellement à la fin de l'alphabet si besoin.
Cette distance fixe est la clé partagée, appelée dans ce cas *chiffre de César*.

Exemple :

- lorsqu'on chiffre avec un chiffre de César égal à 3, le A devient D, le Y devient B.
- lorsqu'on déchiffre avec un chiffre de César égal à 5, le F devient A, le C devient X.

1. Tester sur le message en clair le chiffrement de César avec un nombre de César égal à 6.
2. (a) Quel est le nombre entier représentant le code Unicode du caractère `'A'` ? du caractère `'Z'` ?
   (b) Si `carac` est un caractère de l'alphabet, que peut renvoyer l'instruction suivante ?

```
ord(carac) - 65
```

   (c) Que peut renvoyer `a % 26` où `a` est un entier ?
   (d) Implémenter le chiffrement de César dans une fonction `chiffrement_cesar(message_clair, clef)` qui prend en paramètres une chaîne de caractères `message_clair` et un entier `clef`. Cette fonction renvoie une chaîne de caractères correspondant au message chiffré.

3. Implémenter une fonction `dechiffrement_cesar(message_chiffre, clef)` qui prend en paramètres une chaîne de caractères `message_chiffre` et un entier `clef`. Cette fonction renvoie une chaîne de caractères correspondant au message en clair originel.

Le chiffre de Vigenère

Le chiffrement de Vigenère tient son nom de Blaise de Vigenère qui l'a décrit dans son *Traité des chiffres* paru en 1586.

La clef partagée est une chaîne de caractères (et pas un nombre comme dans le chiffrement de César) composée des caractères de l'alphabet.

Chaque caractère de la clef partagée est placé sous un caractère du message en clair appartenant à l'alphabet, dans l'ordre d'écriture, en répétant la clef autant de fois que nécessaire.

Choisissons la clef partagée suivante : $INFORMATIQUE$

```
BONJOUR A TOUS. VIVE LA MATIERE NSI !
INFORMA T IQUE  INFO RM ATIQUEI NFO
```

On applique le chiffrement de Vigenère selon la table de Vigenère ci-dessous.

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

La lettre chiffrée est au croisement de la lettre en clair et de la lettre de la clef.
On peut remarquer qu'il s'agit d'un chiffrement par décalage de distance variable selon la lettre de la clef.

1. Tester ce chiffrement de Vigenère sur le message avec la clef $INFORMATIQUE$.
2. A quel décalage correspond la lettre `I` de la clef partagée ? Et la lettre `A` ? Et la lettre `Z` ?
3. Implémenter la fonction `chiffrement_vigenere(message_clair, clef)` qui prend en paramètres deux chaînes de caractères `message_clair` et `clef`. Cette fonction renvoie une chaîne de caractères correspondant au message chiffré.
4. Implémenter la fonction `dechiffrement_vigenere(message_chiffre, clef)` qui prend en paramètres deux chaînes de caractères `message_chiffre` et `clef`. Cette fonction renvoie une chaîne de caractères correspondant au message en clair originel.

II. Un autre chiffrement symétrique : le chiffrement avec XOR bit à bit

Dans cette section, tous les caractères Unicode seront utilisés (pas seulement les caractères de l'alphabet en majuscule comme dans les deux exemples précédents).

Le XOR est un opérateur binaire dont le symbole est $\oplus$ mais implémenté en Python par `^`.
On a pour toute variable binaire x et y :

$$x \oplus y = (x \wedge \neg y) \vee (\neg x \wedge y)$$

où $\neg$ est l'opérateur unaire de la négation logique, $\wedge$ est l'opérateur binaire `ET` et $\vee$ l'opérateur binaire `OU`.

1. Rappeler les tables de vérité de l'opérateur unaire `¬` et des opérateurs binaires `∧` et `∨`.
2. Compléter alors la table de vérité de l'opérateur binaire `⊕` ci-dessous.

x	y	¬x	¬y	x ET ¬y	¬x ET y	x ⊕ y
0	0					
0	1					
1	0					
1	1					

3. L'algorithme de chiffrement avec XOR bit à bit est le suivant :
   - Écrire le message en clair et la clef dans le code Unicode en binaire.
   - On place les bits de la clef sous ceux du message en clair, en répétant autant de fois que nécessaire les bits de la clef.
   - On effectue l'opération binaire XOR bit à bit.
   - Chaque octet obtenu correspond à un caractère Unicode.

   Tester cet algorithme sur les trois premiers caractères du message en clair avec la clef $INFORMATIQUE$

4. (a) Expliquer le résultat de `66 ^ 73` dans la console. Est-il utile, en Python, de convertir dans le code Unicode en binaire pour implémenter ce chiffrement ?
   (b) Quel caractère est renvoyé par l'instruction `chr(66 ^ 73)` ?
   (c) Implémenter une fonction `chiffrement_xor(message_clair, clef)` qui prend en paramètres deux chaînes de caractères `message_clair` et `clef`. Cette fonction renvoie une chaîne de caractères correspondant au message chiffré.

5. (a) Dresser la table de vérité de l'expression booléenne suivante : $(x \oplus y) \oplus y$.
   (b) Que peut-on en déduire pour le déchiffrement ?
   (c) Implémenter une fonction `dechiffrement_xor(message_chiffre, clef)` qui prend en paramètres deux chaînes de caractères `message_chiffre` et `clef`. Cette fonction renvoie une chaîne de caractères correspondant au message en clair originel.