Cryptographie et Sécurité
Série 2 : TP Python - Chiffrement de Vigenère

29 Septembre 2022

A rendre sur **Moodle**, sous forme de fichier **.py** (implémenté avec **Python 3**), avant le **Mercredi 5 Octobre 2022 à 23h59**.

Votre code doit être **suffisamment commenté**.

Introduction

Dans la série précédente, on a forcé, par une analyse à la main, le chiffrement de César et la substitution monoalphabétique. Cette fois, avec l'aide du langage de programmation Python, on va s'attaquer au chiffrement de Vigenère.

Le chiffrement de Vigenère est inconditionnellement sûr si on utilise une clé de la même longueur que le plaintext et qu'elle n'est pas réutilisée (c'est le principe du one-time pad). Seulement, ce n'est en général pas le cas, et la clé est souvent répétée un grand nombre de fois. C'est ce qui est alors exploité pour casser le chiffrement de Vigenère.

Rappel : Chiffrement de Vigenère

Le chiffrement de Vigenère est ce qu'on appelle une substitution polyalphabétique (c'est à dire un système de chiffrement ou un même caractère situé à plusieurs endroits du message peut être encrypté par des caractères différents à chaque fois).
On donne une valeur à chaque caractère : le A vaut 0, le B vaut 1, et ainsi de suite jusqu'au Z qui vaut 25. La substitution est faite en sommant la valeur du caractère du message à la valeur du caractère à la même position de la clé, modulo 26. On obtient alors la valeur du nouveau caractère (et le modulo assure que l'on reste dans le même espace de caractères). La clé est une chaine de caractère, que l'on répète autant de fois que nécessaire pour atteindre la même longueur que le message.

Exemple : Le message "BONJOUR" encodé avec la clé "ABRA" donne le ciphertext "BPEJOVI" ($B + A = B$, $O + B = P$, $N + R = E$, $J + A = J$, puis on recommence au début de la clé, $O + A = O$, et ainsi de suite).

Cet algorithme s'étend facilement à un alphabet plus grand (et même à trois alphabets différents pour le plaintext, la clé, et le ciphertexts, tant qu'on associe une numérotation claire aux trois alphabets). Dans le cas de l'alphabet simple de 26 caractères (lettres de "A" à "Z"), la figure 1 montre sous forme d'une table le résultat de ce chiffrement.

	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

Figure 1: Table de Vigenère

Principe pour forcer le chiffrement de Vigenère

Pour forcer ce chiffrement, on applique un processus en trois étapes :

- Trouver la longueur de la clé (via une méthode de superposition, décrite en détail ci-après),
- Séparer le texte en sous-textes (ce qui nécessite la longueur de la clé), et appliquer une analyse fréquentielle sur chaque sous-texte (méthode décrite en détail ci-après) pour retrouver la clé,

- Et finalement déchiffrer le ciphertext avec la clé.

Etape 1 : Déterminer la longueur de la clé

Pour déterminer la longueur de la clé, on va utiliser ici une méthode de superposition. Le but est de superposer deux copies du texte, décalées d'un certains nombre de pas (la longueur de clé qu'on veut tester), et d'estimer ce qu'on appelle l'indice de coincidence (il en existe plusieurs variantes, on ne verra ici qu'une version simplifiée). On va donc obtenir un indice de coincidence différent pour chaque longueur de clé possible. La longueur pour laquelle l'indice est maximal est très probablement la bonne longueur de clé.
Soit N le nombre de caractères dans le texte, L la longueur testée, A et B les deux textes (A l'original, B celui décalé de L caractères) et a_i et b_i le caractère à la i-ème position dans chacun des deux textes (b_i est donc le i-ème caractère de B, c'est à dire le (i+L)-eme caractère dans le texte original A). On calcule l'indice comme :

$$\frac{\sum_{j=1}^{N-L}[a_j == b_j]}{N-L}$$

Avec $[a_j == b_j]$ qui vaut 1 si les caractères sont égaux, et 0 sinon. Donc on compte simplement combien de caractères sont identiques entre les deux textes, rapportés au nombre total de caractères testés. Cela nous donne la proportion de redondances.

La répartition des lettres dans la langue anglaise fait que certaines lettres sont plus fréquentes que d'autres. Cela augmente donc la probabilité d'obtenir des redondances si les textes sont encodés par le même caractère. Cette méthode permet donc de déterminer quelle longueur maximise ces redondances, et c'est donc très probablement la longueur de la clé.

Ici, pour simplifier, on suppose qu'on sait déjà que la clé ne fait pas plus de 10 caractères. Il suffit donc de calculer cette valeur pour les longueurs de clés jusqu'à 10, puis de choisir celle qui maximise cet indice.

Etape 2 : Vigenère n'est qu'une accumulation de Césars

Maintenant qu'on a la longueur de la clé, qu'on note k, on peut résoudre Vigenère simplement.

Séparons en k groupes les caractères encryptés, pour créer k sous-textes.
Dans le premier sous-texte, mettons les caractères aux positions 1, 1+k, 1+2k,

etc.
Tous ces caractères sont encryptés par le même caractère de la clé. C'est donc un simple chiffrement de César !
Et même si le contenu n'a aucun sens sémantiquement, on peut utiliser une analyse statistique pour déterminer quel est le décalage, et donc quel est le caractère de la clé correspondant (encore une fois, grâce aux différentes fréquences de caractères dans la langue).

Pour cela, on compare les fréquences des caractères entre ce sous-texte particulier et les fréquences de la langue courante. On teste pour chacun des 26 décalages possibles quelle est la distance (au sens mathématique) entre les fréquences de notre sous-texte et les fréquences du langage. On utilisera ici la distance la plus classique, c'est à dire la distance euclidienne :

Si F_i sont les fréquences connues du langage (i de 0 à 25, 0 pour A, 25 pour Z), et M_i les fréquences mesurées pour le sous-texte, alors on a pour un décalage Dec donné :

$$Dist_{Dec} = \sqrt{\sum_{i=0}^{25}(F_i - M_{(i+Dec)mod26})^2}$$

Il suffit de calculer cette valeur pour chaque décalage, et le décalage avec la distance la plus petite est très probablement le bon décalage. Ce décalage correspond donc au premier caractère de la clé.

Il suffit alors de répéter cela pour chaque sous-texte. Dans le second sous-texte, on place les caractères 2, 2+k, 2+2k, etc. Et ainsi de suite pour les autres sous-textes. On trouve alors le décalage pour chaque sous-texte, et donc chaque caractère de la clé.

Etape 3 : Déchiffrer le message

Rien de bien compliqué ici : on a la clé, on a le ciphertext, il suffit de répéter la clé et déchiffrer le message.

TP : Casser Vigenère

Ici, le langage du texte à décoder est l'anglais.

Vous trouverez avec le TP un fichier Python sur Moodle ("ciphertextTP2.py"), contenant les fréquences moyennes des lettres de la langue anglaise, la taille maximale possible pour la clé, ainsi qu'un texte encrypté par le chiffrement de

Vigenère (uniquement constitué des 26 lettres minuscules de l'alphabet, sans aucun ponctuation, ni espaces ni autres caractères).

Votre but est de créer un code permettant d'appliquer les processus décrits plus haut, et de l'utiliser pour retrouver la clé qui à chiffré ce texte et le déchiffrer.

Dans le détail, vous devez implémenter un code :

- Qui **implémente la méthode de l'indice de coincidence**, pour déterminer la longueur de la clé.
- Qui **implémente l'analyse fréquentielle** pour déterminer quel est le décalage de chaque groupe de caractères (et donc quel est le caractère de la clé correspondant à chaque fois) dans le but de trouver la clé. Pour cela, séparez votre texte en k groupes (k est la longueur de clé trouvée précédemment), et faites une analyse fréquentielle sur chaque groupe pour déterminer un à un les caractères de la clé.
- Qui **utilise la clé pour déchiffrer le message**.
- Votre code prendra le texte fourni, et retournera la longueur de la clé, la clé elle-même, et le texte déchiffré.

N'oubliez pas de **tester votre code régulièrement**, afin de détecter plus facilement et rapidement ou sont vos erreurs.