TP6 : Cryptographie symétrique

1 Mise en bouche

Créer une classe `Cesar` contenant en attribut la clé : le constructeur initialise la clé avec son argument, et avec une valeur aléatoire si aucun argument ni fournis. Implémenter deux méthodes permettant de chiffrer et déchiffrer un message. **Important : dans ce TP, on considère des messages sur 27 caractères (A-Z + espace), où les espaces ne sont pas modifiés lors du chiffrement. Pourquoi n'est-ce pas (trop) restrictif ?**

Écrire ensuite trois fonctions permettant de déterminer la clé utilisée par un chiffrement de César. Chacune de ces fonctions pourra utiliser les différents niveaux d'attaque suivants :

— Texte chiffré choisi : elle prend en argument un objet `Cesar` et peut utiliser la méthode de déchiffrement.

— Texte clair choisi : elle prend en argument un objet `Cesar` et peut seulement utiliser la méthode de chiffrement.

— Texte clair connu : elle prend en argument deux chaînes (correspondant au message en clair et chiffré).

Avant de s'attaquer au dernier niveau (texte chiffré seul), il faut développer quelques outils.

2 Boîte à outils

Implémenter des fonctions permettant de :

— calculer les fréquences de chaque lettre dans un message

— calculer la distance (somme des valeurs absolues des différences) entre un tableau de fréquences et celui du français (en %)

A	8.15	N	7.12
B	0.97	O	5.28
C	3.15	P	2.80
D	3.73	Q	1.21
E	17.39	R	6.64
F	1.12	S	8.14
G	0.97	T	7.22
H	0.85	U	6.38
I	7.31	V	1.64
J	0.45	W	0.03
K	0.02	X	0.41
L	5.69	Y	0.28
M	2.87	Z	0.15

— calculer l'indice de coïncidence

Utiliser ces fonctions pour déterminer la clé la plus probable dans une attaque de type texte chiffré seul.

Pouvez-vous décoder le message suivant ?

VK MSQKVO KIKXD MRKXDO DYED VODO CO DBYEFK PYBD NOZYEBFEO AEKXN VK LSCO PED FOXEO

3 Substitution

Adapter la classe `Cesar` pour chiffrer et déchiffrer par substitution. Implémenter des méthodes d'attaque dans les cadres de texte chiffré choisi, texte clair choisi et texte clair connu.

Implémenter ensuite une attaque à texte chiffré seul par analyse de fréquences. Cette fonction sera raffinée en section 5. Essayez de décrypter quelques messages (assez longs). Que constatez-vous ?

4 Vigenere

À nouveau, adapter la classe `Cesar` pour chiffrer et déchiffrer un code de Vigenère. Implémenter des méthodes d'attaque dans les cadres de texte chiffré choisi, texte clair choisi et texte clair connu. Pour le décryptage en texte chiffré seul, on s'intéresse aux deux attaques du cours.

4.1 Friedman

Si la longueur de la clé est k, alors pour tout i, le sous-message formé par les lettres aux positions i modulo k est chiffré avec un chiffrement de César. En particulier, son indice de coïncidence doit être proche de 0.07 (et non de 0.03).

En utilisant ce critère, écrire une fonction qui détermine la longueur (la plus probable) de la clé. Écrire ensuite une fonction qui calcule la clé (la plus probable).

4.2 Babbage-Kasiski

On cherche à nouveau à déterminer la longueur de la clé.

Implémenter une fonction qui détermine la liste des (positions des) sous-mots d'au moins p lettres apparaîssant plusieurs fois dans un message.

La plupart de ces répétitions correspondent (sauf faux-positif) à des sous-mots du message en clair qui ont été chiffrés avec les mêmes portions de la clé. Dans ce cas, l'écart entre leurs positions est un multiple de la longueur de la clé. En déduire une seconde implémentation de la méthode qui fournit la longueur (probable) de la clé.