Méthode de décryptage de Vigenère

BODIAN Ibrahima

October 2019

1 Introduction

Au Moyen-âge, Vigenère a imaginé une extension poly-alphabétique du Code de César, c'est à dire un chiffrement par addition en parallèlle. Le décalage d'une lettre varie en fonction du caractere de la clé qui se trouve "en phase" avec cette dernière. Il n'y a bien sûr pas de mystère à dire que le code de César n'apporte aucune sécurité car ,premierement,l'espace des clés n'est pas suffissament grand pour garantir une sûreté fiable, de plus, elle est vulnéable aux attaques chiffré seul, c'est à dire,on peut deviner le message initial grâce à une analyse fréquentielle. Contrairement à César, le chiffrement de Vigenère résiste à cette attaque car une même lettre du message peut suivant sa position dans celui-ci être remplacée par des lettres différentes. Cependant,il peut être cassé en deux étapes :

1] Déterminer la taille 'n' de la clé grâce à l'indice de coïncidence

2] Résoudre les 'n' chiffres de César sur les sous suites k modulo n grâce aux indices de coîncidence mutuels

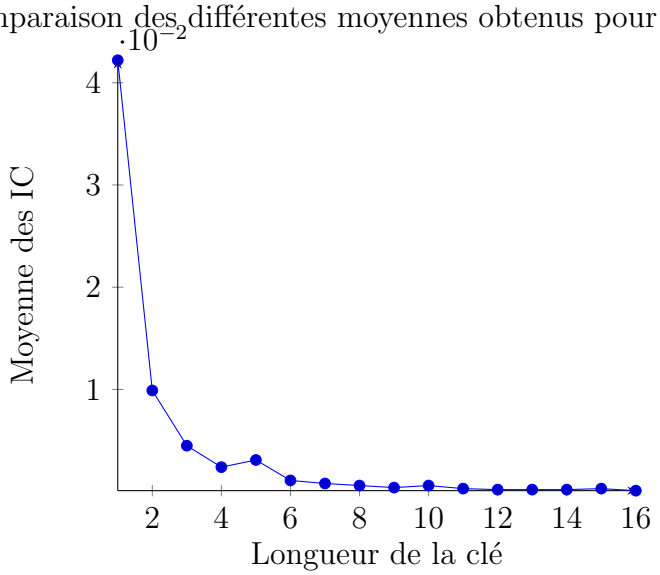
2 Recherche de la longueur de la clé

Soit un message initial de longueur l. On suppose que le texte chiffré provient d'un texte clair écrit dans une langue naturelle dont on connaît les propriétés. Comme dit précèdamant, la première étape consiste à deviner la taille probable de la clé qui varie entre 2 et 'l'. Pour cela, on s'aide de l'indice de coïncidence. Elle s'applique sur un texte et calcule la probabilité que deux caractères aléatoires du texte soient égaux. Si pour i allant de 0 à 25, f_i représente les fréquences des 26 lettres dans le message,nous avons la formule suivante:

$$IC(x) = \frac{\sum_{i=0,}^{25} f_i(f_i - 1)}{l(l-1)}$$

Le calccul de l'indice de coïncidence d'un texte en français vaut environ 0,074 et sachant que la fréquence des distributions est invariante lors d'un décalage d'un texte, l'IC d'un texte chiffré et le même que celui du clair pour un chiffrement mono-alphabétique. Nous allons donc proceder de la manière suivante : Pour une longueur de clé déterminée, nous allons calculer la moyenne des indices de coïncidence des sous-suites. Exemple, imaginons que nous avons un texte s=aaxn sdzp djlj zmdk et que nous avons fixé la longueur de la clé à 4, alors nous ferons la moyenne des indices de coïncidence des textes suivantes : s_1=asdz, s_2=adjm, s_3=xzld et s_4=npjk. Nous effectuons cette opérations pour toutes les valeurs possibles de longueur de clé et puis nous comparons les différentes moyenne obtenues.

Comparaison des différentes moyennes obtenus pour vigCryp13



Sur les 145 valeurs possible de longueur de clé de vigCrypt13, on a affiché 16 valeurs sur le graphique (les 16 premières valeurs) car les autres étant non significative (IC inf à 0.0001). Etant dans le cas d'un chiffrement de César pour une longueur de clé de taille 1 , concentrons nous a partir de l=2. A première vue, la moyenne des indices de coïncidence pour une longueur de clé valant 2 est largement supérieur aux autres valeurs, mais il faut également prendre en compte le faite que plus la taille de la clé est grande, plus on

applique la moyenne des IC sur des textes de longueur moindre et plus il est difficile de differencier la fréquence de chacune des lettres. Ceci fausse grandement l'IC et donc au lieu de regarder la plus grande valeur, on va plutôt observer la valeur de la clé qui réévalue l'IC par rapport à "ses voisins", c'est à dire celle qui présente un 'pic' bien distingué sur le graphique. Dans l'exemple ci-dessus, la clé a probablement une longueur de taille 5.

Rmq: on observe ce phénomène de 'pic' pour des valeurs de taille 10 et 15 (moins marqué) car ce sont des multiples de la taille de la clé.

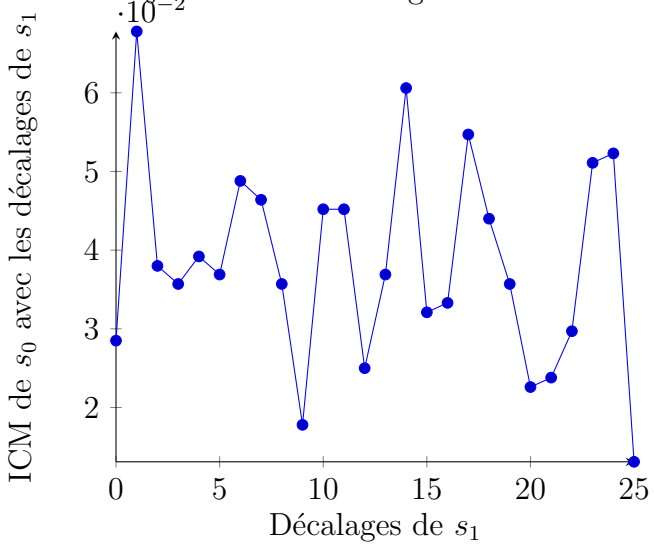
3 Obtention de la clé

Maintenant qu'on a la taille de la clé, on considère les sous-chaînes sur lesquelles on a calculé l'indice de coïncidence. Notons k_i le décalage de la sous-chaîne s_i par rapport au texte clair : cette valeur correspond à la i-ième lettre de la clé. Pour tout couple (i,j), la sous-chaîne s_j aura la même distribution de caractères que la chaîne s_i avec un décalage k_j- k_i. On calcule donc l'indice de coïncidence mutuelle entre s_0 et chacun des s_i ainsi que leurs décalages grâce à la formule suivante.

$$ICM(x,y) = \frac{\sum_{i=0,}^{25} f_i g_i}{lm}$$

avec f_i les fréquences des 26 caractères dans le texte x de longueur l et g_i les fréquences des 26 caractères dans le texte y de longueur m. L'indice de coïncidence mutuelle calcule la probabilité qu'un caractère aléatoire d'une chaîne x soit égal à un caractère aléatoire d'une chaîne y.

ICM de la sous chaîne s_0 avec les 26 décalages de la sous chaîne s_1 de vigCrypt13



Ici, la valeur k_1-k_0 maximisent l'ICM des deux sous-chaînes. Le décalage peut donc valoir probablement soit 1 ou 14 dans notre graphique ci dessus. On effectue le même procédé pour les couples (s_2,s_0) , (s_3,s_0) et (s_4,s_0), on obtient les résultats suivant : k_2-k_0 $\in \{1, 14\}$,k$_3$-k_0=15 et k_1-k_0 $\in \{3, 7, 8\}$.

Ensuite, il suffit de déterminer les valeurs possible d'un des k_i pour accéder aux autres valeurs, pour cela , on fait une analyse de fréquence d'une des sous chaînes s_i. La lettre 'e' correspond forcement a une des lettres les plus fréquentes de cette sous chaîne. Par exemple, prenons la sous chaîne s_3 , pour laquelle la seul valeur de décalage possible est 15 (elle a un ICM de 0,0844 alors que les autres sont à moins de 0.0666), en effectuant l'analyse , on obtient comme substitution éventuelle du 'e' les lettres 'u','l' et 'y' qui correspondent à des valeurs de k_3 qui valent 16(q),7(g) et 20(u). Une fois avoir étudié toutes les possibilités, on teste toutes les clés de façon exhaustive. Dans notre chiffré vigCrypt13, on trouve comme clé VHUGO et le message en clair et le texte suivant :

je ne regarderai ni laor du soir qui tombe ni les voiles au loin descendant vers har fleur et quand j'arriverai je metterai sur ta tombe un bouquet de houx verte et de bruyre en fleura

4 Conclusion

Finalement, d'un problème qui semblait très difficile à résoudre (par force brute , pour une longueur de clé qui vaut 15 ,on a 15! possibilités à tester !), on est passé à un problème qu'on peut résoudre en temps polynomiale en deux étapes. Par contre ce chiffrement est clairement vulnérable aux attaques clair connu (il suffit de faire le chiffré moins le clair).