Décryptage d'un message chiffré avec un algorithme de type ROT

[Réseaux 2] [L3 Informatique] [Nathanael Bayard] [2018/2019]

Soit le message chiffré suivant :

```
secret :=
  QZSINRFYNSQJRUJWJZWXFKJRRJJYQJUJYNYUWNSHJXTSYAJSZXHMJER
  TNUTZWRJXJWWJWQFUNSHJRFNXHTRRJOJYFNXUFWYNQJUJYNYUWNSHJFINYUZNXVZJHJXYF
  NSXNSTZXWJANJSIWTSXRFWIN
```

On cherche à le décrypter (déchiffrer sans savoir comment il a été chiffré) en supposant qu'il a été chiffré avec un algorithme de type `ROT`. Par exemple un algorithme de type `ROT(3)` décale les lettres de l'alphabet latin de trois crans vers la droite : `A` devient `D`, `Z` devient `C`, etc. Le décalage est bien sûr cyclique (modulo 26), d'où le nom (*ROTation*).

Afin de tenter de trouver la clé de chiffrement (qui ici correspond à la valeur de `n` telle que le message ait été chiffré avec `ROT(n)`), on réalise une étude statistique basique sur le contenu du message chiffré : on associe à chaque lettre de l'alphabet le nombre d'occurrences de celle-ci dans le message `secret`. Le résultat obtenu avec le script python est le suivant :

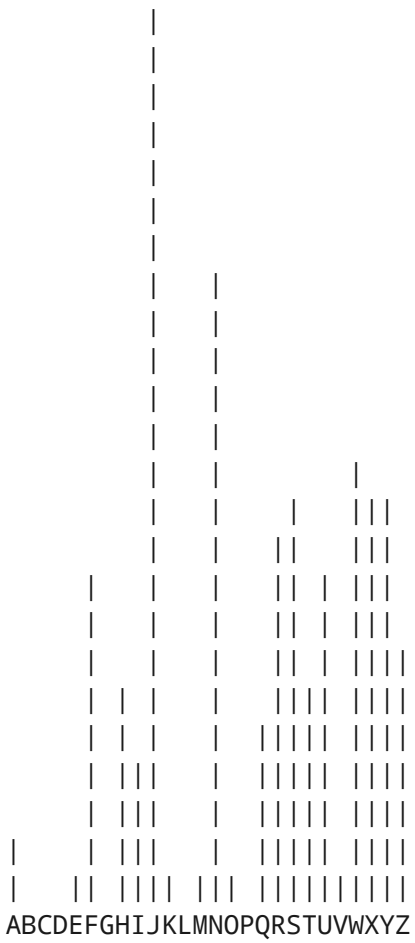
```
[2, 0, 0, 0, 1, 9, 0, 6, 4, 24, 1, 0, 1, 17, 1, 0, 5, 10, 11, 6, 9, 1, 12, 11, 11, 7]
```

À ce stade il nous manque des valeurs de référence qui nous permettraient de reconnaître si un message est "en clair" ou non rien qu'en vérifiant que l'étude statistique sur le message est en correspondance avec lesdites valeurs de référence, qui correspondent à des valeurs moyennes de nombre d'occurrences "en général" de chaque lettre d'un texte en français quelconque. L'énoncé nous donne les valeurs suivantes :

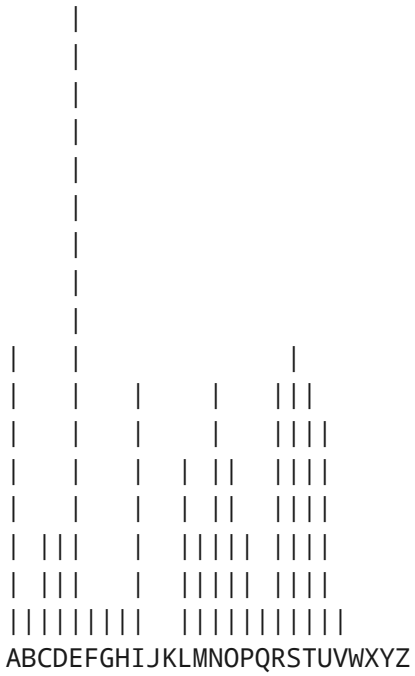
```
[8.25, 1.25, 3.25, 3.75, 17.75, 1.25, 1.25, 1.25, 7.25, 0.75, 0.0, 5.75, 3.25, 7.25, 5.75,
3.25, 1.25, 7.25, 8.25, 7.25, 6.25, 1.75, 0.0, 0.0, 0.75, 0.0]
```

L'idée est donc de représenter graphiquement les deux séries de valeurs. Le script python affiche les graphes suivants en style ASCII :

Valeurs correspondantes au message `secret` *:*



Valeurs de référence de la langue française :



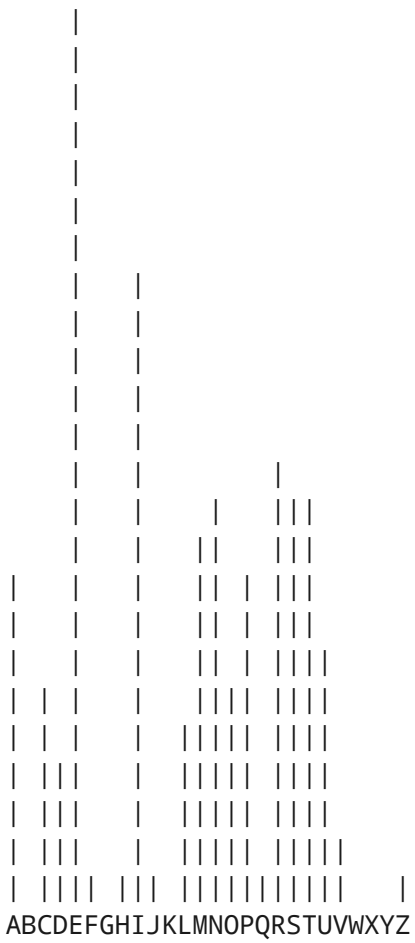
L'échelle verticale n'est bien sûr pas la même puisque le maximum pour le message `secret` est 24 (pour la lettre `J`) mais n'est que de 17 (lettre `E`) dans le cas des valeurs de référence. L'hypothèse que l'on fait est de supposer que n'importe quel texte en clair en français aura une courbe d'occurrences de forme quasi-identique à celle des valeurs de référence (sans tenir compte donc de l'échelle verticale).

Mais ce qui nous permet vraiment de déduire le chiffrement utilisé est l'observation que la courbe d'occurrences avant et après chiffrement d'un quelconque message est nécessairement la même, à un facteur de translation près correspondant exactement au facteur de rotation utilisé pour chiffrer le message d'origine. En effet, si par exemple on a une valeur d'occurrence de 42 pour la lettre `A` dans un message en clair quelconque, alors en effectuant un chiffrement `ROT(3)` sur ce message, la nouvelle valeur d'occurrence de `D == A + 3` sera nécessairement 42, car tous les `A` du message de départ seront remplacés par `D`, et aucune autre lettre ne sera remplacée par `D`. D'où les valeurs de la courbe du message en clair auront bien été décalées de 3 vers la droite.

L'observation graphique nous permet donc de constater que la courbe d'occurrence du message `secret` a une forme quasi-identique à celle de référence, en prenant en compte une translation qui remplacerait `E` par `J`.

Pour comparaison, voici ci-dessous la courbe d'occurrences du message `secret` après qu'elle ait subie une translation de valeur `-5` (déplacement de 5 vers la gauche, ce qui équivaut modulo 26 à un déplacement de 21 vers la droite). Comme dit précédemment, cela correspond à la courbe d'occurrence de `ROT(secret, -5)` c'est-à-dire celle du message `secret` après l'avoir fait passer au travers d'un chiffrement de type `ROT(-5) == ROT(21)` :

Valeurs correspondantes au message `secret` *après passage au travers de* `ROT(21)` *:*



Il n'y a plus aucun doute que cette courbe a une forme quasi-identique à celle de celle de référence. On en déduit que `ROT(secret, 21)` a les même propriétés statistiques qu'un texte en clair en français. On peut alors raisonnablement supposer que c'est bien le texte en clair que l'on cherche. On vérifie par application directe :

```
ROT(secret, -5) ==
LUNDIMATINLEMPEREURSAFEMMEETLEPETITPRINCESONTVENUSCHEZMOIPOURMESERRERLAPINCEMAISCOMMEJETAI
SPARTILEPETITPRINCEADITPUISQUECESTAINSINOUSREVIENDRONSMARDI
```

Le texte récupéré est bien en français et il veut bien "dire quelque chose". La conclusion finale est alors que le message `secret` avait à l'origine été crypté par l'algorithme `ROT(5)`, qui remplace `E` en `J`, et qui est bien l'unique algorithme `ROT` inverse de celui utilisé pour déchiffrer le message, à savoir `ROT(21)`. La clé de chiffrement que l'on recherchait est donc `n = 5`.