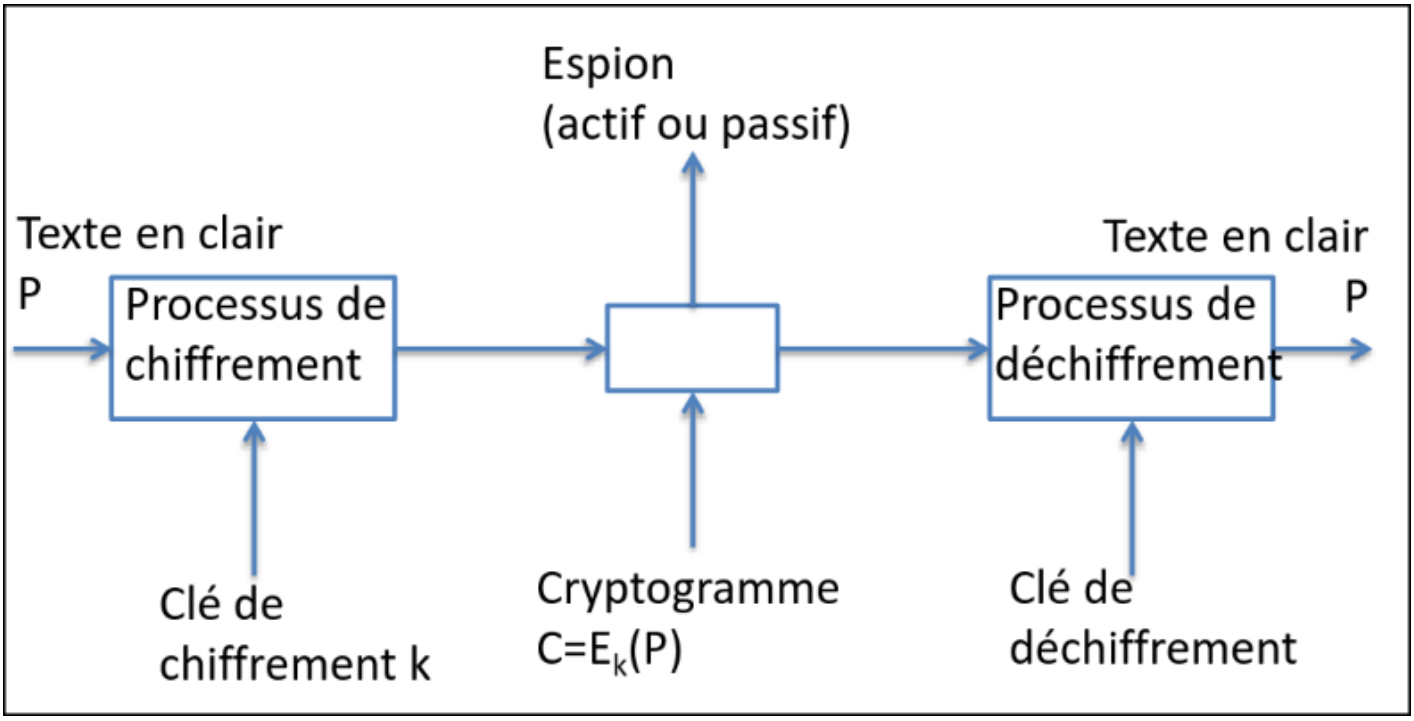
Introduction et histoire de la cryptographie

La **cryptographie** consiste à cacher des informations en utilisant des algorithmes. Il ne faut pas le confondre avec la **stéganographie** qui consiste à cacher des messages dans d'autres messages.

Par exemple, si on prend un message tel que `BONJOUR` (**texte en clair**), et que pour chaque lettre, on la décale d'un certain nombre de positions dans l'alphabet (**processus de chiffrement**), ici, on va choisir 13 positions, (13 sera ici la **clé de chiffrement**), on obtient `OBAWBHE` qui est notre **cryptogramme** (message chiffré).

Pour déchiffrer le message, il suffit alors de faire le processus inverse, ce processus inverse est donc le **processus de déchiffrement** qui retournera `BONJOUR` (notre texte en clair).



Histoire

La cryptographie est utilisée depuis très longtemps par les militaires, diplomates et amants.

Avant l'avènement de l'informatique, la cryptographie était limitée aux capacités du cerveau humain, il fallait pouvoir changer de méthode de chiffrement si quelqu'un se faisait prendre.

Avec l'avènement de l'informatique est venu la capacité de calculer des résultats beaucoup plus complexe.

Substitution et transposition

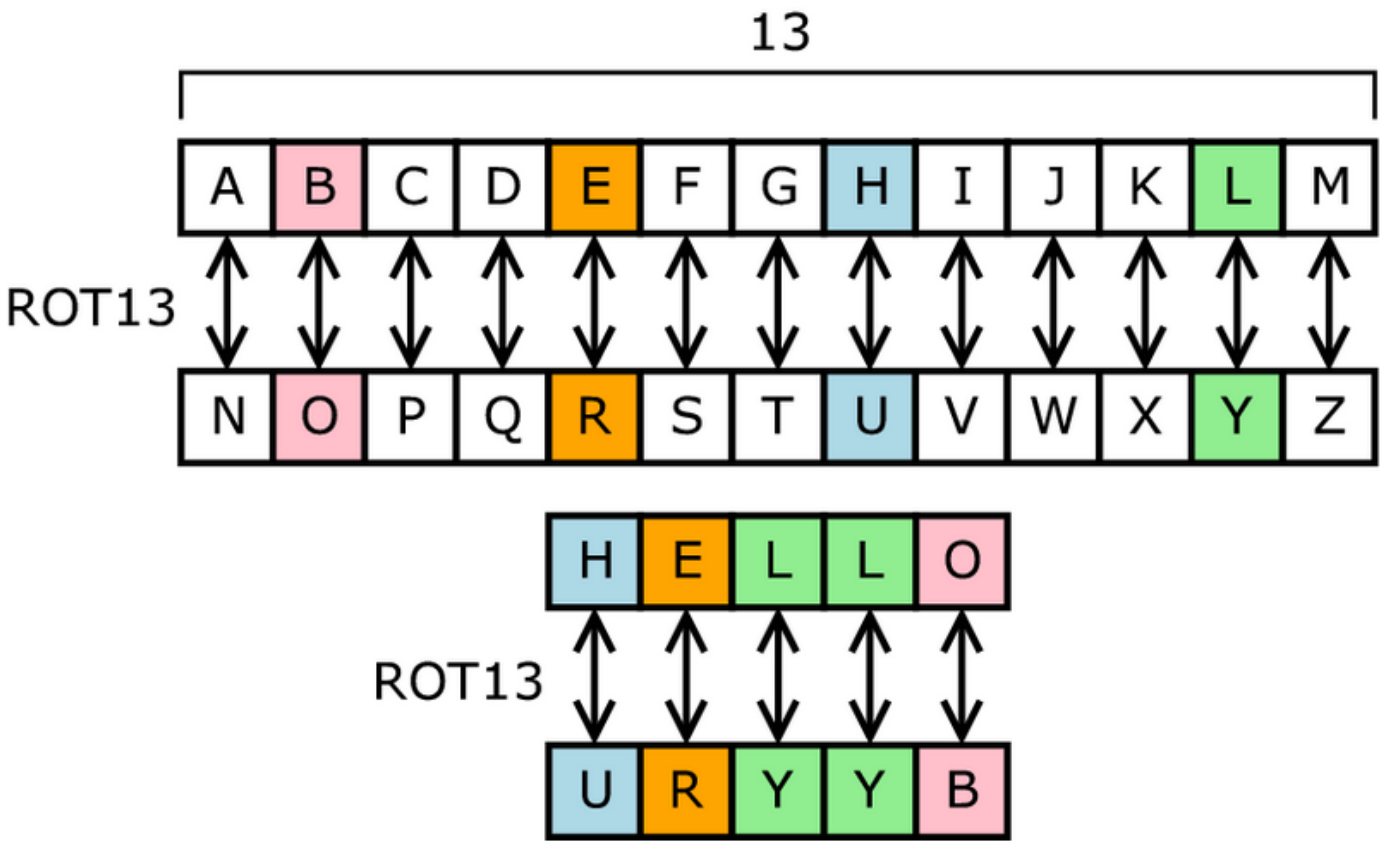
La substitution et la transposition sont deux méthodes historiques de sécurisation des données. Ces méthodes ne devraient plus être utilisées aujourd'hui pour protéger des données.

> **En sécurité, il n'y a rien de pire que d'avoir l'illusion qu'on est protégé**

Substitution

La **substitution** est un mécanisme par lequel chaque caractère du texte clair est remplacé par un autre caractère dans le texte chiffré. Cela signifie qu'on a une table de correspondance de chiffres, sons, mots ou autre à quelque chose. C'est par exemple le cas du code de césar dont nous avons parlé juste avant.

Voici, par exemple, la table de substitution du ROT13, qui est un code de César auquel la clé est à 13.

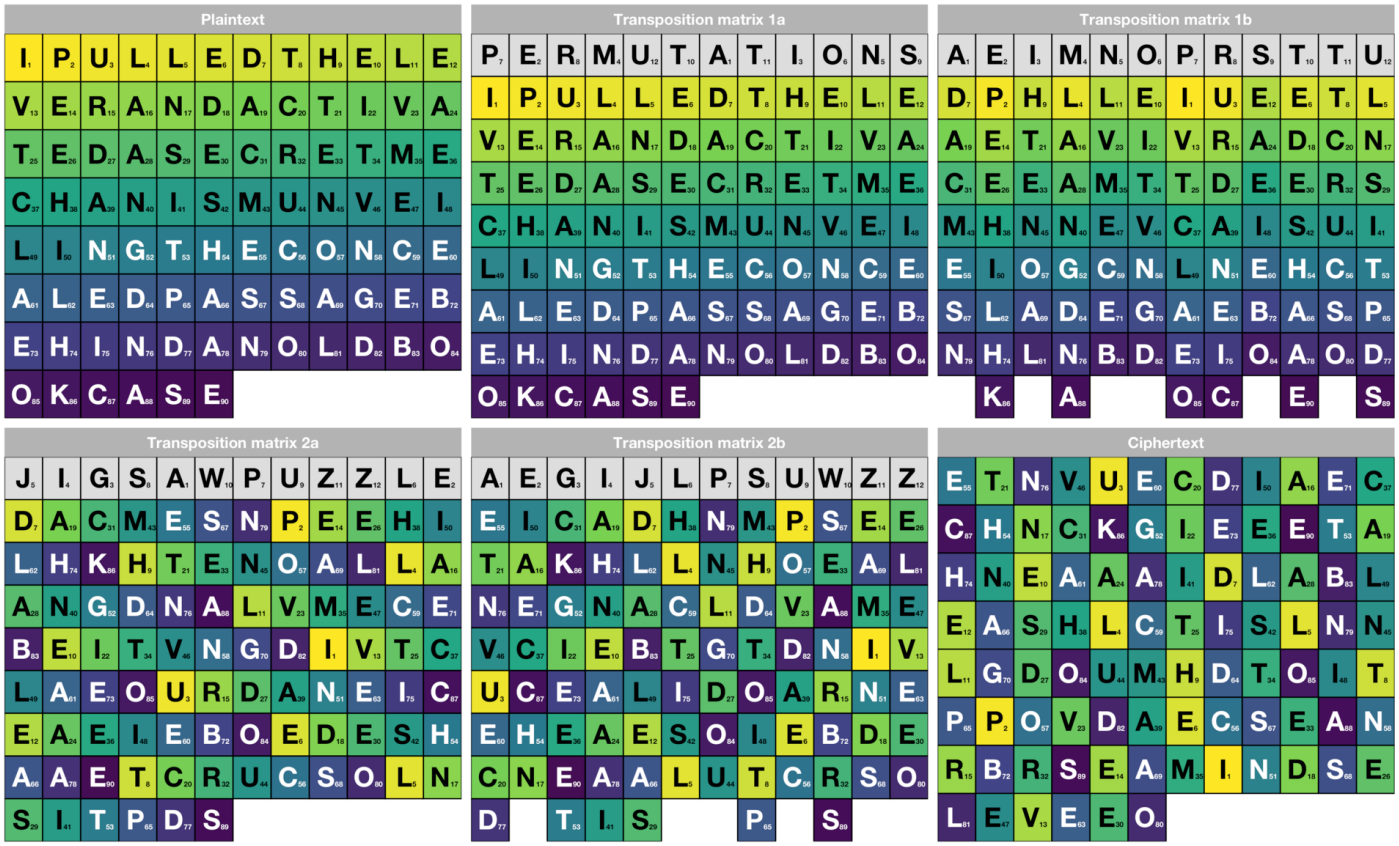


Le problème avec ce système est que l'on peut utiliser la linguistique et un contexte pour supposer la présence de certains mots ou la fréquence de certaines lettres.

Par exemple, si on sait que le texte est rédigé en français, on peut compter la lettre qui revient le plus souvent, supposer que c'est un E et compter la différence entre la lettre du cryptogramme et la lettre E pour obtenir la clé.

Transposition

La **transposition** consiste à mélanger les lettres d'une certaine manière.



> Ici par exemple, on commence par mettre le texte dans une grille, puis on ajoute un premier mot clé au-dessus et on trie les colonnes par ordre alphabétique de la clé.
>
> Ensuite, on transforme les colonnes en lignes et on met un deuxième mot clé au-dessus. On peut ensuite procéder de la même manière en triant les colonnes par ordre alphabétique de la clé. Le résultat de la grille donne donc le cryptogramme final.

Le problème avec cette méthode est similaire à celui de la substitution, on peut faire des inversions et tenter d'identifier des mots.

XOR (ou exclusif)

Le XOR est une opération de base du CPU, ce qui la rend très rapide.

L'idée ici est de faire une opération XOR entre un texte en clair et une clé de chiffrement. De la même manière, on peut déchiffrer le texte en faisant le cryptogramme XOR la clé.

Par exemple, si le message en clair est la lettre `A`, on convertit cela en binaire, ce qui donne `00001010`. Si notre clé est la lettre `H` que l'on convertit en binaire `01001000`. On peut faire un XOR dessus pour obtenir le cryptogramme :

```
   00001010 => A (texte en clair)
XOR 01001000 => H (clé)
   --------
   01000010 => B (cryptogramme)
```

Maintenant pour déchiffrer, il suffit de prendre le cryptogramme et la clé et de refaire un XOR

```
   01000010 => B (cryptogramme)
XOR 01001000 => H (clé)
   --------
   00001010 => A (texte en clair)
```

Le problème avec cette méthode est que si on connait un exemple dans lequel on a à la fois le texte clair et le cryptogramme, on peut retrouver la clé en utilisant le même mécanisme.

De la même manière, on peut facilement trouver la clé en faisant de l'analyse par fréquence sur base de la longueur de la clé.

Cependant, si la clé est complètement aléatoire et de la même longueur que le message, il s'agit alors d'un "one-time pad" ou "masque jetable" et c'est théoriquement un code incassable.

Masque jetable

Un masque jetable est une technique de chiffrement se basant sur une longue liste non répétitive et aléatoire de lettres (le masque). Chaque lettre est utilisée pour code une lettre du texte clair.

Par exemple, on peut additionner le rang de la lettre du masque avec celui du texte clair modulo 26 pour obtenir le rang de la lettre du texte chiffré.

Ou alors, on peut procéder en utilisant un XOR comme précédemment.

Puis ce que la clé est de la même longueur que le message et qu'elle est entièrement aléatoire, il est impossible de la déchiffrer. Cependant, cette technique a le gros désavantage d'avoir des clés très longues et peu pratiques.

Niveaux de chiffrements

Le niveau de chiffrement sera établi en fonction des groupes de personne contre lesquels on désire se protéger. Le niveau sera différent si on souhaite se protéger contre ses concurrents ou de la NSA.

C'est également une question de longueur de clé. Plus la clé est longue, plus ce sera sécurisé. Par exemple, un cadenas à trois chiffres aura 1 000 combinaisons possibles tandis qu'un cadenas à six chiffres aura 1 000 000 combinaisons possibles.

Algorithme comme garant de la sécurité

L'algorithme qui est utilisé pour faire de la cryptographie est le garant (avec la ou les clés) de la sécurité des messages.

Un bon algorithme doit être public (pour être vérifiable), sûr (éprouvé pendant plusieurs années et par des experts) et indépendant (sans coopération avec des organismes ayant des intérêts contradictoires tels que la NSA).

Dans un algorithme sûr, que l'espion ne soit qu'avec du texte chiffré, avec des correspondances texte en clair et texte chiffré ou même avec du texte clair choisi, l'espion ne peut pas trouver la clé.

Revision #2
Created 6 January 2024 16:48:00 by SnowCode
Updated 6 January 2024 19:13:15 by SnowCode