3A : Chiffrement par substitution

Louiza Khati

1 Chiffrement de César

Le chiffrement de César est le chiffrement le plus connu et l'un des plus simples. Il s'agit d'un chiffrement mono-alphabétique.

1. Rappeler le principe de ce chiffrement.
2. Quelle est la clé et sa taille ?

Vous avez à votre disposition le script `cesar.py` à remplir lors de ce TP.

3. Ecrire une fonction `encrypt()` qui prend en paramètre un entier `key` et une chaîne de caractères `plaintext` et qui renvoie le `plaintext` avec chacune des lettres décalées de `key` position dans l'alphabet. Dans un premier temps, le mot `plaintext` ne contient que des lettres minuscules. Tester votre fonction.
4. Ecrire la fonction de déchiffrement `decrypt()` correspondante (de même, elle prend en entrée un entier `key` et une chaine `ciphertext` et renvoie le `plaintext` correspondant) et la tester sur quelques exemples. Une fois cette question terminée, valider avec l'intervenant.
5. Comment attaquer facilement ce chiffrement ? Retrouver le clair correspondant à la valeur `ciphertext` dans `cesar.py`.
6. **Bonus** Choisir une clé secrète et utiliser la fonction `encrypt()` pour chiffrer une phrase de votre choix. Envoyer ce chiffré à votre voisin et demandez lui de retrouver le clair correspondant. De même, vous allez recevoir un chiffré, retrouvez le clair.
7. **Bonus** Enrichir les fonctions `encrypt` et `decrypt` en prenant en compte les majuscules, les minuscules et les caractères spéciaux. Les caractères spéciaux restent inchangés dans un premier temps (autrement dit, ils ne sont pas chiffrés).

2 Substitution totale

Le chiffrement de César n'étant pas robuste, nous allons implémenter une substitution totale. Une **substitution** est une fonction qui va remplacer chaque lettre d'une texte par un symbole (qui peut être une lettre). Le chiffrement de César est un chiffrement mono-alphabétique particulier où chaque lettre est décalée dans l'alphabet. Dans cet exercice, par substitution totale, on entend un chiffrement mono-alphabétique sans cette contrainte. Utilisez le script `substitution.py` pour cet exercice.

1. Écrire une fonction `substitution()` qui va remplacer chaque lettre de l'alphabet par une autre (les caractères spéciaux restent inchangés pour faciliter l'implémentation). Elle prendra en entrée deux tables (l'alphabet clair et l'alphabet chiffré correspondant) . Quelle est la clé de chiffrement dans ce cas ?

2. Réfléchir à un point faible de cette méthode de chiffrement vu en cours. Implémenter une attaque, que vous nommerez `attack()`, grâce à cette faiblesse et retrouver le clair correspondant à la variable `ciphertext` dans le fichier `substitution.py`. (Pour rappels, les caractères spéciaux ne sont pas chiffrés).
3. Il y a, peut-être, une lettre qui n'est pas retrouvée automatiquement. Donner une explication.
4. **Bonus** Choisissez une substitution et chiffrer un texte secret de votre choix. Donner le chiffré à votre voisin pour qu'il tente de l'attaquer. De même, vous recevrez un chiffré, essayez de retrouver le clair. Que constatez-vous ?
5. Quelle remarque pouvez-vous faire sur les conditions de succès de cette attaque ? Autrement dit pourquoi fonctionne-t-elle étonnamment bien dans notre exemple ?
6. **Bonus** Pour aller plus loin : Enrichir les fonctions pour détecter les erreurs : par exemple, si le type des entrées des fonctions n'est pas correcte, etc. . ..