Cycle Fondamental
2A S2 Math/Info (MAT2053)
28/01/2019

ESIEA Paris/Ivry-sur-Seine
Année Académique 2018/2019

Cryptographie

Intervenant : Michael FRANÇOIS (`francois@esiea.fr`)

TD1_3 - - Cryptanalyse du Chiffre de César

A. Travail à faire sur papier

- ► **1.** Qu'est-ce que la cryptanalyse ?
- ► **2.** Dans le jargon de la cryptographie, que signifie une attaque ?
- ► **3.** Que signifie une attaque par force brute ($i.e.$ recherche exhaustive) ? Quand peut-on utiliser cette attaque ?

B. Travail à faire sur ordinateur (prog. C)

a) Cryptanalyse par analyse de fréquences

Un long texte écrit en français a été chiffré en utilisant la méthode de J. César. On ne connaît pas la clé ($i.e.$ décalage) utilisée. Le but ici est de monter une cryptanalyse en se basant sur la fréquence d'apparition de chaque lettre dans le texte chiffré. Le texte chiffré `chiffre1.txt` se trouvant sur le moodle, sera utilisé pour cette cryptanalyse.

- ► **1.** Écrire une fonction `CALC_FREQ_LETTRES_ALPHA`, qui permet de calculer la fréquence d'apparition de chaque lettre de l'alphabet dans un texte. Le tableau des fréquences et le nom du fichier contenant le texte sont donnés en paramètres. Le prototype de la fonction est le suivant :

```
void CALC_FREQ_LETTRES_ALPHA (float alpha[26], char * nom_fic);
```

- ► **2.** Écrire une fonction `AFFIC_PLUS_GRDE_FREQ`, qui permet d'afficher à l'écran la plus grande fréquence ainsi que la lettre correspondante, depuis un tableau donné en paramètre. Voilà le prototype de la fonction :

```
void AFFIC_PLUS_GRDE_FREQ (float alpha[26]);
```

- ► **3.** Quelle est la clé ($i.e.$ décalage) utilisée lors du chiffrement ?
- ► **4.** Utiliser la fonction `DECHIFF_CESAR`, vue au TD précédent pour retrouver le texte initialement caché.

b) Attaque par force brute (brute-force attack)

Le but ici est de monter une attaque par force brute, sachant que la taille de l'espace des clés n'est pas grande. Le texte chiffré `chiffre2.txt` sur lequel se fera l'attaque, se trouve sur le moodle.

► **1.** Écrire une fonction `RECHERCHE_EXHAUSTIVE`, qui permet de tester toutes les clés possibles pendant le déchiffrement. La fonction prend en paramètres le nom du fichier chiffré ainsi que celui qui contiendra tous les déchiffrés possibles. Le prototype est le suivant :

```
void RECHERCHE_EXHAUSTIVE (char * fic_chiff, char * fic_dechiff) ;
```

Il faudra écrire une nouvelle fonction `DECHIFF_JULES_CESAR_2`, qui ouvre cette fois-ci le fichier `fic_dechiff` en mode "a" pour mettre les textes déchiffrés les uns à la suite des autres. Vous pouvez réutiliser la fonction `DECHIFF_JULES_CESAR` vue précédemment.

Dans le fichier `fic_dechiff`, il faut afficher d'abord la clé suivie du texte déchiffré correspondant.

► **2.** Retrouver ensuite visuellement le texte clair initialement caché.