Cryptographie

Intervenant : Michael FRANÇOIS (francois@esiea.fr)

TD2_1&2 - - Chiffrement de Vigenère

A. Travail à faire sur papier

► **1.** Expliquer le principe de la substitution poly-alphabétique.

► **2.** Expliquer rigoureusement le principe du chiffrement de Vigenère.

► **3.** Utiliser le chiffrement de Vigenère avec la clé "IVRY", pour chiffrer le message suivant :

"CRYPTOGRAPHIE SYMETRIQUE"

► **4.** Le chiffrement de Vigenère a été utilisé pour chiffrer un message avec la clé "ESIEA". Le chiffré obtenu est le suivant :

"XZMSRMWLISRGUFRIK"

Déchiffrer ce message et retrouver le message initialement caché.

► **5.** Voici un message clair et son chiffré correspondant. Retrouver la clé utilisée lors du chiffrement.

Texte clair	R	O	L	E	D	E	S	A	L	G	O	R	I	T	H	M	E	S	E	N	I	N	F	O	R	M	A	T	I	Q	U	E
Texte chiffré	G	G	R	T	V	K	H	S	R	V	G	X	X	L	N	B	W	Y	T	F	O	C	X	U	G	E	G	I	A	W	J	W

► **6.** Quel avantage possède le chiffrement de Vigenère par rapport à celui de César ?

► **7.** Dans le cas où la longueur de la clé utilisée lors du chiffrement est connue, comment peut-on monter une attaque afin de retrouver les éléments de la clé ?

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

a) Chiffrement / Déchiffrement de Vigenère

Le but ici est de créer un programme permettant de chiffrer/déchiffrer un texte depuis un fichier, en utilisant la méthode de Vigenère.

► **1.** Tout d'abord, écrire une fonction `DETERM_LONG_TEXTE`, permettant de calculer et renvoyer la longueur d'un texte contenu dans un fichier. On prend en compte uniquement les lettres de l'alphabet. Le prototype est le suivant :

```
int DETERM_LONG_TEXTE (char * nom_fic);
```

► **2.** Écrire une fonction `CHARGER_CLE`, permettant de charger la clé de chiffrement depuis un fichier. Le prototype est donné par :

```
void CHARGER_CLE (int T, char cle[T], char * nom_fic_cle);
```

► **3.** Écrire une fonction `CHIFF_VIGENERE`, permettant de chiffrer un texte depuis un fichier en utilisant la méthode de Vigenère. Le prototype est donné par :

```
void CHIFF_VIGENERE (int T, char cle[T], char * nom_fic_clair, char * nom_fic_chiff);
```

Tester cette fonction sur l'exemple vu dans la première partie.

► **4.** Écrire une fonction `DECHIFF_VIGENERE`, permettant d'inverser la fonction de chiffrement précédente. Le prototype est donné par :

```
void DECHIFF_VIGENERE (int T, char cle[T], char * nom_fic_chiff, char * nom_fic_dechiff);
```

Tester cette fonction sur les exemples d'avant.

b) Recherche et extraction de la clé en connaissant un couple (clair, chiffré)

On considère ici que l'on possède un texte clair ainsi que son chiffré correspondant. L'objectif est d'extraire le motif de la clé qui est répété lors du chiffrement. Vous pouvez prendre l'exemple de votre choix.

► **1.** Écrire une fonction `RECHERCHE_CLE`, permettant de retrancher le texte clair à son texte chiffré suivant la méthode de Vigenère. Le résultat qui est une répétition de la clé utilisée lors du chiffrement, sera stocké dans un fichier. Le prototype est le suivant :

```
void RECHERCHE_CLE (char * nom_fic_clair, char * nom_fic_chiff, char * nom_fic_cle_rep);
```

Normalement si tout s'est bien passé, on est capable de déterminer visuellement la clé de chiffrement.

► **2.** Écrire une fonction `EXTRACTION_CLE`, permettant d'extraire exactement depuis le fichier `nom_fic_cle_rep` la clé de chiffrement. La clé résultante sera affichée à l'écran. Réfléchir à la stratégie à adopter afin d'extraire que le motif correspondant à la clé utilisée. Le prototype de la fonction est donné par :

```
void EXTRACTION_CLE (char * nom_fic_cle_rep);
```