INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

Département Sécurité et Technologies Informatiques

Cryptograhie – TD1

Jérémy Briffaut <jeremy.briffaut@insa-cvl.fr>

Jean-Christophe Deneuville <jean-christophe.deneuville@insa-cvl.fr>

Lundi 17 septembre 2018

Exercice 1 Ou exclusif, *aka* Xor

Le Xor est l'opération ou exclusif, notée ˆ dans bien des langages (le C notamment) ou $\oplus$ en mathématiques. C'est une opération classqiue sur les bits :

$\oplus$	0	1
0	0	1
1	1	0

$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$
$$1 \oplus 0 = 1$$
$$1 \oplus 1 = 0$$

1. Échauffement : Calculer $2018 \oplus 42$.
2. Soient deux bits $a, b \in \{0, 1\}$.
   a) Montrer que $a \oplus a = 0$.
   b) En déduire que $a \oplus b \oplus b = a$

On considère maintenant le code suivant.

```c
#include <stdio.h>
/*
* Usage: main key input_file output_file
*/
int main(int argc, char *argv[]) {
    FILE *fi, *fo;
    char *cp;
    int c;
    if (cp = argv[1]) {
        if ((fi = fopen(argv[2], "rb")) != NULL) {
            if ((fo = fopen(argv[3], "wb")) != NULL) {
                while ((c = getc(fi)) != EOF) {
                    if (*cp == '\0' ) cp = argv[1];
                    c ^= *(cp++);
                    putc(c,fo);
                }
                fclose(fo);
            }
            fclose(fi);
        }
    }
    return 1;
}
```

3. Que fait ce programme ?
4. Comment chiffrer un texte avec cette fonction ? Comment le déchiffrer ?
5. Tester cette fonction.
6. Quelle propriété assure le fonctionnement correct du chiffrement et du déchiffrement ?

Exercice 2 Cæsar et Vigenère

La clé du chiffré de Vigenère est une séquence finie de d lettres. Le chiffrement de la première lettre du clair se fait en ajoutant modulo 26 la première lettre du clair et la première lettre de la clé (A=0, ..., Z=25). La deuxième lettre du chiffré s'obtient en ajoutant les deuxièmes lettres du clair et de la clé, etc... Lorsque la clé est épuisée, on la reprend au début. Par exemple, si l'on chiffre "CRYPTOGRAPHIE" avec "BATO", on obtient "DRRDUOZFBPAWF".

On appelle "chiffre de César" le cas particulier du chiffre de Vigenère avec une clé de taille 1. Le chiffre de César est très facile à cryptanalyser (c'est-à-dire le déchiffrer sans connaître la clé) par analyse de fréquence (une fois repérée la lettre E, le décalage est connu).

1. En utilisant le chiffre de César avec la clé "D", chiffrer le message "CESTTROPFACILE".
2. Écrire (en C) un programme qui prend en paramètre un fichier et une clé (décalage pour le chiffrement de césar) et qui écrit le chiffré obtenu à partir du fichier d'entrée et de la clé dans un nouveau fichier. Par exemple, l'appel `./cesar text.txt 3` chiffrera le fichier `text.txt` avec la clé "D" dans le fichier `text.txt.crypt`.
3. En utilisant le chiffre de Vigenère avec la clé "VIGENERE", chiffrer le message "VRAIMENTETONNANTCECI".
4. Écrire (toujours en C) un programme qui prend en paramètre un fichier et une clé (une chaîne de caractères) et qui écrit le chiffré obtenu à partir du fichier d'entrée et de la clé dans un nouveau fichier. Par exemple, l'appel `./viginere text.txt VIGENERE` chiffrera le fichier `text.txt` avec la clé "VIGENERE" dans le fichier `text.txt.crypt`.

Exercice 3 Création d'une librairie cryptographique

À partir des deux exercices précédents, nous allons commencer l'implantation d'une librairie cryptographique.

Attention !

Cette librairie vous suivra tout au long de ce module et comptera pour une partie de votre note finale. Nous ne pouvons que vous recommander de la développer avec le plus grand soin (commentaires, débuggage, optimisation, ...).

1. Créer deux fichiers `crypto.h` et `crypto.c`. Dans le fichier `crypto.h`, définir les prototypes suivants :

```
1 /*
2 * Chiffrement utilisant le ou exclusif
3 */
4 void xor_crypt(char* key, char* texte, char* chiffre);
5 /*
6 * Déchiffrement utilisant le ou exclusif
7 */
8 void xor_decrypt(char* key, char* chiffre, char* clair);
9 /*
```

```c
/*
* Chiffrement utilisant cesar
*/
void cesar_crypt(int decallage, char* texte, char* chiffre);
/*
* Déchiffrement utilisant cesar
*/
void cesar_decrypt(int decallage, char* chiffre, char* clair);
/*
* Chiffrement utilisant viginere
*/
void viginere_crypt(char* key, char* texte, char* chiffre);
/*
* Déchiffrement utilisant viginere
*/
void viginere_decrypt(char* key, char* chiffre, char* clair);
/*
* Chiffrement utilisant des
*/
void des_crypt(char* key, char* texte, char* chiffre, int size);
/*
* Déchiffrement utilisant des
*/
void des_decrypt(char* key, char* chiffre, char* clair, int size);
/*
* Chiffrement utilisant 3des
*/
void tripledes_crypt(char* key1, char* key2, char* texte, char* chiffre,
↪  int size);
/*
* Déchiffrement utilisant 3des
*/
void tripledes_decrypt(char* key1, char* key2, char* chiffre, char*
↪  clair, int size);
/*
* Chiffrement RSA
*/
void rsa_crypt(int e, int n, char* texte, char* chiffre, int size);
/*
* Déchiffrement RSA
*/
void rsa_decrypt(int d, int n, char* chiffre, char* clair);
```

2. Intégrer votre code issu des réponses précédentes au fichier `crypto.c`.