Université de Limoges

Développement Logiciel Cryptographique

TP n° 2

Exercice 1

Écrire deux fonctions `RotateLeft()` et `RotateRight()` qui prennent en entrée deux entiers a et n de type `unsigned int` et qui renvoient la valeur de a ayant subi une rotation de n bits.

Le résultat dépend-il du type choisi pour représenter cet entier ?

Exercice 2

Écrire les fonctions similaires en macros `#define`.

Exercice 3

Soit $a = (a_7 a_6 \dots a_1 a_0)_2$ un `unsigned char`. Écrire un programe qui saisit l'entier a et qui calcule l'entier b formé à partir de la permutation (3,6,1,0,2,7,4,5) des bits de a. L'écriture binaire de b devra donc être $b = (a_3 a_6 a_1 a_0 a_2 a_7 a_4 a_5)_2$.

Exercice 4

Le programme `exhaustive_0.c` donné en annexe est un template de programme de recherche exhaustive d'une clé de 4 octets.

Écrivez la partie manquante de ce programme pour effectivement réaliser cette recherche exhaustive.

Exercice 5

Écrivez une version du programme précédent qui puisse fonctionner quelle que soit la taille de la clé, ce paramètre étant défini en macro `#define`.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// On définit le type BYTE

typedef unsigned char BYTE;

BYTE secret_key[4] = {0x0B, 0x2F, 0xD9, 0x24};

// ------------------------------------------------

int encrypt(BYTE* c, BYTE* m, BYTE* k)
{
  int i;

  for (i = 0; i < 4; i++)
  {
    c[i] = m[i] ^ k[i];
  }

  return 0;
}

// ------------------------------------------------

int main()
{
  BYTE m0[4] = {0xCA, 0x08, 0xFE, 0x35};
  BYTE c0[4];
  BYTE k[4];

  int i;

  // Génère un couple clair-chiffré de référence
  encrypt(c0, m0, secret_key);

  printf("\n");
  printf("Message de reference : ");
  for (i = 0; i < 4; i++)
  {
    printf(" %02X", m0[i]);
  }
```

```
  printf("\n");
  printf("Chiffre de reference : ");
  for (i = 0; i < 4; i++)
  {
    printf(" %02X", c0[i]);
  }
  printf("\n");
  printf("\n");

  // Recherche exhaustive

  /*

    Pour chaque candidat clé k de 4 octets il faut tester si le chiffré de m0
    sous la clé k est égal à c0. On s'arrête alors en affichant k

    A vous de jouer ...

  */

  return 0;
}
```