

## TP3

### Implémentation du chiffre RC4

Ce TP porte sur l'implémentation et la mise en œuvre du système de chiffrement symétrique RC4 en langage C.

## Système de chiffrement symétrique RC4

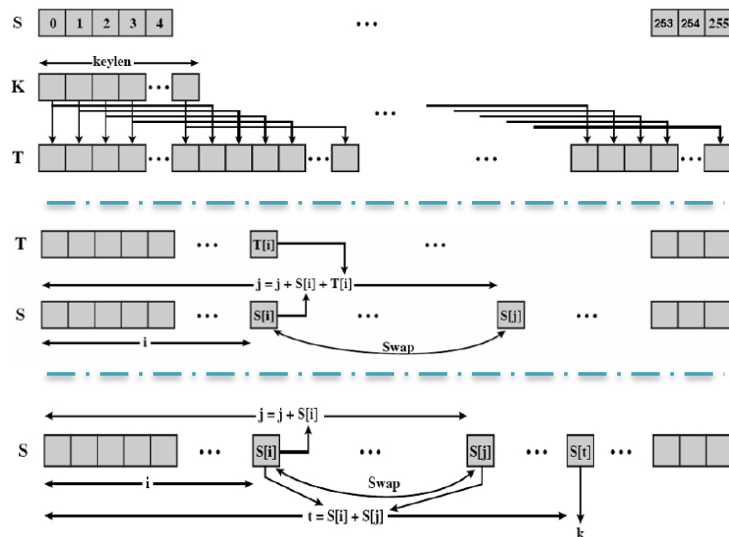
- Conçu par RSA Labs (Ronald Rivest), 1987
- Publié anonymement sur Internet en 1994
- **Chiffrement par flot** à clé de taille variable ( $\leq 2048$  bits) : un générateur de bits pseudo-aléatoires combiné avec le texte en clair via une opération XOR
- Le déchiffrement se fait de la même manière
- Algorithme très rapide (5 fois DES et 15 fois 3DES) mais considéré maintenant comme peu sûr (différentes attaques potentielles connues)
- Utilisé dans WEP et WPA (Wi-Fi)
- RC4-40 a été cassé en 1995
- Principe du système RC4
  - Initialisation du tableau S (suite pseudo-aléatoire) à partir de la clé secrète K
  - Tant que flux d'entrée non vide
    - calcul d'une suite aléatoire d'octets S // Réorganisation à chaque tour
    - $m$  = octet courant du flux d'entrée //  $m$  : un octet du message clair
    - $c = S[?] \text{ XOR } m$  //  $c$  : un octet du chiffré ;  $S[?]$  : un octet de S

- Structures de données
  - Tableau K d'octets // La clé
  - Tableau S de 256 octets // La suite pseudo-aléatoire
  - Tableau T de 256 octets // Pour l'initialisation de la suite pseudo-aléatoire

```
/* Initialisation de S et T */
Pour i de 0 à 255 Faire
  S[i] = i ;
  T[i] = K[i (mod keylen)] ;
FinPour
```

```
/* Première permutation de S */
j = 0 ;
Pour i de 0 à 255 Faire
  j = (j + S[i] + T[i]) (mod 256) ;
  Swap(S[i], S[j]) ;
FinPour
```

```
/* Codage */
i, j = 0 ;
TantQue (m = LectureOctetFluxEntrée()) Faire
  i = (i + 1) (mod 256) ;
  j = (j + S[i]) (mod 256) ;
  Swap(S[i], S[j]) ;
  t = (S[i] + S[j]) (mod 256) ;
  c = m XOR S[t] ;
  EcritureOctetFluxSortie(c) ;
FinTantQue
```



## Travail à faire

- 1) Récupérez les fichiers *RC4.h* et *test\_RC4.c* sur UMTICE (fichier implementation.zip)
- 2) Créez et éditez le fichier *RC4.c* en implémentant les fonctions de l'algorithme RC4.
- 3) Compilez le programme de test avec la commande : `gcc RC4.c test_RC4.c -o test_RC4`

Vous pouvez, si vous le souhaitez, définir un fichier d'aide à la compilation de type *makefile*.

L'exécution du programme de test doit être le suivant :

```
ic2-aragorn:solution lemeunie$ gcc RC4.c test_RC4.c -o test_RC4
ic2-aragorn:solution lemeunie$ ./test_RC4
BBF316E8D940AF0AD3
1021BF0420
45A01F645FC35B383552544B9BF5
ic2-aragorn:solution lemeunie$
```

- 4) Que faire pour déchiffrer un message précédemment chiffré avec RC4 ?

Pour montrer comment procéder au déchiffrement, vous partirez de la fonction de chiffrement suivante :

$$E(k, m) = k \oplus m = c$$

avec  $c$  le texte chiffré,  $m$  le texte clair,  $k$  la clé et  $\oplus$  l'opération XOR.

(a) Que vaut  $c$  lorsque  $k=01011011$  (en binaire) et  $m=1001\ 0010$  (en binaire) ?

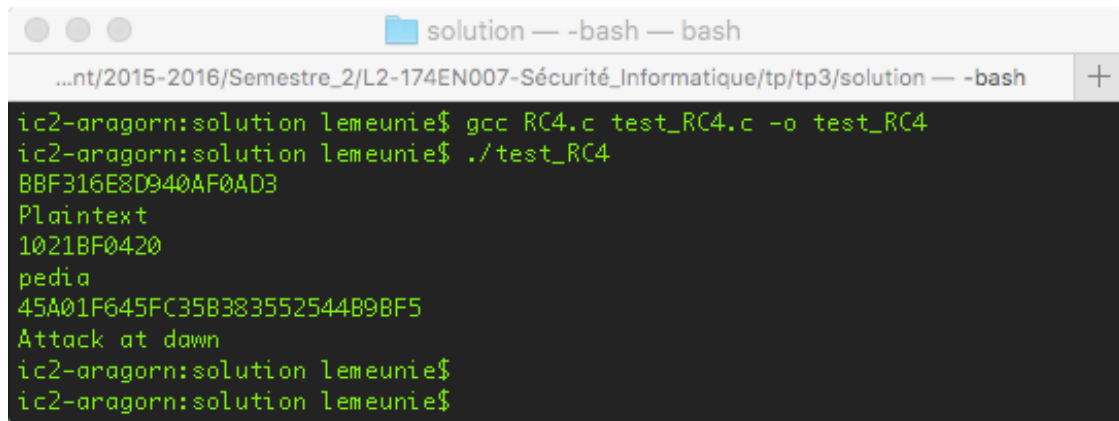
(b) Quel est le résultat de l'opération  $c \oplus k$  ?

Réessayez (a) puis (b) avec d'autres valeurs pour  $k$  et  $m$ .

Ecrivez alors la fonction  $D(k, c)$ .

Modifiez les fichiers nécessaires (y compris `test_RC4.c`) pour déchiffrer.

L'exécution du programme de test devra maintenant être le suivant :



```
ic2-aragorn:solution lemeunie$ gcc RC4.c test_RC4.c -o test_RC4
ic2-aragorn:solution lemeunie$ ./test_RC4
BBF316E8D940AF0AD3
Plaintext
1021BF0420
pedia
45A01F645FC35B383552544B9BF5
Attack at dawn
ic2-aragorn:solution lemeunie$
ic2-aragorn:solution lemeunie$
```

5) Modifiez le programme de test pour chiffrer avec une même clé deux messages proches l'un de l'autre. Par exemple testez avec :

$text4 = "From: Bob"$  et  $text5 = "From: Eve"$ .

Que remarquez-vous sur le chiffrement de  $m1$  par rapport au chiffrement de  $m2$  ?  
Donnez une explication.

6) Vérifiez (en modifiant le programme de test) que :

$$\text{Si } c' = E(k, m) \oplus p = m \oplus k \oplus p$$

$$\text{Alors } D(k, c') = m \oplus p$$

avec pour  $m$  la valeur `text4` de la question précédente

et  $p = \{0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x19, 0x07\}$ .

Qu'obtenez-vous comme valeur déchiffrée par  $D(k, c')$  ? A quoi correspond  $p$  ? Est-ce que  $D(k, c')$  correspond bien à  $m \oplus p$  ?

Donnez une explication.

## Travail à rendre

Dans un fichier nommé `Prénom_Nom_RC4.zip` (remplacez `Prénom_Nom` par vos propres prénom et nom), compressez au format Zip les fichiers `RC4.h`, `RC4.c` et `test_RC4.c` (+ éventuellement un fichier `makefile`) ainsi que les réponses aux questions puis déposez le fichier Zip sur UMTICE.