Devoir Maison: Cryptographie Symétrique - Rapport

ALI AHMEDI Mycipssa

BONNET Ludivine

MOLINER Emma

April 2020

1 Générateur de type Geffe pour le chiffrement à flot

1. Le programme correspondant à cette question se trouve dans le fichier *dm.c*

2. Pour calculer la corrélation entre la sortie s de la suite chiffrante et la sortie x d'un LFSR, il faut calculer la probabilité s_i = x avec i < taille de s à partir de la table représentant la fonction de filtrage F. Si cette probabilité est différente de 1/2 , on considère qu'il y a une corrélation.
   Le programme correspondant à cette question se trouve dans le fichier *probas.c*

3. L'attaquant connaît la suite chiffrante s et la fonction de filtrage F(f0, f1, f2, f3, f4, f5, f6, f7) = (1, 0 , 0, 0, 1, 1, 1, 0).
   On note y, un bit de la suite chiffrante s et x_i la sortie du LFSR i. On calcule la probabilité de corrélation entre les deux.
   P($x_0 = s_i$) = 2/8 = 1/4
   P($x_1 = s_i$) = 2/8 = 1/4
   P($x_2 = s_i$) = 2/8 = 3/4
   L'attaque montée est une attaque par corrélation. Le but de cette attaque est de traiter indépendamment les initialisations des LFSR à l'aide de leur corrélation avec la suite s. Pour chaque LFSR, on teste toutes les initialisations possibles jusqu'à trouver des clés potentielles où le nombre de similitudes avec la suite chiffrante correspond environ à la probabilité de corrélation. On effectue l'opération pour chaque LFSR et on fait une recherche exhaustive avec les clés qu'on a trouvé pour chaque LFSR.

4. La complexité de l'attaque est de $3*2^{16}$ en temps et de 0 en mémoire. La complexité de la recherche exhaustive est de 2^{48} en temps et de 0 en mémoire.

5. Le programme correspondant à cette question se trouve dans le fichier *attaque.c*.

6. On prend F = 11111111 et on obtient

   $P(x_0 = s_i) = 1/2$

   $P(x_1 = s_i) = 1/2$

   $P(x_2 = s_i) = 1/2$

   Cette fonction F rend l'attaque contre ce générateur très difficile.

2 Un chiffrement par bloc faible

1. D'après le schéma, on sait que :

   $x_{R+1}^L = ((x_R^L \oplus x_R^R) <<< 7) \oplus k_0$

   et

   $x_{R+1}^R = (((((x_R^L \oplus x_R^R) <<< 7) \oplus k_0) \oplus x_R^R) <<< 7) \oplus k_1$

   $x_{R+1}^R = ((x_{R+1}^L \oplus x_R^R) <<< 7) \oplus k_1$

   donc

   $x_1^L = ((x_0^L \oplus x_0^R) <<< 7) \oplus k_0$

   $x_1^L = ((0100\ 0101\ 0000\ 0001\ 1001\ 1000\ 0010\ 0100 \oplus 0101\ 0001\ 0000\ 0010\ 0011\ 0011\ 0010\ 0001) <<< 7) \oplus 0000\ 0001\ 0000\ 0010\ 0000\ 0011\ 0000\ 0100$

   $x_1^L = (0001\ 0100\ 0000\ 0011\ 1010\ 1011\ 0000\ 0101 <<< 7) \oplus 0000\ 0001\ 0000\ 0010\ 0000\ 0011\ 0000\ 0100$

   $x_1^L = 0000\ 0001\ 1101\ 0101\ 1000\ 0010\ 1000\ 1010 \oplus 0000\ 0001\ 0000\ 0010\ 0000\ 0011\ 0000\ 0100$

   $x_1^L = 0000\ 0000\ 1101\ 0111\ 1000\ 0001\ 1000\ 1110$

   x_1^L **= 0x00d7818e**

   $x_1^R = (((((x_0^L \oplus x_0^R <<< 7) \oplus k_0) \oplus x_0^R <<< 7) \oplus k_1$

   $x_1^R = ((x_1^L \oplus x_0^R <<< 7) \oplus k_1)$

   $x_1^R = ((0000\ 0000\ 1101\ 0111\ 1000\ 0001\ 1000\ 1110 \oplus 0101\ 0001\ 0000\ 0010\ 0011\ 0011\ 0010\ 0001) <<< 7) \oplus 1001\ 1000\ 0111\ 0110\ 0101\ 0100\ 0011\ 0010$

   $x_1^R = (0101\ 0001\ 1101\ 0101\ 1011\ 0010\ 1010\ 1111 <<< 7) \oplus 1001\ 1000\ 0111\ 0110\ 0101\ 0100\ 0011\ 0010$

   $x_1^R = 1110\ 1010\ 1101\ 1001\ 0101\ 0111\ 1010\ 1000 \oplus 1001\ 1000\ 0111\ 0110\ 0101\ 0100\ 0011\ 0010$

   $x_1^R = 0111\ 0010\ 1010\ 1111\ 0000\ 0011\ 1001\ 1010$

   x_1^R **= 0x72af039a**

2. On sait que $x_1^L = ((x_0^L \oplus x_0^R) <<< 7) \oplus k_0$,
   On peut donc en déduire que $k_0 = x_1^L \oplus ((x_0^L \oplus x_0^R) <<< 7)$
   Dans cette égalité x_1^L, x_0^L et x_0^R sont connus, on peut donc calculer k_0

   On sait que $x_1^R = ((x_1^L \oplus x_0^R) <<< 7) \oplus k_1$.
   On peut donc en déduire que $k_1 = x_1^R \oplus ((x_1^L \oplus x_0^R) <<< 7)$
   Dans cette égalité x_1^R, x_1^L et x_0^R sont connus, on peut donc calculer k_1.
   Le programme pour trouver k_0 et k_1 pour un tour se trouve dans le fichier *q2exo2.c*

3. On sait que
   $k_0 = x_{12}^L \oplus ((x_{11}^L \oplus x_{11}^R) <<< 7)$
   $k_1 = x_{12}^R \oplus ((x_{12}^L \oplus x_{11}^R) <<< 7)$
   et
   $k_0 = x_1^L \oplus ((x_0^L \oplus x_0^R) <<< 7)$
   $k_1 = x_1^R \oplus ((x_1^L \oplus x_0^R) <<< 7)$

4.

5. En faisant 48 tours, on retrouve x_0, c'est à dire le message de départ. Donc ce n'est pas forcément en augmentant le nombre de tours qu'on renforce la sécurité.

6. Ce schéma n'est pas exactement le schéma de Feistel. Pour améliorer la sécurité, il faudrait que le schéma suive les règles de Feistel c'est à dire qu'il faudrait échanger les blocs à chaque tour.