Cryptanalyse — 4TCY902U
Responsable : G. Castagnos

TP 6 — Attaques par corrélation et algébrique

Générateur de Geffe et attaque par corrélation

Le générateur de Geffe combine les trois LFSR suivants avec la fonction booléenne $f(x_1, x_2, x_3) := x_1x_2 \oplus x_2x_3 \oplus x_3$:

- LFSR1 de longueur 13 de polynôme de rétroaction $x^{13} \oplus x^4 \oplus x^3 \oplus x \oplus 1$;
- LFSR2 de longueur 11 de polynôme de rétroaction $x^{11} \oplus x^2 \oplus 1$;
- LFSR3 de longueur 9 de polynôme de rétroaction $x^9 \oplus x^4 \oplus 1$.

À l'initialisation, trois clefs, K_1, K_2 et K_3 sont chargées dans les registres des LFSR. On note $z_1^{(t)}, z_2^{(t)}$ et $z_3^{(t)}$ les bits de sortie respectifs des trois LFSR au temps t.

Pour $t \geqslant 0$, la production de suite chiffrante $(z^{(t)})_{t\in\mathbf{N}}$ et la mise à jour de l'état interne se font dans l'ordre suivant : sortir le bit $z^{(t)} := z_1^{(t)}z_2^{(t)} \oplus z_2^{(t)}z_3^{(t)} \oplus z_3^{(t)}$ puis mise à jour de l'état des trois LFSR.

1 Écrire une fonction qui simule le générateur de Geffe. Elle doit prendre en entrée le nombre de bits à produire et les trois clefs utilisées K1, K2, K3 correspondant aux états initiaux des trois LFSR. Pour tester votre fonction, avec les clefs

$$\mathrm{K1} = [1,0,1,0,1,0,1,0,1,0,1,0,1], \quad \mathrm{K2} = [1,0,1,0,1,0,1,0,1,0,1],$$

$$\mathrm{K3} = [1,0,1,0,1,0,1,0,1]$$

les 20 premiers bits de sortie du générateur sont

$$1,0,1,0,1,0,1,0,1,1,1,0,1,0,0,0,0,1,0,1$$

2 En utilisant Sage, trouver la complexité linéaire de la suite $(z^{(t)})_{t\in\mathbf{N}}$.

3 À l'instant t, on considère $\mathrm{Z}_1 = z_1^{(t)}$, $\mathrm{Z}_2 = z_2^{(t)}$ et $\mathrm{Z}_3 = z_3^{(t)}$ comme des variables aléatoires binaires indépendantes, prenant les valeurs 0 et 1 avec équiprobabilité. On note de même $\mathrm{Z} = f(\mathrm{Z}_1, \mathrm{Z}_2, \mathrm{Z}_3)$. Montrer que $\Pr[\mathrm{Z}_1 = \mathrm{Z}] = \Pr[\mathrm{Z}_3 = \mathrm{Z}] = \frac{3}{4}$ et que $\Pr[\mathrm{Z}_2 = \mathrm{Z}] = \frac{1}{2}$.

4 Soit X une variable aléatoire binaire prenant les valeurs 0 et 1 avec équiprobabilité, indépendante de Z. Quelle est la probabilité $Pr[X = Z]$?

5 Vous avez intercepté la suite de bits $z1$ produite par le générateur de Geffe, disponible dans le fichier :

`https://www.math.u-bordeaux.fr/~gcastagn/Cryptanalyse/tp6-suiteGeffe.sage`.

Retrouvez les initialisations des trois LFSR qui ont engendré cette suite à l'aide d'une **attaque par corrélation**. Pour cela faire une recherche exhaustive sur le registre du LFSR3 en comparant la sortie produite avec la suite z en utilisant les valeurs des probabilités calculées précédemment (on pourra utiliser la fonction de corrélation, qui compte le nombre d'égalité moins le nombre de différence). Faire de même avec le LFSR1, puis finir par une recherche exhaustive sur la clef K2.

LFSR filtré et attaque algébrique

6 Écrire une fonction qui simule le LFSR de longueur 13 engendré par le polynôme de rétroaction

$$x^{13} + x^4 + x^3 + x + 1$$

et filtré par la fonction booléenne $f(x_1, x_2, x_3, x_4, x_5, x_6) = x_1x_2 + x_3x_4 + x_5x_6$ en les positions $12, 11, 6, 5, 1, 0$. Comme précédemment, on sort le bit puis on met à jour l'état du LFSR.

La fonction doit prendre en entrée la clef (c'est à dire l'état initial) et le nombre de bits à produire. Pour tester votre fonction, avec la clef $K = [1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]$, les 20 premiers bits de sortie du générateur sont

$$0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1$$

7 Vous avez intercepté la suite de bits $z2$ produite par un tel LFSR filtré, disponible dans le fichier

`https://www.math.u-bordeaux.fr/~gcastagn/Cryptanalyse/tp6-suiteFiltre.sage`

Retrouver l'initialisation du LFSR par une attaque algébrique. Pour cela, on exprime l'état initial par des variables formelles $x_0, \dots, x_{12}$ (utiliser `BPR = BooleanPolynomialRing(13,'x') ; v = BPR.gens()`). Ensuite,

(a) En déduire les 13 équations quadratiques exprimant les 13 premiers bits de suite chiffrante en fonction de $x_0, \dots, x_{12}$ (utiliser la fonction de l'exercice précédent, initialisée par la clef formelle $x_0, \dots, x_{12}$) ;

(b) Créer l'idéal engendré par ces équations puis calculer une base de Gröbner pour trouver la clef (utiliser `I = Ideal(listeEquation) ; I.groebner_basis()`).

8 Retrouver la clef par la méthode de linéarisation : créer maintenant les 100 équations exprimant les 100 premiers bits de suite chiffrante en fonction de la clef. Puis créer une matrice dont les 100 lignes correspondent aux coefficients des différents monômes de degré inférieur à 2 intervenant dans ces 100 équations, puis résoudre le système linéaire obtenu pour retrouver la clef.