TD8 - Cryptanalyse Algébrique

Responsable : M. Bombar

1 LFSR Filtré

On considère un LFSR (binaire) de longueur 13 dont le polynôme de rétroaction est

$$P(X) = 1 + X + X^3 + X^4 + X^{13},$$

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 $12, 11, 6, 5, 1, 0$.

Programmez un simulateur de ce LFSR. Vous pouvez bien évidemment vous servir des fonctions que vous aviez déjà écrites lors des TD en Septembre. Votre fonction doit prendre en entrée la clé (c'est-à-dire l'état initial) et le nombre de bits à produire. Pour tester votre fonction, avec la clé

$$K = [0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1],$$

les 20 premiers bits de sortie du générateur sont

$$[0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0],$$

2 Challenge : Cryptanalyse Algébrique d'un LFSR Filtré

3 Relations algébriques d'une SBox

3.1 Déterminer les relations algébriques

Une boîte S est par définition une fonction (non linéaire) $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$. En particulier, il existe nécessairement des relations algébriques entre l'entrée et la sortie (ne serait-ce que la forme normale algébrique). Mais on a vu qu'il existait aussi d'autres relations qui peuvent aider à modéliser le cryptosystème comme un système d'équations algébriques. Plus précisément, rajouter des équations, si elles n'ont pas trop de monômes, et qu'elles restent de bas degré, peut permettre de déterminer la variété beaucoup plus rapidement. Il est donc important de déterminer ces relations algébriques de bas degré.

Heureusement, il est relativement facile de le faire. Pour cela, on considère n variables $x_0, \ldots, x_{n-1}$ correspondant aux bits de l'entrée, et m variables $y_0, \ldots, y_{m-1}$ correspondant

à la sortie et on définit l'anneau $\mathbb{F}_2[x_0, \ldots, x_{n-1}, y_0, \ldots, y_{m-1}]$ à $m+n$ variables de sorte que $S(x_0, \ldots, x_{n-1}) = (y_0, \ldots, y_{m-1})$. Ensuite, on écrit une grosse matrice dont les colonnes sont indéxées par $\mathbb{F}_2^n$ (ou de manière équivalente par les entiers de 0 à $2^n - 1$), donc 2^n colonnes, et dont les lignes correspondent à tous les monômes de degré borné (par disons 2 si on ne souhaite que des relations quadratiques) : $\sum_{i=0}^{d} \binom{n+m}{i}$ lignes. L'entrée correspondant à un entier a (colonne) et à un monôme $\mu = \prod_i x_i^{\alpha_i} \prod_j y^{\alpha_j}$ (ligne) est simplement le bit obtenu en remplaçant les x_i par les bits correspondants dans a, et les y_j par les bits correspondant dans $S(a)$.

Pour mieux comprendre, testons sur un exemple et considérons la boîte S définie par

x	0	1	2	3	4	5	6	7
$S(x)$	7	6	0	4	2	5	1	3

De sorte que par exemple $S(1) = 6$, *i.e.*, pour $\mathbf{x} = (x_0, x_1, x_2) = (0, 0, 1)$ on a $\mathbf{y} = (y_0, y_1, y_2) = (1, 1, 0)$. Il y a 1 monôme de degré 0, 6 monômes de degré 1, et 15 monômes de degré 2 (en prenant en compte que $x_i^2 = x_i$ et $y_i^2 = y_i$. On peut alors écrire la grosse matrice suivante :

0	1	2	3	4	5	6	7	
1	1	1	1	1	1	1	1	1
–	–	–	–	1	1	1	1	$x0$
–	–	1	1	–	–	1	1	$x1$
–	1	–	1	–	1	–	1	$x2$
1	1	–	1	–	1	–	–	$y0$
1	1	–	–	1	–	–	1	$y1$
1	–	–	–	–	1	1	1	$y2$
–	–	–	–	–	–	1	1	$x0 * x1$
–	–	–	–	–	1	–	1	$x0 * x2$
–	–	–	–	–	1	–	–	$x0 * y0$
–	–	–	–	1	–	–	1	$x0 * y1$
–	–	–	–	–	1	1	1	$x0 * y2$
–	–	–	1	–	–	–	1	$x1 * x2$
–	–	–	1	–	–	–	–	$x1 * y0$
–	–	–	–	–	–	–	1	$x1 * y1$
–	–	–	–	–	–	1	1	$x1 * y2$
–	1	–	1	–	1	–	–	$x2 * y0$
–	1	–	–	–	–	–	1	$x2 * y1$
–	–	–	–	–	1	–	1	$x2 * y2$
1	1	–	–	–	–	–	–	$y0 * y1$
1	–	–	–	–	1	–	–	$y0 * y2$
1	–	–	–	–	–	–	1	$y1 * y2$

Et une simple élimination Gaussienne permet de déterminer les relations algébriques : Elles correspondent aux lignes de 0 :

0	1	2	3	4	5	6	7	
1	–	–	–	–	–	–	–	$x0*y0+x1+x2+y0+y1+1$
–	1	–	–	–	–	–	–	$x0*y0+x0+x1+y2+1$
–	–	1	–	–	–	–	–	$x0*y0+x0+y0+1$
–	–	–	1	–	–	–	–	$x0*y0+x0+x2+y1+y2$
–	–	–	–	1	–	–	–	$x0*y0+x0+x1+x2+y0+y1+y2+1$
–	–	–	–	–	1	–	–	$x0*y0$
–	–	–	–	–	–	1	–	$x0*y0+x2+y0+y2$
–	–	–	–	–	–	–	1	$x0*y0+x1+y1+1$
–	–	–	–	–	–	–	–	$x0*x2+x1+y1+1$
–	–	–	–	–	–	–	–	$x0*x1+x1+x2+y0+y1+y2+1$
–	–	–	–	–	–	–	–	$x0*y1+x0+x2+y0+y2$
–	–	–	–	–	–	–	–	$x0*y0+x0*y2+x1+x2+y0+y1+y2+1$
–	–	–	–	–	–	–	–	$x1*x2+x0+x1+x2+y2+1$
–	–	–	–	–	–	–	–	$x0*y0+x1*y0+x0+x2+y1+y2$
–	–	–	–	–	–	–	–	$x0*y0+x1*y1+x1+y1+1$
–	–	–	–	–	–	–	–	$x1*y2+x1+x2+y0+y1+y2+1$
–	–	–	–	–	–	–	–	$x0*y0+x2*y0+x1+x2+y1+1$
–	–	–	–	–	–	–	–	$x2*y1+x0+y1+y2$
–	–	–	–	–	–	–	–	$x2*y2+x1+y1+1$
–	–	–	–	–	–	–	–	$y0*y1+x0+x2+y0+y1+y2$
–	–	–	–	–	–	–	–	$y0*y2+x1+x2+y0+y1+1$
–	–	–	–	–	–	–	–	$y1*y2+x2+y0$

(Q1) Déterminer les relations algébriques de degré 2 pour la boîte S

x	0	1	2	3	4	5	6	7
$S(x)$	4	1	3	6	5	7	2	0

(Q2) Construire avec Sage l'idéal des relations algébriques de S et en déterminer une base de Gröbner.

Rappel : Pour construire l'anneau des polynômes multivariés on peut faire comme dans l'exemple suivant. Notez que l'ordre monomial considéré par défaut n'est pas l'ordre lexicographique.

```
sage: R.<x0, x1, x2, y0, y1, y2> = PolynomialRing(GF(2))
sage: R Multivariate Polynomial Ring in x0, x1, x2, y0, y1, y2
over Finite Field of size 2
```

```
sage: R.term_order()
Degree reverse lexicographic term order
sage: f = x0*y0 + x1*y0 + x0 + x2 + y1 + y2
sage: g = x0*y0 + x1*y1 + x1 + y1 + 1
sage: I = Ideal([f, g])
sage: I
Ideal (x0*y0 + x1*y0 + x0 + x2 + y1 + y2, x0*y0 + x1*y1 + x1 +
    y1 + 1) of Multivariate Polynomial Ring in x0, x1, x2, y0,
    y1, y2 over Finite Field of size 2
```

3.2 Utiliser Sage

Certaines et certains d'entre vous ont déjà expérimenté avec ça, mais en Sage il existe déjà une classe `SBox` que vous pouvez utiliser en important

```
sage: from sage.crypto.sbox import SBox
```

Je vous invite à aller lire la documentation https://doc.sagemath.org/html/en/reference/cryptography/sage/crypto/sbox.html et à expérimenter avec. En particulier, il y a une implémentation des tables DDT et LAT qu'on a vues pour les cryptanalyses différentielles et linéaires, mais bien d'autre choses encore. Sage permet là encore de retrouver les relations algébriques directement. Pour l'exemple précédent, ça donne ça :

```
sage: S = SBox(7, 6, 0, 4, 2, 5, 1, 3)
sage: S.polynomials()
[x0*x2 + x1 + y1 + 1,
x0*x1 + x1 + x2 + y0 + y1 + y2 + 1,
x0*y1 + x0 + x2 + y0 + y2,
x0*y0 + x0*y2 + x1 + x2 + y0 + y1 + y2 + 1,
x1*x2 + x0 + x1 + x2 + y2 + 1,
x0*y0 + x1*y0 + x0 + x2 + y1 + y2,
x0*y0 + x1*y1 + x1 + y1 + 1,
x1*y2 + x1 + x2 + y0 + y1 + y2 + 1,
x0*y0 + x2*y0 + x1 + x2 + y1 + 1,
x2*y1 + x0 + y1 + y2,
x2*y2 + x1 + y1 + 1,
y0*y1 + x0 + x2 + y0 + y1 + y2,
y0*y2 + x1 + x2 + y0 + y1 + 1,
y1*y2 + x2 + y0]
```

On retrouve bien les 14 relations polynomiales correspondant aux lignes nulles de la matrice échelonnée.

4 Modélisation Algébrique de la Factorisation d'Entiers RSA

On considère $N = PQ$ un module RSA (connu) avec P et Q deux nombres premiers (inconnu).

(Q3) Modélisez ce problème sous la forme d'une unique équation polynomiale dont les inconnues sont les bits de P et de Q.

(Q4) Écrivez l'équation dans le cas $N = 35$.

(Q5) Traduire cette unique équation sous la forme d'un système d'équations polynomiales de degré 4 sur $\mathbb{F}_2$. Pensez à introduire des inconnues auxiliaires pour prendre en compte les retenues.

Indication : On note p_0, p_1 les deux bits de poids faibles de P. Considérez alors la somme partielle $S = p_0Q + 2p_1Q$.

(Q6) Généralisez cette approche pour factoriser un module RSA N quelconque. Combien d'équation possède ce système ? Quel est son degré ?

5 Bases de Gröbner

— On considère un anneau $K[x_1, \ldots, x_n]$ où K est un corps. On pourra éventuellement supposer $K = \mathbb{F}_q$ est un corps fini (et même $q = 2$ si ça vous arrange).

— Les variables seront toujours supposées ordonnées de la façon suivante

$$x_1 > x_2 > \cdots > x_n$$

indépendamment de l'ordre monomial choisi (chaque ordre monomial défini dans le cours est en réalité défini à l'ordre près des variables. Il y a donc en réalité $n!$ ordres monomiaux pour chaque famille présentée en cours).

— On utilise la notation standard pour les monômes $\mathbf{x}^{\boldsymbol{\alpha}} \stackrel{\text{def}}{=} x_1^{\alpha_1} \cdot x_n^{\alpha_n}$.

— Pour $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$, on note $|\boldsymbol{\alpha}| \stackrel{\text{def}}{=} \sum_{i=1}^{n} \alpha_i$ le degré du monôme correspondant.

5.1 Ordres monomiaux

On rappelle que l'ordre **lexicographique** (LEX) est défini par

$$\mathbf{x}^{\boldsymbol{\alpha}} <_{lex} \mathbf{x}^{\boldsymbol{\beta}} \iff \text{La première coordonnée non nulle de } \boldsymbol{\alpha} - \boldsymbol{\beta} \text{ est négative.}$$

et l'ordre **degré-lexicographique inverse** (DRL)

$$\mathbf{x}^{\boldsymbol{\alpha}} <_{DRL} \mathbf{x}^{\boldsymbol{\beta}} \iff \begin{cases} |\boldsymbol{\alpha}| < |\boldsymbol{\beta}| \text{ ou} \\ |\boldsymbol{\alpha}| = |\boldsymbol{\beta}| \text{ et } \boldsymbol{\alpha} >_{lex} \boldsymbol{\beta}. \end{cases}$$

(Q7) Vérifiez que LEX et DRL sont bien des ordres monomiaux.

(Q8) On définit la relation d'ordre suivante sur les monômes :

$$\mathbf{x}^{\boldsymbol{\alpha}} > \mathbf{x}^{\boldsymbol{\beta}} \iff \exists i_0 \text{ tel que } \begin{cases} \forall i > i_0,\ \alpha_i = \beta_i \\ \alpha_{i_0} < \beta_{i_0}. \end{cases}$$

Montrez que ceci ne définit pas un ordre monomial valide.
Indice : Traitez le cas d'une seule variable ($n = 1$).

5.2 Bases de Gröbner

(Q9) On considère l'idéal $\mathcal{I} = \langle f_1, f_2 \rangle$ engendré par les polynômes

$$f_1(x, y) = x^2y + 1 \qquad f_2(x, y) = xy^2 - 2.$$

Montrez que le polynôme $f = y + 2x \in \mathcal{I}$.

(Q10) En déduire que f_1, f_2 **n'est pas** une base de Gröbner, et ce, **quel que soit** l'ordre monomial choisi.

(Q11) Déterminez avec Sage une base de Gröbner de $\mathcal{I}$ par rapport à $<_{lex}$ et à $<_{DRL}$.

(Q12) Prouver le lemme du cours :

Lemme

Soit $\mathcal{J} = \langle \mathbf{x}^{\boldsymbol{\alpha}(1)}, \ldots, \mathbf{x}^{\boldsymbol{\alpha}(k)} \rangle$ un idéal **monomial**, et soit $\mathbf{x}^{\boldsymbol{\beta}} \in K[x_1, \ldots, x_n]$ un monôme. Alors

$$\mathbf{x}^{\boldsymbol{\beta}} \in \mathcal{J} \iff \exists \ell,\ \mathbf{x}^{\alpha(\ell)} \text{ divise } \mathbf{x}^{\boldsymbol{\beta}}, \quad i.e.,\ \boldsymbol{\beta} - \boldsymbol{\alpha}(\ell) \geq 0.$$