Cryptanalyse

Cours 3 - Chiffrements par Flots: Cryptanalyse et Contre-Mesures

Maxime Bombar

Mardi 16 Septembre 2024

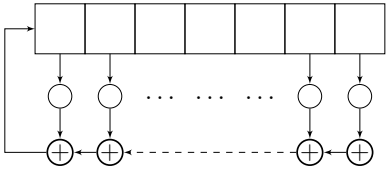
Introduction

Objectifs du Jour

- LFSR Combinés et Filtrés
- Attaques probabilistes
- Exemples de Design

Rappels de la semaine dernière

LFSR

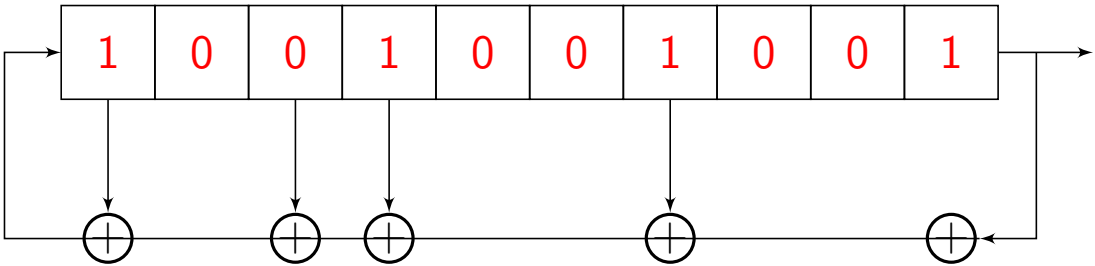


- Une suite est produite par un LFSR ssi elle est **récurrente linéaire** de la forme $\mathbf{s_{n+\ell} = c_1 s_{n+\ell-1} + \cdots + c_\ell s_0}$.
- Un LFSR est uniquement déterminé par son état à un instant t, et son **polynome de rétroaction** $P(X) = 1 - \sum_{i=1}^{\ell} c_i X^i$.
- La suite est périodique, de période $\leqslant q^{\deg(P)} - 1$, avec égalité si par exemple P est **primitif**.

Complexité linéaire d'une suite

- Le polynôme de rétroaction d'une suite récurrente linéaire n'est pas unique.
- Parmi tous les polynômes de rétroaction, il en existe un de **degré minimal**.
- Le degré $\Lambda(\mathbf{s})$ du polynôme minimal d'une suite récurrente linéaire $\mathbf{s} \stackrel{\text{def}}{=} (s_n)_{n\in\mathbb{N}}$ est appelé **complexité linéaire de la suite**.
- L'**algorithme de Berlekamp-Massey** permet de retrouver $\Lambda(\mathbf{s})$ et le polynôme minimal en ayant accès à $2\Lambda(\mathbf{s})$ termes, en temps $O(\Lambda(\mathbf{s})^2)$.

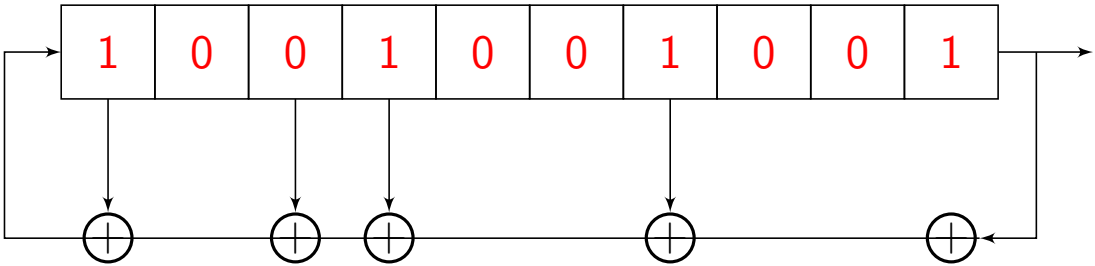
Polynôme de rétroaction



Exercice (2 min)

Quel est le polynôme de rétroaction de la suite binaire définie par ce LFSR ?

Polynôme de rétroaction



Réponse

$$P(X) = 1 + X + X^3 + X^4 + X^7 + X^{10}$$

Calculons le polynôme minimal : Méthode 1 - Sage direct

- $P(X) = 1 + X + X^3 + X^4 + X^7 + X^{10}$
- $s = (1, 0, 0, 1, 0, 0, 1, 0, 0, 1, \dots)$

Rappel : $S(X) \stackrel{\text{def}}{=} \sum_{j=0}^{\infty} s_j X^j = \dfrac{Q(X)}{P(X)}$ avec $\deg(Q) < \deg(P)$.

- On calcule les premiers termes de suite et on construit la série formelle $S(X)$ à un $O(\cdot)$ près.
- On sait que $P(X) \cdot S(X)$ (vu comme série formelle) est en fait un polynôme $Q(X)$
- On simplifie la fraction rationnelle $\dfrac{Q(X)}{P(X)}$

Démonstration

Calculons le polynôme minimal : Méthode 2 - À la main

- $P(X) = 1 + X + X^3 + X^4 + X^7 + X^{10}$
- $s = (1, 0, 0, 1, 0, 0, 1, 0, 0, 1, \dots)$

$$
\begin{aligned}
S(X) \cdot P(X) &= \left(\sum_{j=0}^{\infty} s_j X^j \right) \left(\sum_{n=0}^{9} p_j X^j \right) \\
&= \sum_{j=0}^{\infty} X^j \left(\sum_{n=0}^{j} p_n \mathbb{1}_{n \leqslant 9} \cdot s_{j-n} \right) \\
&= \underbrace{\sum_{j=0}^{9} X^j \left(\sum_{n=0}^{j} p_n \cdot s_{j-n} \right)}_{=Q(X)} + \underbrace{\sum_{j=10}^{\infty} X^j \left(\sum_{n=0}^{9} p_n \cdot s_{j-n} \right)}_{=0}
\end{aligned}
$$

Calculons le polynôme minimal : Méthode 2 - À la main

- $P(X) = 1 + X + X^3 + X^4 + X^7 + X^{10}$
- $s = (1, 0, 0, 1, 0, 0, 1, 0, 0, 1, \dots)$

- On a donc ici
$$Q(X) = 1 + X + X^7$$
- On cherche (P_0, Q_0) avec $\deg(Q_0) < \deg(P_0) \leqslant \deg(P)$ et tel que
$$R(X) = \frac{Q_0(X)}{P_0(X)} = \frac{Q(X)}{P(X)} = \frac{1 + X + X^7}{1 + X + X^3 + X^4 + X^7 + X^{10}}$$
- On voit que $P(X) = (1 + X + X^7) \cdot (1 + X^3)$ donc $R(X) = \frac{1}{1 + X^3}$.
- On en déduit que $Q_0(X) = 1$ et $P_0(X) = 1 + X^3$.

Calculons le polynôme minimal : Méthode 3 - Sage + la preuve

- $P(X) = 1 + X + X^3 + X^4 + X^7 + X^{10}$
- $s = (1, 0, 0, 1, 0, 0, 1, 0, 0, 1, \dots)$

Par la preuve, on a l'expression explicite de $Q(X)$:

$$Q(X) = \sum_{j=0}^{\deg(P)-1} X^j \left(\sum_{n=0}^{j} p_n \cdot s_{j-n} \right)$$

- On définit $Q(X)$ explicitement. On a juste besoin de $(s_j)_{0 \leqslant j \leqslant 9}$ (*i.e.* l'état initial).
- On simplifie $\dfrac{Q(X)}{P(X)}$ avec Sage.

Démonstration

Calculons le polynôme minimal : Méthode 4 - Berlekamp-Massey

Rappel

L'algorithme de Berlekamp-Massey a besoin de $2\Lambda(s)$ termes de la suite, et retourne $\Lambda(s)$ ainsi que le polynôme de rétroaction minimal.

Remarque : Utile si on ne connaît pas un polynôme de rétroaction (*e.g.* cryptanalyse)

- On calcule quelques termes de la suite.
- On lance Berlekamp-Massey.

Démonstration

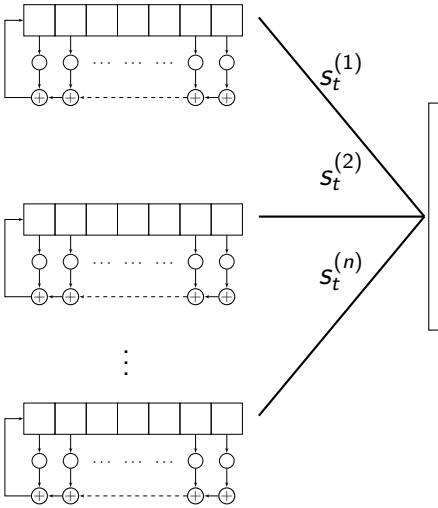
LFSR Combinés et Filtrés

Le Constat

- Si la complexité linéaire est **trop faible**, Berlekamp-Massey retourne le résultat immédiatement.
- Mais la complexité linéaire est directement reliée au **degré du polynôme minimal**, *i.e.* à la taille du registre interne.
- Il faudrait donc avoir des LFSR de taille **gigantesque**.

Notre but : **augmenter** artificiellement la complexité linéaire en gardant un cryptosystème de taille raisonnable.

Première méthode : LFSR combinés



$\longrightarrow s_t = f(s_t^{(1)}, \ldots, s_t^{(n)})$ où $f : \mathbb{F}_2^n \to \mathbb{F}_2$.

Nos hypothèses de travail

Chaque LFSR a un polynôme de rétroaction **primitif** pour avoir de bonnes propriétés statistiques.

Les caractéristiques (longueur, polynômes de rétroaction, ...) sont **publiques**.

Les seules données **secrètes** sont les états initiaux de chaque LFSR (obtenus via la clé secrète, et les différents IV).

Interlude : Fonction Booléenne

Definition (Définition)

On appelle **fonction booléenne** à n variables une fonction $f : \mathbb{F}_2^n \to \mathbb{F}_2$. Elle peut-être décrite par sa **table de vérité** qui donne l'image de tous les éléments de $\mathbb{F}_2^n$.

Exercice

Combien y a-t-il de fonctions booléennes à n variables ?

Support et Poids

Définition

- Le **support** d'une fonction booléenne $f : \mathbb{F}_2^n \to \mathbb{F}_2$ est

$$Supp(f) \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathbb{F}_2^n \mid f(\mathbf{x}) \neq 0\}.$$

- Le **poids** de f est $w(f) \stackrel{\text{def}}{=} |Supp(f)|$.
- f est dite **équilibrée** si $w(f) = 2^{n-1}$. Dans ce cas,

$$|Supp(f)| = |\mathbb{F}_2^n \setminus Supp(f)|.$$

Afin de conserver les propriétés statistiques des LFSR, on demande à ce que la fonction booléenne soit équilibrée.

Rappel : Pour toute fonction $f : \mathbb{F}_q \to \mathbb{F}_q$, il existe un polynôme P tel que $f(x) = P(x)$ pour tout $x \in \mathbb{F}_q$.

Pourquoi ?

Fonctions, corps finis et polynômes

Rappel : Pour toute fonction $f : \mathbb{F}_q \to \mathbb{F}_q$, il existe un polynôme P tel que $f(x) = P(x)$ pour tout $x \in \mathbb{F}_q$.

Soit $x_1, \ldots, x_q$ une énumération de $\mathbb{F}_q$ et soit $y_1, \ldots, y_q$ leurs images par f. Par hypothèse, les x_i sont tous distincts, donc par **interpolation de Lagrange**, il existe un unique polynôme P de degré au plus $q-1$ tel que $P(x_i) = y_i = f(x_i)$, à savoir :

$$P(X) = \sum_{i=1}^{q} \prod_{j \neq i} \frac{X - x_j}{x_i - x_j} f(x_i)$$

Remarque : En réalité on a même $P(X) \in \mathbb{F}_q[X]/(X^q - X)$.

Forme Normale Algébrique

- Par récurrence sur n, toute fonction $f : \mathbb{F}_q^n \to \mathbb{F}_q$ coïncide avec un unique polynôme $P(X_1, \ldots, X_n) \in \mathbb{F}_q[X_1, \ldots, X_n]/(X_1^q - 1, \ldots, X_n^q - 1)$, qui est appelé **forme normale algébrique** de f.
- Le **degré** d'une telle fonction est le degré de sa forme normale.
- Dans le cas des fonctions booléennes, on peut noter

$$f(X_1, \ldots, X_n) = \sum_{\mathbf{u} \in \mathbb{F}_2^n} a_{\mathbf{u}} X_1^{u_1} \cdots X_n^{u_n}$$

et les coefficients s'obtiennent par

$$a_{\mathbf{u}} = \sum_{\mathbf{x} \preccurlyeq \mathbf{u}} f(x) \qquad \text{où } \mathbf{x} \preccurlyeq \mathbf{y} \text{ ssi } x_i \leqslant y_i \text{ pour tout } 1 \leqslant i \leqslant n.$$

Degré algébrique

- On appelle **degré algébrique** (ou tout simplement **degré**) d'une fonction booléenne, son degré en tant que polynôme multivarié.
- C'est grâce à des fonctions Booléennes de degré > 1 qu'on va pouvoir augmenter la complexité linéaire des LFSR combinés.

Si

$$f(X_1, \ldots, X_n) = \sum_{\mathbf{u} \in \mathbb{F}_2^n} a_{\mathbf{u}} X_1^{u_1} \cdots X_n^{u_n}$$

alors son degré est le **nombre maximal de variables dans un monôme**.

Exemple

Soit $f : \mathbb{F}_2^3 \to \mathbb{F}_2$ la fonction booléenne définie par la table de vérité suivante :

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
$f(x_1, x_2, x_3)$	0	1	0	0	0	1	1	1

Vérifiez que la forme normale algébrique de f est

$$f(X_1, X_2, X_3) = X_3 + X_2X_3 + X_1X_2 \quad \text{et donc} \quad \deg(f) = 2.$$

Remarque : c'est un IF !

- Si $x_2 = 0$, alors $f(x_1, x_2, x_3) = x_3$
- Sinon, $f(x_1, x_2, x_3) = x_1$.

Complexité linéaire de la suite combinée

Lemme (Rueppel, Staffelbach 1987)

Soient $(s^{(1)})$ et $(s^{(2)})$ deux suites récurrentes linéaires, de polynômes de rétroaction minimaux respectifs $P^{(1)}$ et $P^{(2)}$. Alors

- $\Lambda(\mathbf{s}^{(1)} + \mathbf{s}^{(2)}) \leqslant \Lambda(\mathbf{s}^{(1)}) + \Lambda(\mathbf{s}^{(2)})$, avec égalité ssi $\mathrm{pgcd}(P^{(1)}, P^{(2)}) = 1$.
- $\Lambda(\mathbf{s}^{(1)} \star \mathbf{s}^{(2)}) \leqslant \Lambda(\mathbf{s}^{(1)})\Lambda(\mathbf{s}^{(2)})$, où $\star$ désigne le produit terme à terme.

  Si de plus $P^{(1)}$ et $P^{(2)}$ sont **primitifs**, de degrés **distincts** et supérieurs à 2, alors il y a **égalité**.

Complexité linéaire de la suite combinée

Corollaire

Soient $(\mathbf{s}^{(1)}), \ldots, (\mathbf{s}^{(n)})$ des suites récurrentes linéaires produites par des LFSR minimaux de longueurs respectives $\ell^{(1)}, \ldots, \ell^{(n)}$. Soit $f : \mathbb{F}_2^n \to \mathbb{F}_2$ une fonction booléenne. Alors, la suite combinée $f(\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(n)})$ a pour complexité linéaire

$$\Lambda = f(\ell^{(1)}, \ldots, \ell^{(n)})$$

obtenue en évaluant la forme normale algébrique de f vue comme un polynôme dans $\mathbb{Z}$.

Exercice

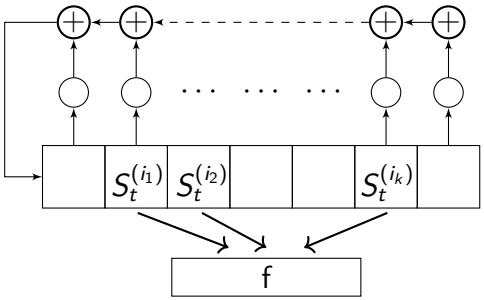
Prouver ce résultat pour le cas de la somme.

Exemple

- Le chiffrement de Geffe (1973) utilise 3 LFSR de longueurs $\ell^{(1)}, \ell^{(2)}, \ell^{(3)}$ deux à deux premières entre elles.
- Ils sont combinés par la fonction booléenne $f(x_1, x_2, x_3) = x_3 + x_2x_3 + x_1x_2$.
- La suite produite a alors une complexité linéaire

$$\Lambda(s) = \ell^{(3)} + \ell^2\ell^{(3)} + \ell^{(1)}\ell^{(2)}$$

Méthode 2 : LFSR filtrés



Revient à combiner k LFSR avec le **même polynôme de rétroaction** mais des états initiaux décalés.

Attention : les résultats précédents sur les complexités linéaires ne s'appliquent pas.

Complexité linéaire d'un LFSR filtré

(Edwin L. Key, 1976)

La complexité linéaire $\Lambda(\mathbf{s})$ d'une suite chiffrante $\mathbf{s}$ produite par un LFSR de longueur ℓ et filtré par une fonction booléenne de **degré algébrique** d vérifie

$$\Lambda(\mathbf{s}) \leqslant \sum_{i=0}^{d} \binom{\ell}{i}.$$

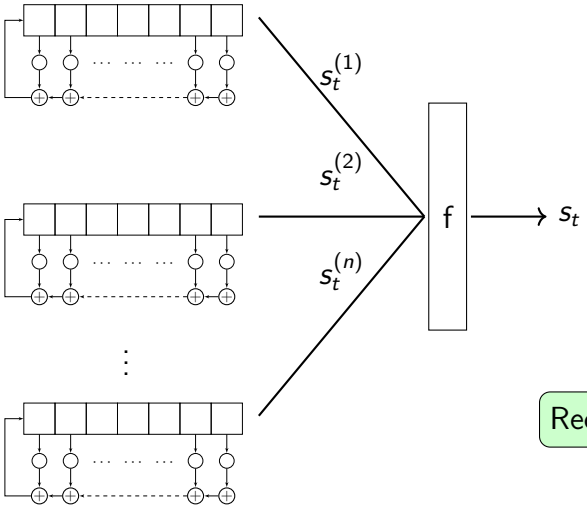
(Rueppel, 1986)

Lorsque ℓ est premier, assez grand, alors $\Lambda(s) \approx \binom{\ell}{d}$ pour la plupart des fonctions booléennes de degré d.

En pratique, c'est souvent le cas.

Quelques Cryptanalyses

Attaque par Corrélations

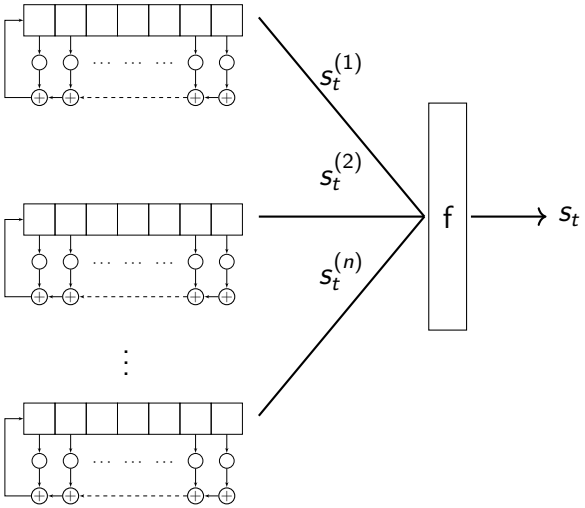


On considère n LFSR de longueur $\ell_1, \ldots, \ell_n$.

But : Retrouver les états internes.

Recherche exhaustive $= \prod_{i=1}^{n}(2^{\ell_i} - 1)$

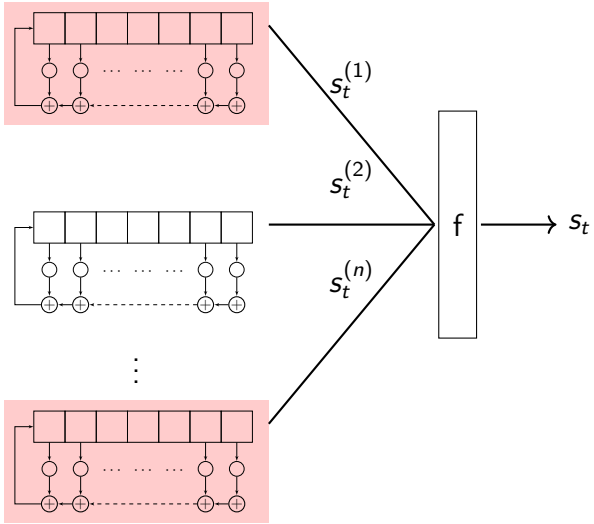
Attaque par Corrélations



Si f est **mal choisie**, alors la suite s_t peut être corrélée à une suite formée par une combinaison de **moins** de LFSR.

Par exemple, $s_t = s_t^{(1)} + s_t^{(n)}$ avec une grosse probabilité.

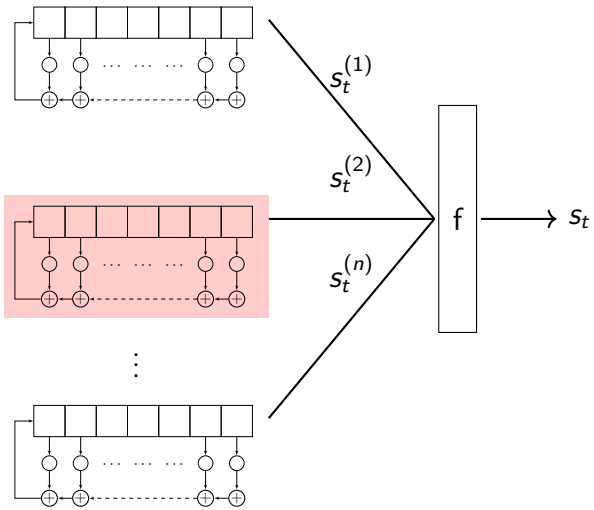
Attaque par Corrélations



Principe :

- Recherche exhaustive sur les états internes de ces LFSR.
- Vérifier si la corrélation est observée.
- Lorsqu'on obtient un des états internes, continuer avec les autres.

Attaque par Corrélations



Principe :

- Recherche exhaustive sur les états internes de ces LFSR.
- Vérifier si la corrélation est observée.
- Lorsqu'on obtient un des états internes, continuer avec les autres.

$$e.g.\ \prod_{i=1}^{n}\left(2^{\ell_i}-1\right) \longrightarrow \sum_{i=1}^{n}\left(2^{\ell_i}-1\right)$$

Contre-Mesures

- Une fonction booléenne $f : \mathbb{F}_2^n \to \mathbb{F}_2$ est **non-corrélée** à l'ordre k si pour toutes variables aléatoires binaires, et indépendantes $X_1, \ldots, X_n$, la variables aléatoire $f(X_1, \ldots, X_n)$ est **indépendante** de n'importe quel $(X_{i_1}, \ldots, X_{i_k})$.
- Le plus grand tel k est l'**immunité** de f aux corrélations.

Idée : On veut choisir f avec une **forte immunité**, et un **degré algébrique** élevé.

Il y a 2^{2^n} fonctions booléennes à n variables
$\rightarrow$ Les bonnes fonctions booléennes sont difficiles à trouver.

Attaques algébriques

Clairs et chiffrés reliés par des relations algébriques.

Si le système est trop simple, on peut parfois le résoudre (linéarisation, bases de Gröbner...). En pratique, plus efficaces pour les LFSR filtrés.

On aura l'occasion d'en reparler.