LFSR

1 Le registre à décalage à rétroaction linéaire

Définition 1.1. *Un* registre à décalage à rétroaction linéaire (LFSR : Linear Feedback Shift Register) *binaire de longueur L est composé d'un registre à décalage contenant une suite de L bits $(s_i, \ldots, s_{i+L-1})$ et d'une fonction de rétroaction linéaire.*

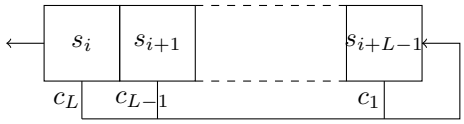

FIGURE 1 – Un LFSR de longueur L à l'instant i

Le fonctionnement d'un LFSR binaire de longueur L est le suivant : à chaque top d'horloge, le bit de « gauche » s_i constitue la sortie du registre, et les autres sont décalés vers la gauche ; le nouveau bit s_{i+L} placé dans la cellule de « droite » du registre est donné par une fonction linéaire :

$$s_{i+L} = c_1 s_{i+L-1} + c_2 s_{i+L-2} + \cdots + c_{L-1} s_{i+1} + c_L s_i \tag{1}$$

où les coefficients c_i sont binaires.

Définition 1.2. *Les bits $(s_0, \ldots, s_{L-1})$ qui déterminent entièrement la suite produite constituent* l'état initial du registre.

La suite $(s_n)_{n\geq 0}$ produite par un LFSR de longueur L est donc une suite à récurrence linéaire homogène d'ordre L. Inversement, ce type de suite peut toujours être produite par un LFSR. On peut remarquer qu'une telle suite est ultimement périodique, *i.e.* il existe une pré-période n_0 telle que la suite $(s_n)_{n\geq n_0}$ est périodique.

Théorème 1.1. *Toute suite binaire à récurrence linéaire homogène d'ordre L est ultimement périodique, et sa plus petite période T est inférieure ou égale à $2^L - 1$. De plus, si le coefficient c_L est non nul, alors la suite est périodique.*

Je vais maintenant m'intéresser à la *complexité linéaire* des LFSRs. Pour ce faire, associons à la suite $(s_n)_{n\geq 0}$ la série géneratrice $s(X) = \sum_{n\geq 0} s_n X^n$. Cette approche a été introduite par N. Zierler en 1959 et fait intervenir le *polynôme de rétroaction du registre.*

Définition 1.3. *Soit un LFSR dont la fonction de rétroaction est donnée par la relation (1). Son* polynôme de rétroaction *f est le polynôme de $\mathbb{F}_2[X]$*

$$f(X) = 1 + c_1 X + c_2 X^2 + \ldots + c_L X^L.$$

On peut alors écrire le développement en série formelle de la suite $(s_n)_{n\geq 0}$ sous forme d'une fraction rationnelle. Dans la suite, je noterai $deg(f)$ le degré d'un polynôme $f \in \mathbb{F}_2[X]$.

Théorème 1.2. *La suite $(s_n)_{n\geq 0}$ est produite par un LFSR dont le polynôme de rétroaction est $f(X) = 1 + c_1 X + \ldots + c_L X^L$ si et seulement si son développement en série formelle $s(X) = \sum_{n=0}^{\infty} s_n X^n$ s'écrit*

$$s(X) = \frac{g(X)}{f(X)}$$

où g est un polynôme de $\mathbb{F}_2[X]$ tel que $deg(g) < deg(f)$. En outre, le polynôme g est entièrement déterminé par l'état initial du registre :

$$g(X) = \sum_{i=0}^{L-1} X^i \sum_{j=0}^{i} c_{i-j} s_j.$$

Afin d'obtenir une forme canonique de la série génératrice de $(s_n)_{n\geq 0}$, on définit le *polynôme de rétroaction minimal de la suite (ou du registre)* : c'est un diviseur de $f(X)$, qui de plus est le polynôme de plus bas degré parmi les polynômes de rétroaction de tous les LFSRs possibles qui génèrent la suite $(s_n)_{n\geq 0}$.

Définition 1.4. *Soit $(s_n)_{n\geq 0}$ une suite binaire à rétroaction linéaire d'ordre L dont l'état initial est non nul. Son* polynôme de rétroaction minimal *est l'unique polynôme unitaire f_0 de $\mathbb{F}_2[X]$ tel qu'il existe $g_0 \in \mathbb{F}_2[X]$, avec $deg(g_0) < deg(f_0)$ et $pgcd(g_0, f_0) = 1$, vérifiant*

$$s(X) = \frac{g_0(X)}{f_0(X)}.$$

La complexité linéaire *du LFSR produisant la suite $(s_n)_{n\geq 0}$, notée $\Lambda(s)$, est alors égale au degré de f_0.*

En clair, la complexité linéaire d'un LFSR produisant une suite $(s_n)_{n\geq 0}$ est la longueur du plus petit LFSR permettant d'engendrer $(s_n)_{n\geq 0}$.

Exemple 1-1. Soit $(s_n)_{n\geq 0}$ la suite produite par le LFSR de longueur 10, de polynôme de rétroaction

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

et qui commence par 1001001001. Soit g le polynôme déterminé par l'état initial du registre :

$$g(X) = \sum_{n=0}^{9} X^n \sum_{i=0}^{n} c_{n-i} s_i = X^7 + X + 1.$$

La série génératrice de $(s_n)_{n\geq 0}$ est :

$$s(X) = \sum_{n=0}^{\infty} s_n X^n = \frac{g(X)}{f(X)} = \frac{X^7 + X + 1}{X^{10} + X^7 + X^4 + X^3 + X + 1} = \frac{1}{X^3 + 1}.$$

On a donc $f_0(X) = X^3 + 1$, et $(s_n)_{n\geq 0}$ peut être générée par un LFSR de longueur 3. La complexité linéaire du LFSR est donc 3.

2 Du choix du polynôme de rétroaction

Il est clair, au vu de ce qui précède, que si l'on a pris soin d'utiliser un polynôme de rétroaction irréductible, alors la fraction ne pourra être réduite, et l'on est certains de ne pas pouvoir générer la même suite avec un registre plus court !

Voyons maintenant comment influer sur la période de la suite produite.

Définition 2.1. *Soit f un polynôme de $\mathbb{F}_2[X]$. Son* ordre*, noté $ord(f)$, est le plus petit entier t tel que $X^t \equiv 1 \mod f(X)$.*

Théorème 2.1. *Soit $(s_n)_{n\geq 0}$ une suite binaire à récurrence linéaire de polynôme de rétroaction minimal f_0, et dont l'état initial est non nul. Alors sa plus petite période est égale à l'ordre de f_0.*

On peut donc contrôler la période de la suite produite en contrôlant l'ordre du polynôme de rétroaction.

Définition 2.2. *Soit $f(X)$ un polynôme irréductible de $\mathbb{F}_2[X]$, de degré L. Il est dit* primitif *s'il est d'ordre $2^L - 1$.*

Ainsi, si l'on veut construire un LFSR optimal (au regard de la période de la suite produite) de longueur L, il faut s'assurer que le polynôme de rétroaction choisi est de degré L et primitif. On sera alors assuré d'obtenir une période maximale, pour peu que l'on prenne un état initial non nul.

Un autre intérêt des polynômes de rétroaction primitifs est la qualité statistique des suites produites.

Théorème 2.2. *Considérons la suite produite par un LFSR de longueur L. Si le poltnôme de rétroaction f est de degré n et primitif, et si l'état intitial est non nul, alors :*

- *toutes les suites binaires de longueur $\ell < L$ se retrouvent avec la même fréquence dans la suite produite : les sous-suites non nulles apparaissent chacune $2^{L-\ell} - 1$ fois, et les sous-suites nulle apparaissent chacune $2^{L-\ell}$ fois ;*
- *toutes les suites binaires non nulles de longueur n apparaissent également avec la même fréquence.*

Ainsi, les suites produites par des LFSR utilisant des polynômes de rétroaction primitifs sont particulièrement intéressantes. Elles sont appelées *suites ML (Maximum Length Sequences)*. On peut noter qu'elles présentent des propriétés de corrélations particulière, et sont à ce titre très utilisées également dans les techniques de télécommunications.

Revenons maintenant sur cette notion de primitivité. Soit $f(X)$ le polynôme irréductible de $\mathbb{F}_2[X]$ en question. On peut contruire le corps $\mathbb{F}_{2^L}$ comme $\mathbb{F}_{2^L} = \mathbb{F}_2[X]/f(X)$ (polynômes réduits modulo $f(X)$). On peut alors donner une définition équivalente de la primitivité, que nous utiliserons pour tester si tel ou tel polynôme est primitif ou non. Rappelons tout d'abord que le sous-groupe multiplicatif d'un corps fini est cyclique. Autrement dit, pour tout $\alpha \in \mathbb{F}_{2^L}^* = \mathbb{F}_{2^L} \backslash \{0\}$, on a $\alpha^{2^L-1} = 1$.

Définition 2.3. *Soit $f(X)$ un polynôme irréductible de $\mathbb{F}_2[X]$, de degré L. Il est dit* primitif *si l'une de ses racines engendre le sous-groupe multiplicatif $\mathbb{F}_{2^L}^*$.*

Soit α une racine de f ; on a $f(\alpha) = 0$. Dire que α génère le groupe multiplicatif, c'est dire que les éléments $\alpha, \alpha^2, \alpha^3, \ldots$ correspondent bien à tous les éléments non nuls du corps, et donc qu'il y en a $2^L - 1$ distincts.

Définition 2.4. *En fait, on peut définir l'*ordre d'un élément α *comme le plus petit t tel que $\alpha^t = 1$.*

L'ordre d'un élément divise le cardinal du groupe (théorème de Lagrange). Ceci signifie ici que l'ordre d'un élément divise $2^L - 1$. Ce qu'on cherche à savoir, c'est si l'ordre de α est égal à $2^L - 1$ (polynôme primitif) ou non (polynôme non primitif).

Exemple 2-2. Soit $f(X) = X^3 + X + 1$ irréductible, et soit α tel que $f(\alpha) = 0$. Enumérons les puissances de α :

$$\begin{aligned}
\alpha^1 &= \alpha \\
\alpha^2 &= \alpha^2 \\
\alpha^3 &= \alpha + 1 \\
\alpha^4 &= \alpha^2 + \alpha \\
\alpha^5 &= \alpha^3 + \alpha^2 = \alpha^2 + \alpha + 1 \\
\alpha^6 &= \alpha^3 + \alpha^2 + \alpha = \alpha^2 + 1 \\
\alpha^7 &= \alpha^3 + \alpha = 1
\end{aligned}$$

On peut en conclure que le polynôme $X^3 + X + 1$ est primitif.

On peut remarquer que puisque l'on sait que l'ordre de α divise $2^L - 1$, il suffit de regarder les α^t pour tous les diviseurs $1 < t < 2^L - 1$ de $2^L - 1$, et de voir si ces α^t sont égaux à 1 ou non. Le polynôme est primitif si et seulement si aucun de ces α^t n'est égal à 1.

Exemple 2-3. Reprenons l'exemple ci-dessus. $L = 3$, donc $2^L - 1 = 7$. Comme 7 est un nombre premier, il n'y a aucun diviseur $1 < t < 7$ de 7, et donc on est certain que le polynôme est primitif.

3 Une attaque : algorithme de Berlekamp-Massey

En 1968, E. Berlekamp a proposé dans son livre un algorithme pour le décodage des codes BCH. En 1969, J.L. Massey a montré que ce dernier permet également de trouver le plus petit LFSR générant la suite $(s_n)_{n\geq 0}$ à partir uniquement de ses $2\Lambda(s)$ premiers bits. Ce résultat repose sur le lemme suivant :

Lemme 3.1. *Soit L_N la longueur minimale d'un LFSR qui génère les bits $s_0, s_1, \ldots, s_{N-1}$ mais qui ne génère pas $s_0, s_1, \ldots, s_N$. Alors la longueur minimale L_{N+1} d'un LFSR générant $s_0, s_1, \ldots, s_N$ vérifie*

$$L_{N+1} \geq \max(N + 1 - L_N, L_N).$$

L'algorithme de Berlekamp-Massey permet de construire pour des valeurs successives de N un LFSR de longueur minimale L_N et de polynôme de rétroaction f_N qui génère les N premiers bits de $(s_n)_{n\geq 0}$. Par convention, on suppose que la suite nulle est générée par le LFSR de longueur 0 et de polynôme de rétroaction $f(X) = 1$.

Théorème 3.1. *Soit $(s_n)_{n\geq 0}$ une suite binaire à récurrence linéaire. Considérons un LFSR de longueur minimale L_N et de polynôme de rétroaction $f_N(X) = 1 + \sum_{i=1}^{L_N} c_i^N X^i$ qui génère les N premiers bits de $(s_n)_{n\geq 0}$. On pose $L_0 = 0$ et $f_0(X) = 1$. Soit $d_N = s_N + \sum_{i=1}^{L_N} c_i^N s_{N-i} \mod 2$. Alors on a :*

- *si $d_N = 0$, alors $L_{N+1} = L_N$;*
- *si $d_N = 1$, alors $L_{N+1} = \max(N + 1 - L_N, L_N)$.*

De plus, on a :

- *si $d_N = 0$, alors $f_{N+1}(X) = f_N(X)$;*
- *si $d_N = 1$, alors $f_{N+1}(X) = f_N(X) + f_m(X)X^{N-m}$, où m est le plus grand entier inférieur à N tel que $L_m < L_N$.*

Je ne donnerai pas ici la preuve de ce résultat. Cependant elle est constructive et donne un algorithme qui permet de trouver un LFSR de taille minimale générant une suite donnée.

Algorithme 1 Algorithme de Berlekamp-Massey

Entrée: Une suite binaire $\mathbf{s} = s_0, \ldots, s_{n-1}$ de longueur n
Sortie: Le LFSR de longueur minimale produisant la suite $\mathbf{s}$

$f(X) \leftarrow 1$, $L \leftarrow 0$, $m \leftarrow -1$, $g(X) \leftarrow 1$
pour N allant de 0 à $n-1$ **faire**
  $d \leftarrow s_N + \sum_{i=1}^{L} c_i s_{N-i} \mod 2$ avec $f(X) = 1 + \sum_{i=1}^{L} c_i X^i$
  si $d = 1$ **alors**
    $t(X) \leftarrow f(X)$
    $f(X) \leftarrow f(X) + g(X)X^{N-m}$
    si $2L \leq N$ **alors**
      $L \leftarrow N + 1 - L$, $m \leftarrow N$, $g(X) \leftarrow t(X)$
    fin si
  fin si
fin pour

Exemple 3-4. Appliquons l'algorithme de Berlekamp-Massey à la suite 0110010101 de longueur 10. Voici les résultats fournis par l'algorithme :

N	s_N	d	L	$f(X)$	m	$g(X)$
			0	1	-1	1
0	0	0	0	1	-1	1
1	1	1	2	$1+X^2$	1	1
2	1	1	2	$1+X+X^2$	1	1
3	0	0	2	$1+X+X^2$	1	1
4	0	1	3	$1+X+X^2+X^3$	4	$1+X+X^2$
5	1	0	3	$1+X+X^2+X^3$	4	$1+X+X^2$
6	0	1	4	$1+X+X^4$	6	$1+X+X^2+X^3$
7	1	1	4	$1+X^2+X^3$	6	$1+X+X^2+X^3$
8	0	1	5	$1+X^4+X^5$	8	$1+X^2+X^3$
9	1	0	5	$1+X^4+X^5$	8	$1+X^2+X^3$

Ainsi un LFSR de taille minimale générant la suite 0110010101 a pour longueur 5 et polynôme de rétroaction $1+X^4+X^5$.

Théorème 3.2. *Soit $(s_n)_{n\geq 0}$ une suite binaire à récurrence linéaire de complexité linéaire $\Lambda(s)$. L'algorithme de Berlekamp-Massey détermine l'unique LFSR de longueur $\Lambda(s)$ qui génère $(s_n)_{n\geq 0}$ à partir de n'importe quelle suite de $2\Lambda(s)$ bits consécutifs de $(s_n)_{n\geq 0}$.*

La complexité linéaire d'un LFSR est donc un paramètre déterminant pour sa sécurité cryptographique : l'observation d'un petit nombre de bits seulement de la suite permet en effet de la reconstituer entièrement si $\Lambda(s)$ est petit.

Ceci amène à l'étude de la combinaison ou du filtrage de un ou plusieurs LFSR, à l'aide de fonctions booléennes. Bien évidemment, ces fonctions doivent être choisies avec beaucoup de soin. Ce contexte donne encore lieu aujourd'hui à une activité de recherche importante, tant dans la recherche d'attaques que dans la recherche de bonnes fonctions.