Générateurs Pseudo-Aléatoires à Base De Registres Filtrés

Introduction

la plupart des générateurs pseudo-aléatoires sont construits en utilisant les registres filtrés.
En effet ces registres sont faciles à implémenter mais également économique dans leur mise en oeuvre matériel.Raisons pour laquelle on les utilise dans les chiffrements par flot mais aussi par bloc.
Pour mieux illustrer cette partie du mémoire nous :
•manipulerons et étudierons les propriétés des LFSR.
•mais également mettrons au point des algorithmes pour retrouver le plus petit LFSR générant une suite de bits donnée.

3.1 Régistre à Décalage à Rétroaction Linéaire

Définition 3.1.1. Un registre à décalage à rétroaction linéaire est un dispositif permettant d'engendrer une séquence de r bits $(s_i, ..., s_{i+r-1})$ satisfaisant à la relation de récurrence linéaire suivante :

$$a_{n+r} = c_0 a_{n+r-1} + c_1 a_{n+r-2} + ... + c_{r-1} a_n$$

où $(c_0, c_1, ..., c_{r-1}) \in \mathbb{F}_q$ sont les coefficients de connexion du LFSR.

Cependant pour mieux illustrer ces propos nous allons introduit tout juste après l'architecture de fonctionnement d'un LFSR :

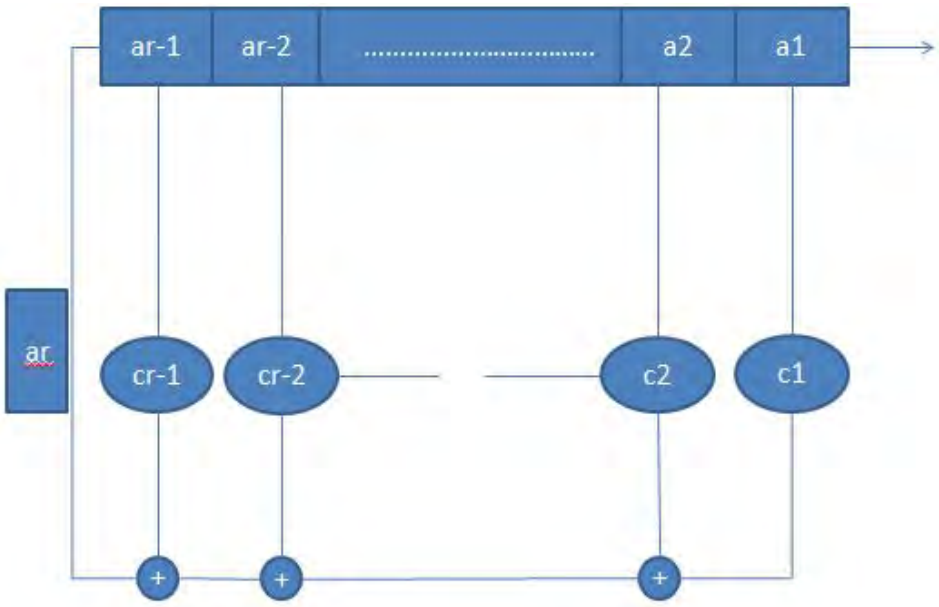


FIGURE 3.1 – Fonctionnement d'un LFSR

Propriété 3.1.1. Les propriétés fondamentales d'un LFSR sont les suivantes :

- être bien adapté à une configuration matérielle
- être capable de produire de grandes périodes de séquences binaires
- mais également des séquences qui ont de bonnes propriétés statistiques
- Et grâce à sa nature, un LFSR peut être facilement analysé en utilisant des modèles mathématiques.

Principe 3.1.1. Le principe de base d'un LFSR de $2bits$ à base de XOR initialisée avec 01 est le suivant :

- d'abord on rempli le premier étage par le résultat (bit s_2) d'une fonction de rétroaction linéaire qui prend en compte l'état d'un ou de plusieurs étages.
- Ensuite ce bit (s_2) est placé dans la cellule de gauche du premier registre.
- Après les autres bits sont décalés vers la droite et le bit s_0 va constituer la sortie du registre.
- Et au troisiéme top d'horloge l'état du registre est identique à son état initial. On dit alors que le LFSR est de période 3.

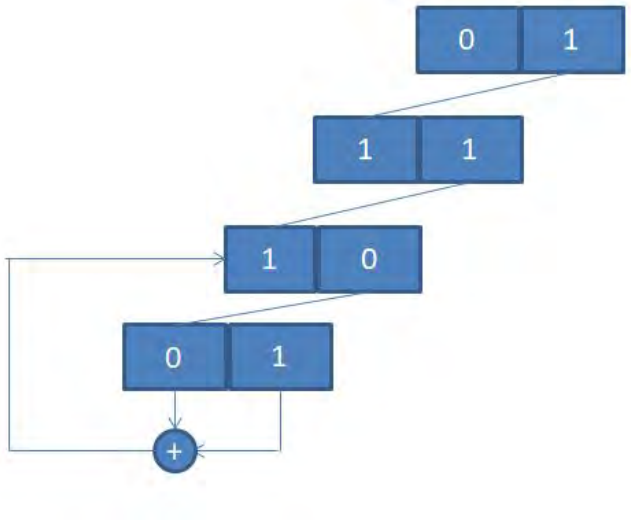


FIGURE 3.2 – Principe du LFSR

3.1.1 Séquences Générées par un LFSR

Définition 3.1.2. Soient $a_0, a_1, ..., a_{r-1}$ une séquence de bits. On dit que les (a_i) sont générés par un LFSR de taille r s'il suit la relation de récurrence linéaire suivante :

$$a_{n+r} = \sum_{i=o}^{r-1} c_i a_{n+i}$$

Proposition 3.1.1. *Toute séquence binaire à récurrence linéaire homogène d'ordre r est dite ultimement périodique si* $\forall n \geq n_0$ *sa plus petite période T est inférieure ou égale à* $2^r - 1$.
De plus si $\forall n \geq 0$ *sa période* $T = 2^r - 1$ *, alors la séquence est dite périodique.*

Exemple 3.1.1. Soient deux suites récurrentes linéaires d'ordre 4 initialisées par $(a_0, a_1, a_2, a_3) = (0, 0, 0, 1)$.

$$a_{n+4} = a_{n+3} + a_{n+2}$$

$$a_{n+4} = a_{n+1} + a_n$$

• Pour la séquence1 on a :

$$S_1 = \underline{00}011011011.....\forall n \geq 2$$

De plus la période $T = 3 < (2^4 - 1)$ donc cette suite est ultimement périodique.
• Pour la séquence2 on a :

$$S_2 = \underline{0}0010011010111\underline{0}0010011010111.....\forall n \geq 0$$

De plus la période $T = 15 = (2^4 - 1)$; donc cette suite est périodique.
$*$ Puisque la sécurité des LFSR dépend de la complexité linéaire, nous allons maintenant nous intéresser à la plus petite relation de récurrence permettant de reproduire la séquence $(a_n) = (a_0, a_1, ..., a_{r-1})$ et pour ce faire associons à cette dernière la série génératrice suivante :

$$a(X) = \sum_{n=o}^{\infty} a_n X^n$$

Cette approche qui a été introduite en 1959 par Niezereiter nous a permis de décrire le développement en série formelle de la séquence (a_n) sous forme d'une fraction rationnelle et ceci grâce à l'intervention du polynôme de rétroaction ci-dessous :

$$f(X) = 1 + c_1 X + c_2 X^2 + c_3 X^3 + ... + c_r X^r$$

Pour ce faire introduisons d'abord le théorème suivant :
Mais avant tout qu'est ce qu'une série formelle

3.1.1.1 Série Formelle

Définition 3.1.3. Soit A un anneau commutatif intègre.
On appelle série formelle(ou série génératrice) sur A toute expression symbolique :
$a_0 + a_1 X + a_2 X^2 + a_3 X^3 + ...$ où les $a_i \in A$, $\forall i \in \mathbb{N}$.
Le symbole X est appelé l'indéterminée.
On note une série formelle en X par :

$$S(X) = (a_n) = \sum_{i \in \mathbb{N}} a_i X^i = a_0 + a_1 X + a_2 X^2 + a_3 X^3 + ...$$

Exemple 3.1.2. $S(x) = 1 + 5x + x^2 + 7x^3 + ...$ série formelle sur
$(\mathbb{Z}, +, \times)$
$S(x) = 1 + \frac{1}{4}x + \frac{3}{5}x^2 + ...$ série formelle sur $(\mathbb{Q}, +, \times)$
$S(x) = 1 + x + x^2 + ...$ série formelle sur $(\mathbb{F}_2, +, \times)$

Remarque. Une série formelle n'est pas une fonction mais simplement une expression liée à une suite d'éléments (a_i).
De plus x n'est pas une variable et a_0 est toujours différent de zéro $(a_0) \neq 0$
Maintenant nous pouvons introduire le théorème :

Théoreme 3.1.2. *Soient $(a_n) = (a_0, a_1, ..., a_{r-1})$ une séquence de r bits ; elle est produite par un LFSR de polynôme de rétroaction $f(X) = 1 + c_1X + c_2X^2 + c_3X^3 + ... + c_rX^r$ si et seulement si son développement en série formelle $a(X) = \sum_{n=o}^{\infty} a_nX^n$ s'écrit de la manière suivante :*

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

où f et g sont des polynômes de $\mathbb{F}_2[X]$ et $deg(g) < deg(f)$
De plus $g(X) = \sum_{i=1}^{r} X^i \sum_{j=0}^{i-1} c_{i-j}a_j$ est déterminé par l'état initial du registre.

3.1.1.2 Preuve du Théorème

Soit $(a_n) = (a_0, a_1, ..., a_{r-1})$ la séquence produite par un LFSR .
$\Rightarrow$ Supposons que son polynôme de rétroaction est $f(X) = 1 + c_1X + c_2X^2 + c_3X^3 + ... + c_rX^r$ et montrons que sa série génératrice s'écrit sous cette forme :

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

• Soit $a(X) = \sum_{n=o}^{\infty} a_nX^n$
avec $(a_n) = a_0, a_1, ...$ la séquence de sortie générée par un LFSR
Or $a_n = \sum_{i=1}^{r} c_ia_{n-i}$
On a donc :
$a(X) = \sum_{n=o}^{\infty} \sum_{i=1}^{r} c_ia_{n-i}X^n = \sum_{i=1}^{r} c_iX^i \sum_{n=0}^{\infty} a_{n-i}X^{n-i}$
$= \sum_{i=1}^{r} c_iX^i \{ \sum_{n=i}^{\infty} a_{n-i}X^{n-i} + \sum_{n=o}^{i-1} a_{n-i}X^{n-i}\}$
Pour la somme $\sum_{n=i}^{\infty} a_{n-i}X^{n-i}$ faisons un changement de variable c'est-à-dire $n' = n - i$
On a donc
$\sum_{n=i}^{\infty} a_{n-i}X^{n-i} = \sum_{n'=0}^{\infty} a_{n'}X^{n'} = \sum_{n=0}^{\infty} a_nX^n$
En remplaçant on a :
$a(X) = \sum_{i=1}^{r} c_iX^i \{ \sum_{n=0}^{\infty} a_nX^n + \sum_{n=o}^{i-1} a_{n-i}X^{n-i}\} = \sum_{i=1}^{r} c_iX^i \{a(X) + \sum_{n=o}^{i-1} a_{n-i}X^{n-i}\}$
$a(X) - \sum_{i=1}^{r} c_iX^i \{a(X)\} = \sum_{i=1}^{r} c_iX^i \sum_{n=o}^{i-1} a_{n-i}X^{n-i}$
$a(X)\{1 - \sum_{i=1}^{r} c_iX^i\} = \sum_{i=1}^{r} c_iX^i \sum_{n=o}^{i-1} a_{n-i}X^{n-i}$
Puisque $f \in \mathbb{F}_2[X]$ on a donc $\{1 - \sum_{i=1}^{r} c_iX^i\} = \{1 + \sum_{i=1}^{r} c_iX^i\} = f(X)$
Donc
$a(X)f(X) = \sum_{i=1}^{r} c_iX^i \sum_{n=o}^{i-1} a_{n-i}X^{n-i} = \sum_{n=o}^{i-1} X^n \sum_{i=1}^{r} c_ia_{n-i}$
En posant $g(X) = \sum_{n=o}^{i-1} X^n \sum_{i=1}^{r} c_ia_{n-i}$

On a donc
$a(X)f(X) = g(X)$
D'où

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

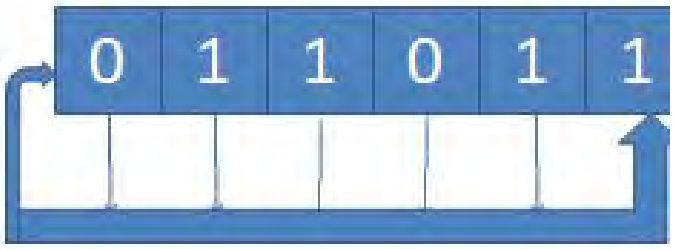
$\Leftarrow$ Supposons que le développement en série formelle de la suite $(a_n)_{n\geq 0}$ s'écrit sous cette forme : $a(X) = \frac{g(X)}{f(X)}$ avec $pgcd(g(X), f(X)) \neq 1$ et montrons l'existence de son polynôme de rétroaction.
Puisque $pgcd(g(X), f(X)) \neq 1$ nous pouvons donc introduire les 4 points suivants :

- Soit (a_n) une séquence binaire à rétroaction linéaire d'ordre r 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 :

$$a(X) = \frac{g_0(X)}{f_0(X)}$$

- Ce polynôme de rétroaction minimal f_0 est le polynôme de plus bas degré parmi les polynômes de rétroaction de tous les LFSRs possibles qui peuvent générés la séquence (a_n).
- Et ce plus bas degré de f_0 également appelé complexité linéaire du LFSR est la longueur du plus petit LFSR permettant d'engendrer la séquence (a_n).

Exemple 3.1.3. Soit (a_n) la séquence produite par le LFSR suivant :



Son polynôme de rétroaction est :

$$f(X) = 1 + X + X^2 + X^3 + X^4 + X^5$$

Soit g le polynôme déterminé par l'état initial du registre :

$$g(X) = \sum_{n=o}^{4} X^n \sum_{i=o}^{n} c_{n-i} a_i$$

On a ainsi :

$g_0(X) = X^0 \cdot c_0 \cdot a_0 = 1$

$g_1(X) = X^1(c_1 \cdot a_0 + c_0 \cdot a_1) = X^1(1+1) = 0$

$g_2(X) = X^2(c_2 \cdot a_0 + c_1 \cdot a_1 + c_0 \cdot a_2) = X^2(1+1+0) = 0$

$g_3(X) = X^3(c_3 \cdot a_0 + c_2 \cdot a_1 + c_1 \cdot a_2 + c_0 \cdot a_3) = X^3(1+1+0+1) = X^3$

$g_4(X) = X^4(c_4 \cdot a_0 + c_3 \cdot a_1 + c_2 \cdot a_2 + c_1 \cdot a_3 + c_0 \cdot a_4) = X^4(1+1+0+1+1) = 0$

Finalement on a :

$$g(X) = X^3 + 1$$

Or la série génératrice de $(a_n)_{n\geq 0}$ est :

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

Donc on a :

$$a(X) = \frac{X^3+1}{1+X+X^2+X^3+X^4+X^5} = \frac{X^3+1}{(X^3+1)\cdot(1+X+X^2)} = \frac{1}{1+X+X^2} = \frac{1}{f_0(X)}$$

$\Rightarrow f_0(X) = 1+X+X^2$ donc la séquence (a_n) est produite par un LFSR de longueur $(2^2-1) = 3$ et de complexité linéaire égale à 2.

Remarque. Si on connait la séquence on connait le polynôme de rétroaction. Cependant si ce polynôme est irréductible , unitaire et de période égale à $2^L - 1$ avec L la complexité linéaire dans un corps donné $\mathbb{F}_q$, il devient automatiquement le polynome minimal.Dans le cas contraire le polynôme minimal sera l'un de ses diviseurs.

$*$ Donc nous pouvons conclure que d'après le théorème énoncé ci-dessus il existe une bijection entre l'ensemble des suites d'ordre r et l'ensemble des séries formelles et pour mieux illustrer cela nous avons proposé tout juste après un schéma comme guise d'exemple.

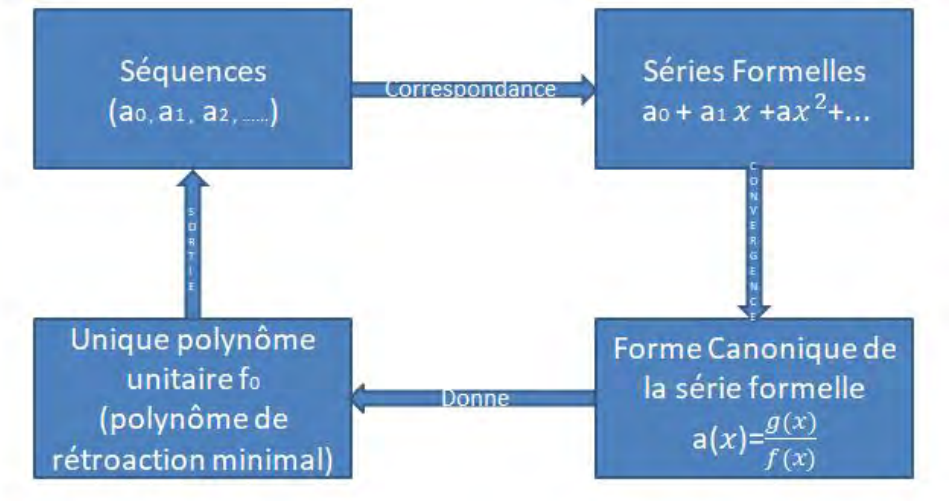


FIGURE 3.3 – Bijection entre les séries formelles et les suites récurrentes linéaires

3.2 LFSR et Cryptographie

Un LFSR n'est pas cryptographiquement sûr parce que les progrès faites récemment dans le domaine de la cryptanalyse notamment les attaques dites algébriques proposées dernièrement mettent toutefois en évidence des faiblesses inhérentes à de nombreux générateurs de ce type. En effet

- si les coefficients du polynôme de rétroaction sont publics, il suffit à un attaquant qui connait L bits consécutifs de la suite d'appliquer la relation de récurrence pour retrouver tous les bits suivants.
- dans le cas contraire où les coefficients de rétroaction sont secrets, l'algorithme de Berlekamp-Massey permet de les reconstituer ainsi que l'état initial à partir de $2L$ bits de suite chiffrante.

3.2.1 Attaque dite Algèbrique

3.2.1.1 Méthode par résolution du système linéaire

Une méthode plus efficace pour calculer la complexité linéaire d'une suite $A = (a_0, a_1, a_2, ..., a_{r-1})$ est de remarquer qu'un LFSR de longueur r génère la suite A si et seulement si ces coefficients $(c_0, c_1, c_2, ..., c_{r-1})$ vérifient un système de $n - r$ équations linéaires définies sur $\mathbb{F}_q$ par :

$$\left\{\begin{array}{rcl} c_0a_{r-1} + c_1a_{r-2} + ... + c_{r-1}a_0 & = & a_r \\ c_0a_r + c_1a_{r-1} + ... + c_{r-1}a_1 & = & a_{r+1} \\ c_0a_{r+1} + c_1a_r + ... + c_{r-1}a_2 & = & a_{r+2} \\ \cdots & & \\ \cdots & & \\ \cdots & & \\ c_0a_{n-r-2} + c_1a_{n-r-3} + ... + c_{r-1}a_{n-1} & = & a_{n-r-1} \end{array}\right\}$$

Afin de résoudre un tel système à r inconnues dans $\mathbb{F}_q$, une méthode est de l'exprimer sous la forme matricielle

$$M \cdot c = a$$

avec M(matrice) , c(les coefficients) et $a = (a_r, a_{r+1}, a_{r+2}, ..., a_{n-r-1})$ la séquence
Chaque solution de ce système nous donnera un LFSR et toute absence de solutions nous permettra d'affirmer que la complexité linéaire est supérieure à r.

Exemple 3.2.1. Soit $T = (1,1,1,0,1,1,0,1)$ une séquence , pour $l = 4$
on a l'équation matricielle suivante :

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Après avoir appliqué la méthode du pivot de Gauss on a

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Ce qui implique que :

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

finalement

$$\left\{ \begin{array}{rcl} c_0 & = & 1 \\ c_1 + c_3 & = & 0 \\ c_1 + c_2 & = & 0 \end{array} \right\} \Longrightarrow \left\{ \begin{array}{rcl} c_0 & = & 1 \\ c_1 & = & 1 \\ c_2 & = & 1 \\ c_3 & = & 1 \end{array} \right\}$$

En remplaçant dans la formule de la matrice compagnon A on a :

$$\begin{bmatrix} 0 & 0 & 0 & c_0 \\ 1 & 0 & 0 & c_1 \\ 0 & 1 & 0 & c_2 \\ 0 & 0 & 1 & c_3 \end{bmatrix} \quad = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Maintenant calculons l'inverse de la matrice A puisqu'elle est inversible.
D'après la formule :

$$A^{-1} = \frac{1}{detA} \cdot {}^{T}comA$$

on a :
$detA = -1 = 1 \neq 0$ dans $\mathbb{F}_2$

De plus

$$\text{ComA} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \Longrightarrow 1 \times {}^{T}comA = A^{-1} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Pour la vérification on a

$$AA^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I_4$$

Maintenant essayons de retrouver les premiers termes de la suite :

On a donc :

$$(1110)A^{-1} = (1110) \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = (\mathbf{1}111)$$

$$(1111)A^{-1} = (1111) \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = (\mathbf{0}111)$$

$$(0111)A^{-1} = (0111) \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = (\mathbf{1}011)$$

$$(1011)A^{-1} = (1011) \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = (\mathbf{1}101)$$

$$(1101)A^{-1} = (1101) \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = (1110)$$

On remarque que le dernier résultat est identique aux 4 premiers bits utilisés dans la démonstration. Donc pour déterminer l'état initial du LFSR correspondant on utilisera les chiffres écrient en gras des 4 premiers résultats. Ainsi la sortie sera :

$$S = 1011110111101111$$

Cependant retrouvez l'état initial revient à retrouver le LFSR correspondant.

3.2.2 Attaque par Berlekamp-Massey

En 1969, J.Massey a montré que l'algorithme proposé par Berlekamp pour le décodage des codes BCH pouvait être adapté pour retrouver le polynôme de rétroaction d'un LFSR à partir uniquement des $2L$ premiers bits de la séquence produite s. D'où le fameux nom de Berlekamp-Massey.

Définition 3.2.1. La complexité linéaire d'une suite est la dimension de l'espace vectoriel engendré par la suite et l'ensemble de ses décalées. Elle est aussi notée :
$L = k(a) = dim(vect((a), D.(a), D^2(a), ...)$ où D est l'opérateur décalage et $D^i.(a)$ est la i^{eme} décalée de (a).

Principe 3.2.1. Cet algorithme retourne pour $N = 2L$ bits le polynôme de rétroaction f du LFSR de départ.
Ce LFSR génére les N premiers bits de la suite et est de complexité linéaire L.

Entrée : $s_0, s_1, ..., s_{n-1}$ une séquence de longueur n
Initialisation :
$g_j(x)$; $h_j(x)$; m_j / $g_0(x) = 1$; $h_0(x) = x$; $m_0 = 0$
Pour **i** $= 0$.....**do** :
$* \; g_{j+1}(x) = g_j(x) - b_j \cdot h_j(x)$
Avec b_j le coefficient de x^j dans $G(x) \cdot g_j(x)$
Dans $\mathbb{F}_2$ on a $b_j^{-1} = b_j = 1$ si $b_j \neq 0$

$$* \; h_{j+1} = \begin{cases} b_j^{-1} \cdot x \cdot g_j(x) & \text{si } b_j \neq 0 \text{ et } m_j \geqslant 0 \\ x \cdot h_j(x) \end{cases}$$

Et :

$$* \; m_{j+1} = \begin{cases} -m_j & \text{si } b_j \neq 0 \text{ et } m_j \geqslant 0 \\ m_j + 1 & \text{sinon} \end{cases}$$

Si on définit $m(x)$ comme étant le polynôme minimal de la séquence alors on a :

$$m(x) = x^k \cdot g_{2k}(\frac{1}{x})$$

Exemple 3.2.2. Soit $S = 1101$ une séquence

$G(x) = 1 + x + x^3$ sa série génératrice
On a donc :

j	$g_j(x)$	$h_j(x)$	m_j	b_j
0	1	x	0	1
1	$1+x$	x	0	0
2	$1+x$	x^2	1	1
3	$1+x+x^2$	$x+x^2$	-1	0

Ainsi pour $k = 2$ on a

$$m(x) = x^2 \cdot g_4(\frac{1}{x}) = x^2 \cdot (\frac{1}{x^2} + \frac{1}{x} + 1) = (1 + x + x^2)$$

Finalement $m(x) = 1 + x + x^2$
Donc son LFSR correspondant est : $U_{n+2} = U_{n+1} + U_n$

★ Les registres à décalage à rétroaction linéaires sont des dispositifs extrêmement rapides et d'implémentation peu coûteuse , et les séquences qu'ils engendrent ont de bonnes propriétés statistiques.
Mais pour contourner l'effet du problème de la linéarité de ces LFSR trois méthodes ont été mis sur table :

- Associer une fonction non linéaire aux sorties de plusieurs LFSR(Registre combiné ou filtré)
- Ou utiliser une fonction de filtrage non linéaire (1-résiliente) basée sur le contenu d'un seul LFSR
- Ou bien utiliser plusieurs LFSR en parallèle qui peuvent provenir d'un autre LFSR(k-résilientes)

Ainsi pour une explication plus détaillée on prendra les registres combinés ou filtrés comme guise d'exemple.

3.3 Régistres Combinés ou Filtrés

Même lorsqu'on choisit de manière appropriée le polynôme de rétroaction du registre, la complexité linéaire de la suite produite reste généralement trop faible pour se prémunir des attaques comme celle de l'algorithme de berlekamp-Massey et afin de surmonter cet obstacle et d'augmenter la taille de l'espace des clefs, nous allons utiliser k LFSRs en parallèle et combinés leurs sorties par une fonction booléenne.

3.3.1 Fonction Booléenne

Définition 3.3.1. Soit $\mathbb{F}_{q^n}$ un corps à q^n éléments et $\mathbb{F}_{q^n}^{(r)}$ l'espace vectoriel de dimension r sur $\mathbb{F}_{q^n}$. On appelle fonction booléenne à r variables toute fonction ayant r entrées et une sortie dans $\mathbb{F}_{q^n}$. Autrement dit :

$$\begin{array}{rccc} f & : & \mathbb{F}_{q^n}^{(r)} & \longrightarrow & \mathbb{F}_{q^n} \\ & & (a_0, a_1, ..., a_{r-1}) & \longrightarrow & f(a_0, a_1, ..., a_{r-1}) = a_i \end{array}$$

3.3.2 Exemples de Régistres Combinés

3.3.2.1 Générateur de Geff

Définition 3.3.2. Le générateur de Geff est un générateur défini par 3 LFSRs dont les polynômes de rétroaction sont primitifs et de degrés L_1, L_2, L_3 deux-à-deux premiers entre eux.
Les sorties de ces registres sont réunis dans une même fonction booléenne non seulement équilibrée mais aussi avec un degré le plus élevé possible :
De plus la complexité linéaire de la suite produite par ce générateur est :

$$L = \sum L_i = L_1 + L_2 + L_3$$

et sa période est :

$$T = (2^{L_1} - 1) \cdot (2^{L_2} - 1) \cdot (2^{L_3} - 1) \lesssim 2^L$$

Principe 3.3.1. Soit k un entier naturel , f une fonction booléenne définie par :

$$\begin{array}{rccc} f & : & \mathbb{F}_2^k & \longrightarrow & \mathbb{F}_2 \\ & & (x_0, x_1, ..., x_{k-1}) & \longrightarrow & f(x_0, x_1, ..., x_{k-1}) = x_i \end{array}$$

L'architecture donné ci-dessous illustre mieux le principe abordé :

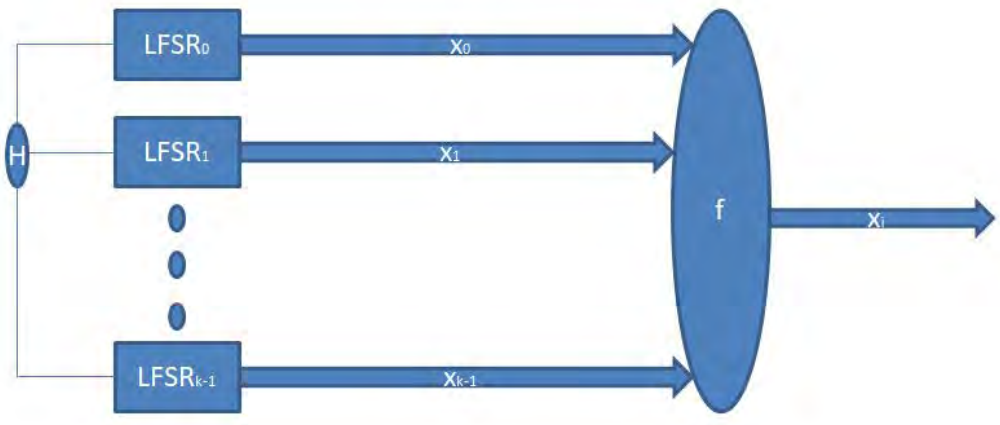


FIGURE 3.4 – Structure des Générateurs Combinés

Exemple 3.3.1. Soient :

$$U_{n+3} = U_{n+1} + U_n$$

$$V_{n+4} = V_{n+1} + V_n$$

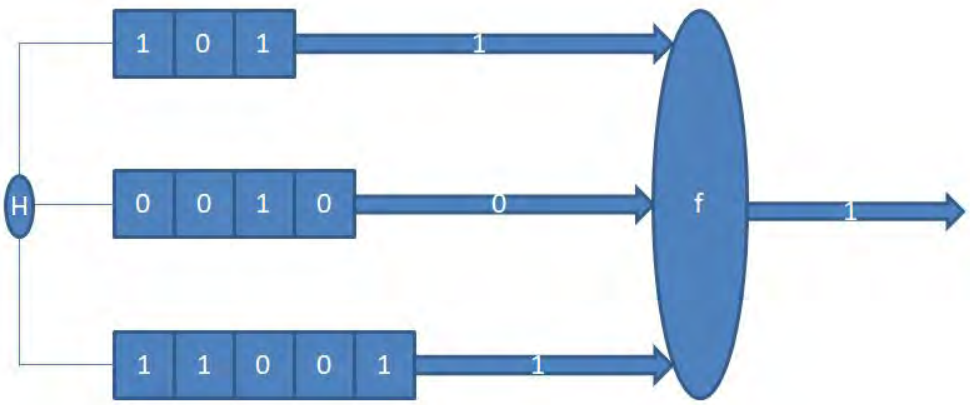
$$W_{n+5} = W_{n+3} + W_n$$

trois LFSRs de complexité linéaire 3 , 4 et 5 respectivement.

Et soit :

$$\begin{array}{rccc} f & : & \mathbb{F}_2^3 & \longrightarrow & \mathbb{F}_2 \\ & & (x_1, x_2, x_3) & \longrightarrow & f(x_1, x_2, x_3) = x_1x_2 + x_2x_3 + x_3 \end{array}$$

la fonction booléenne qui leur est associée.
On a donc après initialisation :



Ainsi $f(x_1, x_2, x_3) = f(1, 0, 1) = x_1x_2 + x_2x_3 + x_3 = 0 + 0 + 1 = 1$
D'où la sortie $X(t) = 1$ et la complexité $L = L_1L_2 + L_2L_3 + L_3 = 12 + 20 + 5 = 37$
Le générateur de Geff a deux avantages importants :
★ son rendement comporte une distribution moyenne égale à 0 et à 1
★ et pour décoder ce dispositif sans la connaissance de la clef, il serait nécessaire de résoudre un système d'équation non-linéaire dont la solution est quasi-impossible.
Mais ceci n'empêchera pas aux attaquants de mettre en évidence les faiblesses inhérentes à de nombreux générateurs de ce genre et ceux-ci grâce aux attaques par Corrélation.

3.3.3 Attaques par Corrélation

Définition 3.3.3. Cette attaque développée par T.Siegenthaler est de type :"diviser pour mieux régner" et consiste à retrouver l'initialisation de chacun des registres indépendamment des autres.

Principe 3.3.2. Pour cela on essaie toute les initialisations du premier registre jusqu'à ce que le nombre de coincidence entre la sortie $X(t)$ de la fonction booléenne et la suite $S(t)$ du premier régistre soit égale à leur probabilité de corrélation.
Cet algorithme s'arrête lorsqu'on est sûr que l'une de ces initialisations marches et on fait de même pour les deux autres régistres $V(t)$ et $W(t)$.

Exemple 3.3.2. Reconsidèrons le générateur de Geff avec :

$$f(x_1, x_2, x_3) = x_1x_2 + x_2x_3 + x_3 = x_2(x_1 + x_3) + x_3 = x_1x_2 + x_3(x_2 + 1)$$

On a donc les probabilités de sorties suivantes :
$P(X(t) = S(t)) = P(X_2(t) = 1) + P(X_3(t) = 0) \cdot P(X_2(t) = 1) = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4}$
$P(X(t) = V(t)) = P(X_1(t) = 1) \cdot P(X_3(t) = 0) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$
$P(X(t) = W(t)) = P(X_2(t) = 0) = \frac{1}{2}$

Remarque. Cette attaque peut-être généralisée en regardant la corrélation de la sortie avec une somme de m sortie de LFSRs.
Donc pour s'en protéger, il faudrait prendre des fonctions de combinaisons garantissant qu'il n'y ait pas de corrélation. Parmi ces fonctions , nous pouvons citer les m-résilientes.
En effet ces fonctions d'ordre m sont non corrélées mais également équilibrées.

Conclusion

La plupart des générateurs pseudo-aléatoires sont construits en utilisant les registres à décalage à rétroaction linéaire.
Mais ces derniers on pour inconvénient de générer des suites linéaires et c'est pour cette raison que les PRNG sont construits en combinant à l'aide d'une fonction non linéaire plusieurs registres à décalage de tailles différentes.
Ainsi pour mesurer la qualité de ces nouveaux générateurs ,nous allons leur faire passer des tests statistiques .