Registre à décalage

à rétroaction linéaire

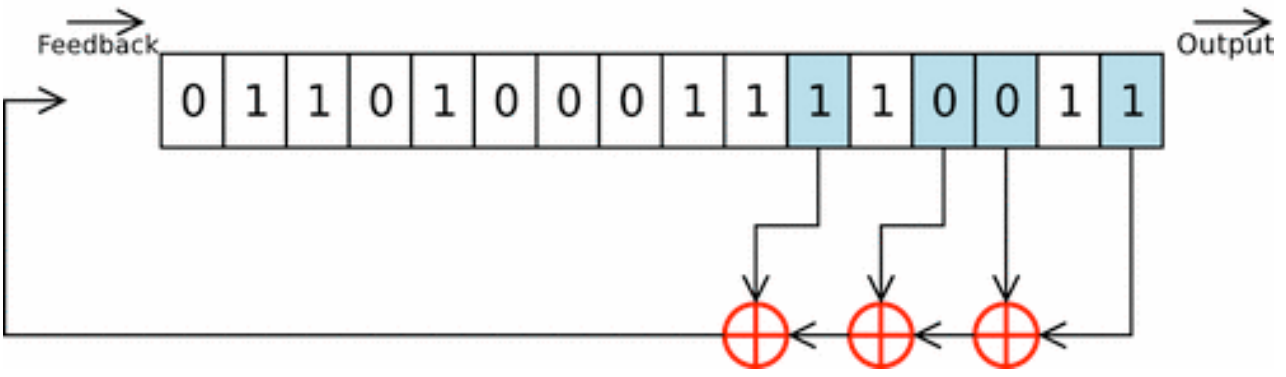
Le registre à décalage à rétroaction linéaire

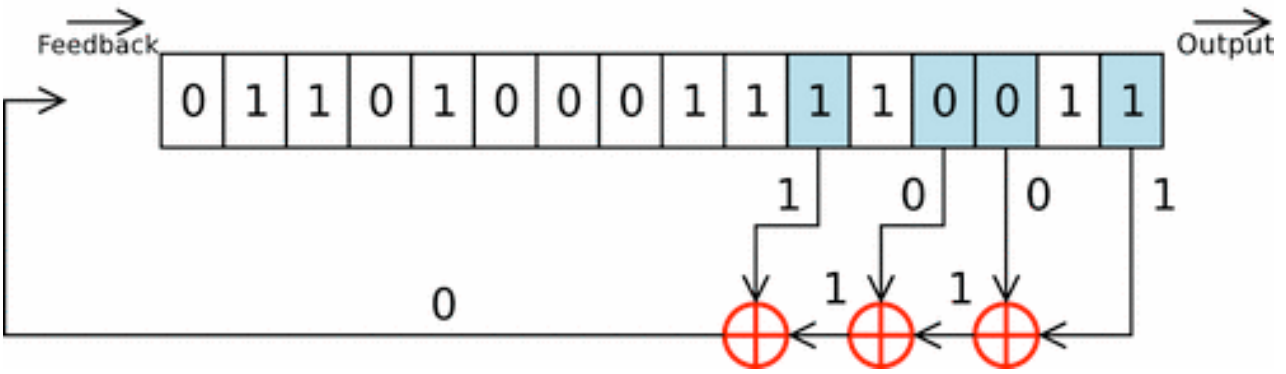
Le registre à décalage à rétroaction linéaire constitue l'élément de base des générateurs pseudo-aléatoires utilisés pour la génération de la suite chiffrante.

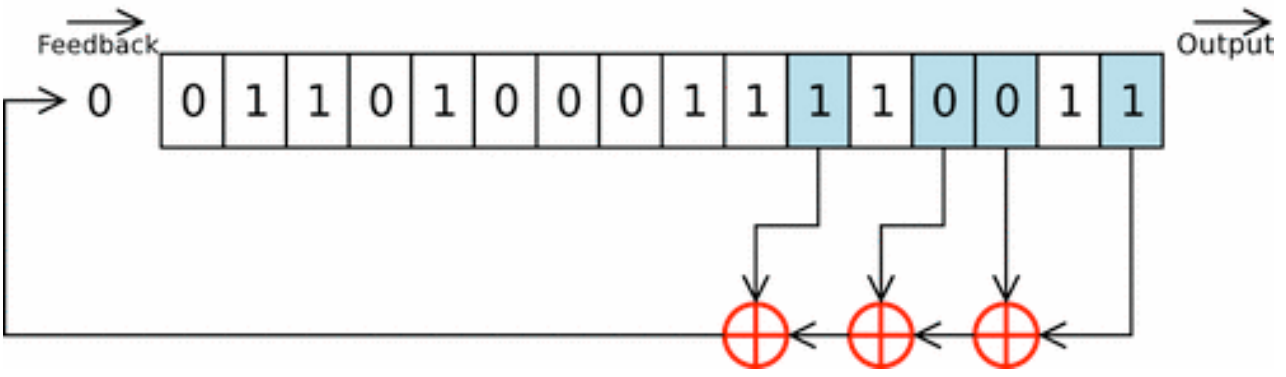
Définition 1 *Un* **registre à décalage à rétroaction linéaire** *binaire de longueur* L *est composé d'un registre à décalage contenant une suite de* L *bits* $(s_i, \ldots, s_{i+L-1})$ *et d'une fonction de rétroaction linéaire.*

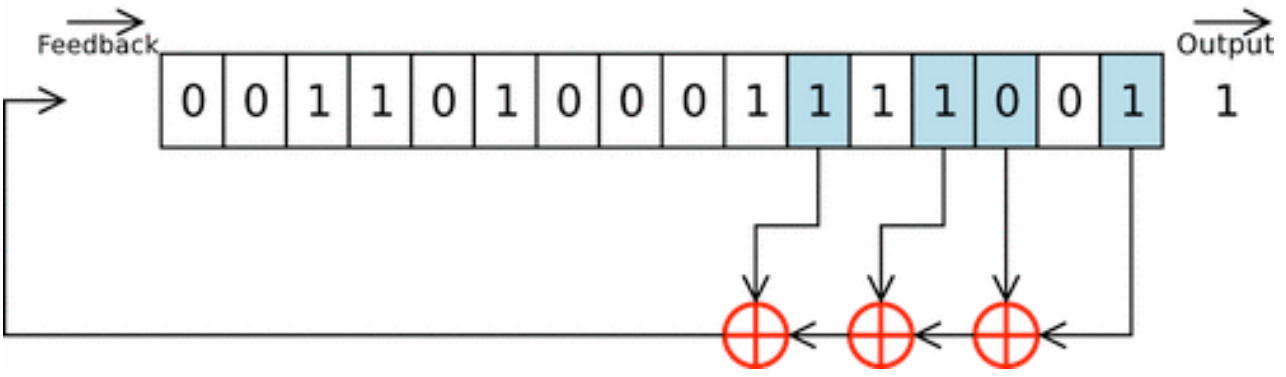
On l'appelle aussi par son acronyme anglais:

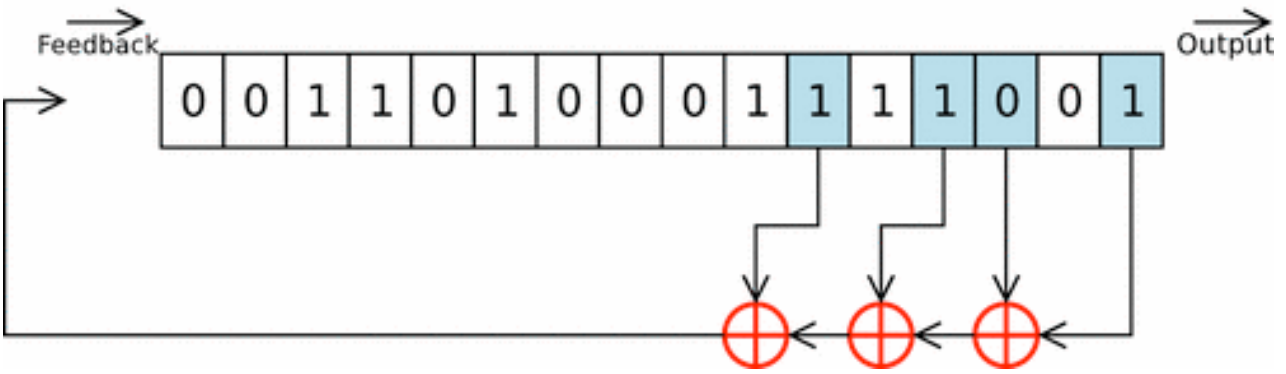
LFSR (Linear Feedback Shift Register)

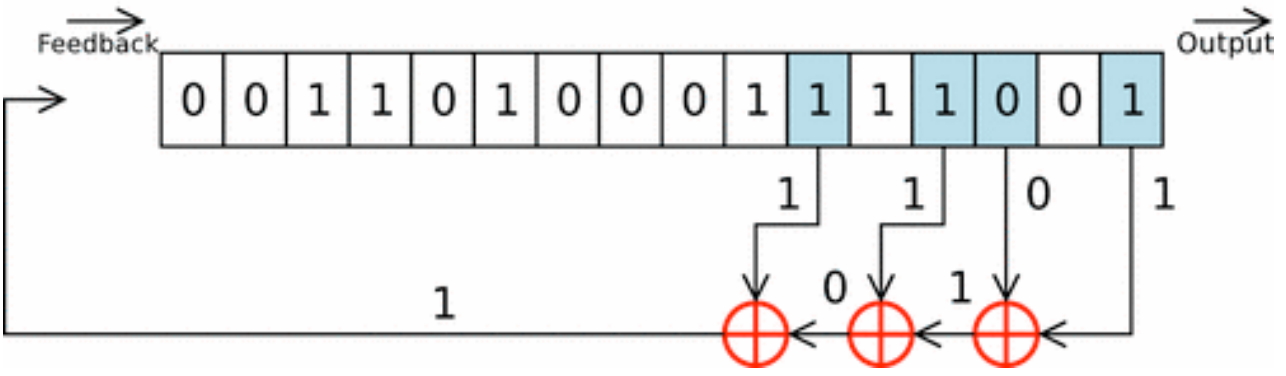

Feedback
Output
0 1 1 0 1 0 0 0 1 1 1 1 0 0 1 1

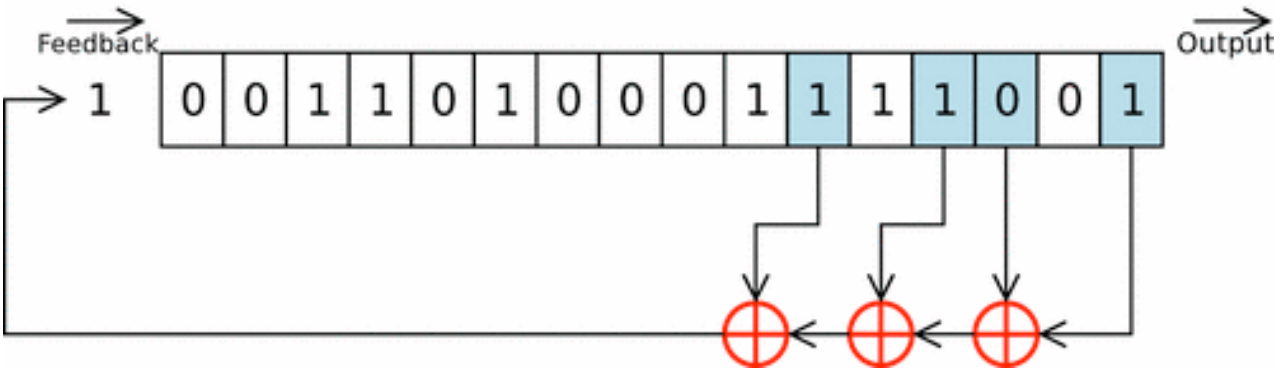

Feedback
Output
0 1 1 0 1 0 0 0 1 1 1 1 0 0 1 1
1 0 0 1
0
1 1

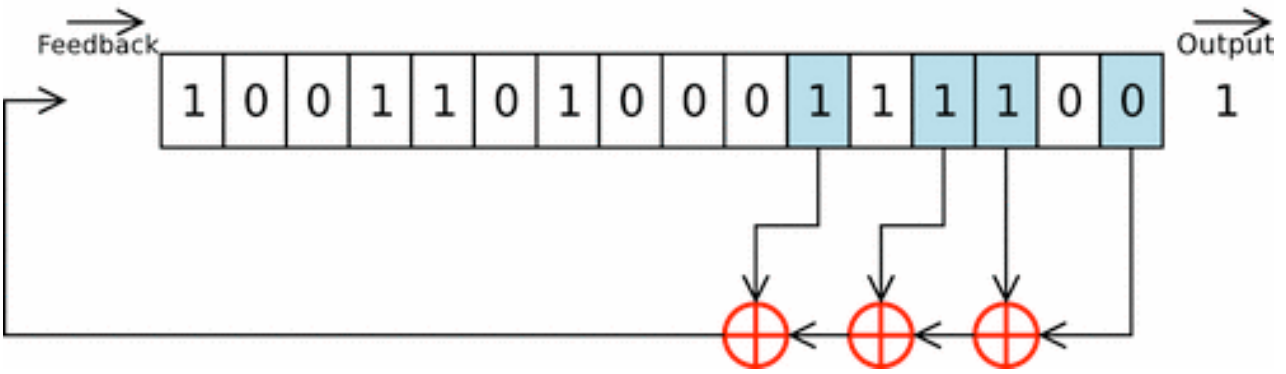

Feedback
Output
0
0 1 1 0 1 0 0 0 1 1 1 1 0 0 1 1

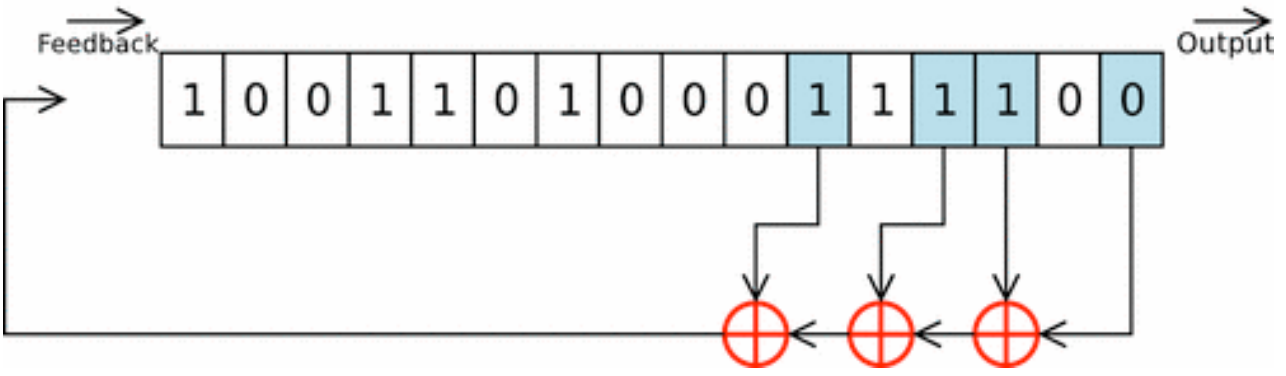

Feedback
Output
0 0 1 1 0 1 0 0 0 1 1 1 1 0 0 1
1

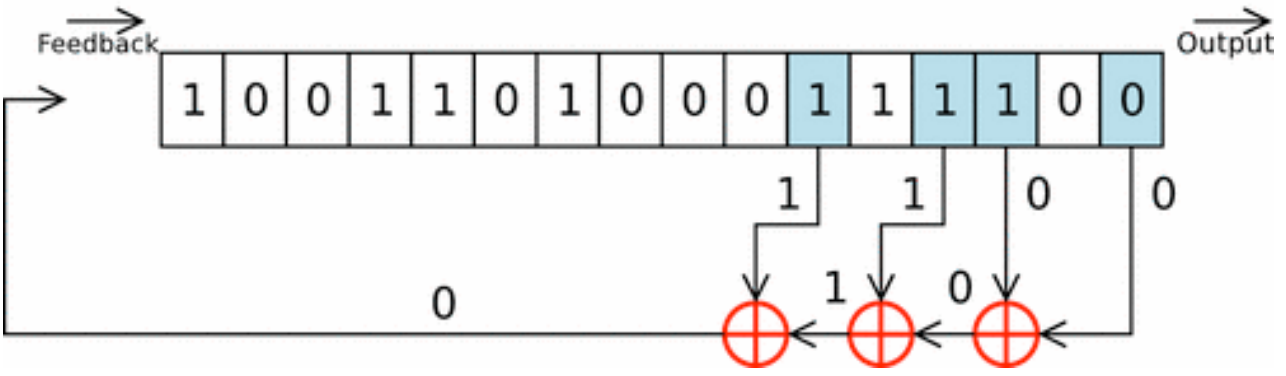

Feedback
Output
0 0 1 1 0 1 0 0 0 1 1 1 1 0 0 1

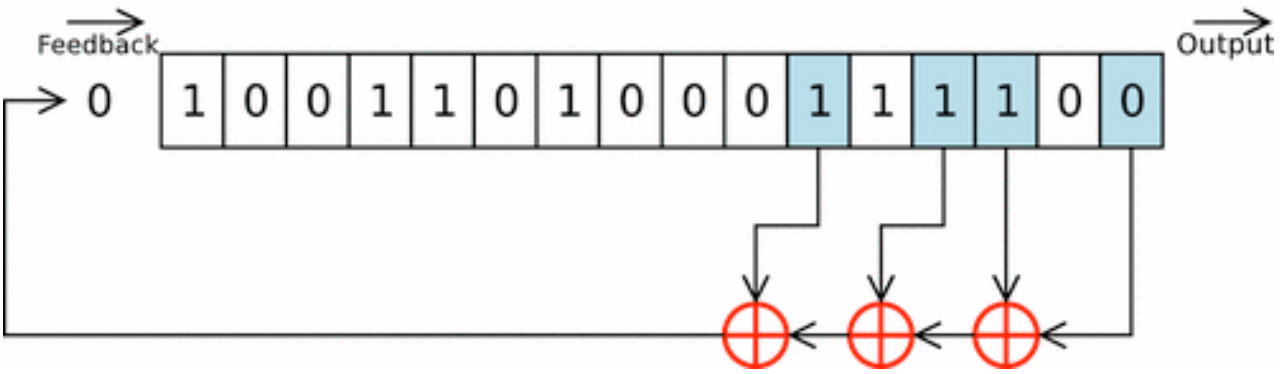

Feedback
Output
0 0 1 1 0 1 0 0 0 1 1 1 1 0 0 1
1 1 0 1
1 0 1

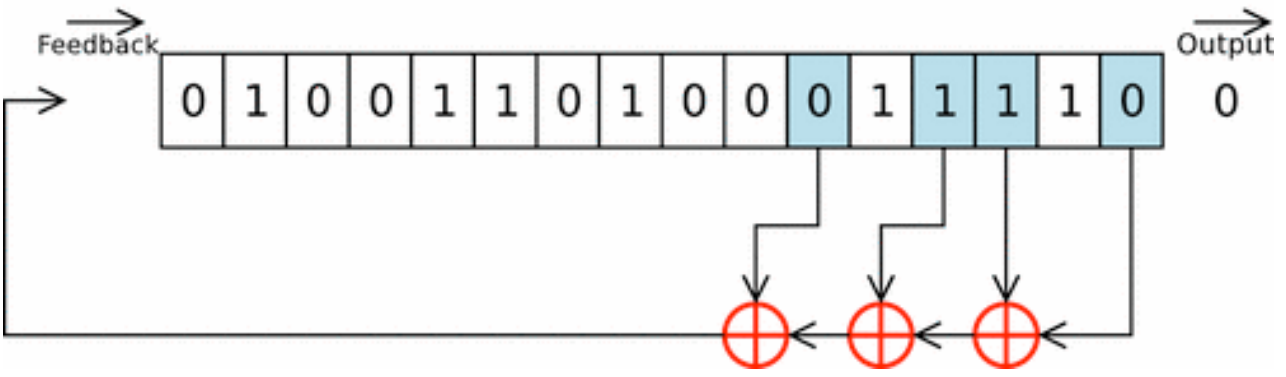

Feedback
Output
1
0 0 1 1 0 1 0 0 0 1 1 1 1 0 0 1

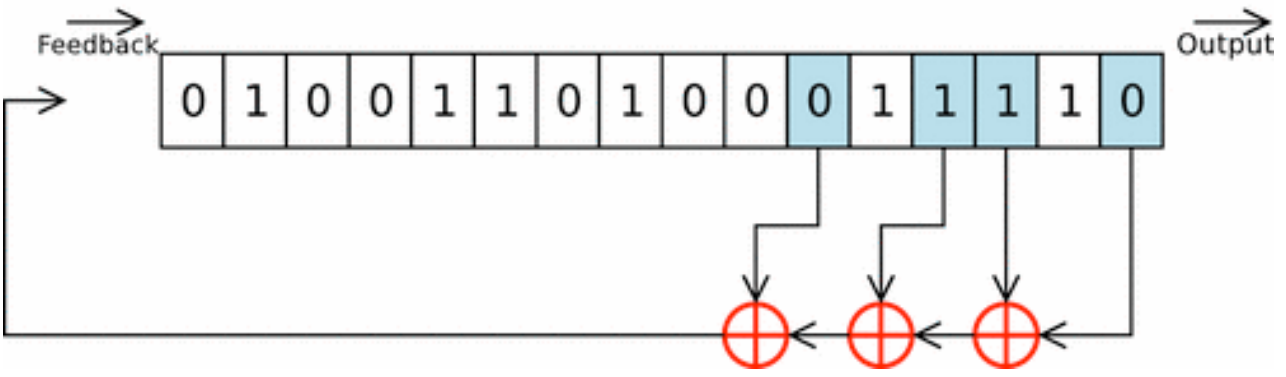

Feedback
Output
1 0 0 1 1 0 1 0 0 0 1 1 1 1 0 0
1

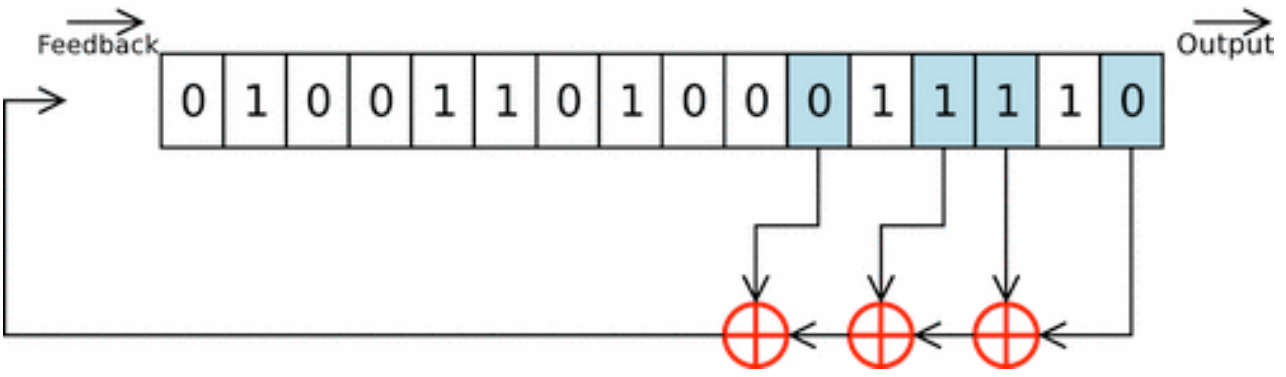

Feedback
Output
1 0 0 1 1 0 1 0 0 0 1 1 1 1 0 0


Feedback
Output
1 0 0 1 1 0 1 0 0 0 1 1 1 1 0 0
1
1
0
0
0
1
0


Feedback
Output
0
1 0 0 1 1 0 1 0 0 0 1 1 1 1 0 0


Feedback
0 1 0 0 1 1 0 1 0 0 0 1 1 1 1 0
Output
0


Feedback
Output
0 1 0 0 1 1 0 1 0 0 0 1 1 1 1 0


Feedback
Output
0 1 0 0 1 1 0 1 0 0 0 1 1 1 1 0

Fonctionnement d'un LFSR binaire de longueur L

A chaque top d'horloge, le bit de poids faible s_i constitue la sortie du registre, et les autres sont décalés vers la droite ; le nouveau bit s_{i+L} placé dans la cellule de poids fort du registre est donné par une fonction linéaire :

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

où les coefficients c_i sont binaires.

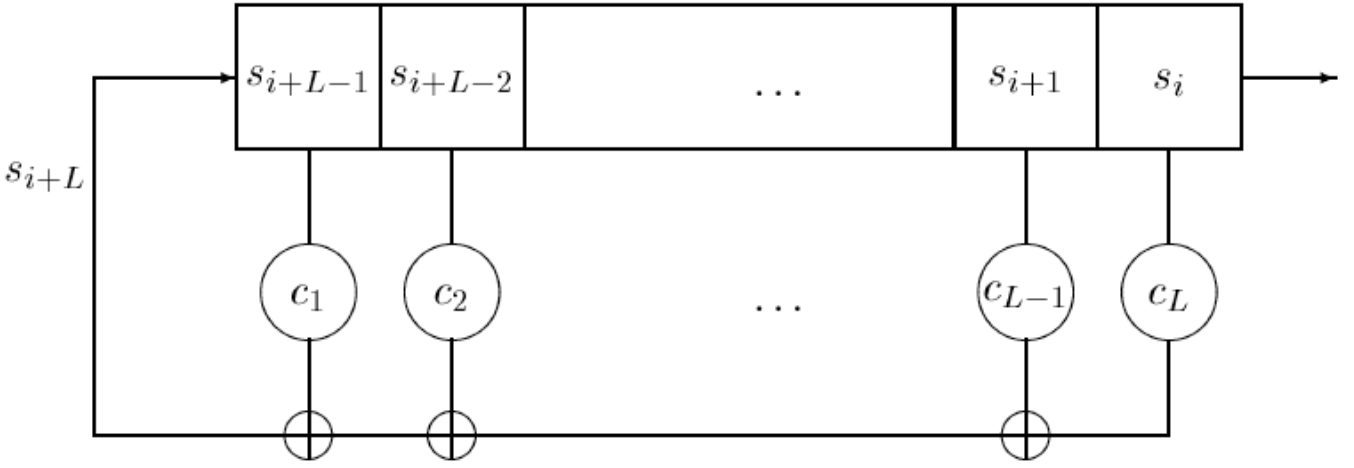


Figure 1 : Registre à décalage à rétroaction linéaire de longueur L.

Définition 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 a 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,** c'est-à-dire qu'il existe une pré-période n_0 telle que la suite $(s_n)_{n\geq n_0}$ est périodique.

Proposition 1 *La suite* s *est ultimement périodique, de période*

$$T \leq 2^L - 1$$

(i.e. il existe un entier i_0 *tel que* $s_i = s_{i+T}$ *pour tout* $i \geq i_0$*).*

Si, de plus, $c_L = 1$*, la suite* s *est périodique (i.e.* $s_i = s_{i+T}$ *pour tout* $i \geq 0$*).*

Démonstration –

Notons $R_i = (s_i, s_{i+1}, \ldots, s_{i+L-1})$ le i-ème registre. Celui-ci détermine complètement les registres ultérieurs. Ce registre peut prendre au plus 2^L états.

S'il atteint l'état $0 = (0, \ldots, 0)$ alors les registres successifs sont tous nuls et la suite elle-même est nulle à partir de là.

S'il n'est jamais nul, parmi $[R_0, R_1, \ldots, R_{2^L-1}]$, au moins deux registres sont identiques. Supposons $R_{i_0} = R_{i_0+T}$; alors la suite des registres $[R_{i_0}, R_{i_0+1}, \ldots, R_{i_0+T-1}]$ se répète indéfiniment.

On a donc $s_i = s_{i+T}$ pour tout $i \geq i_0$ avec $T \leq 2^L - 1$.

On peut interpréter aussi la relation entre deux registres successifs en termes matriciels. Notons

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 \\ c_L & c_{L-1} & \dots & c_3 & c_2 & c_1 \end{pmatrix}$$

En considérant les R_i comme des vecteurs colonnes, on a

$$R_{i+1} = AR_i.$$

Ainsi, on a $R_n = A^n R_0$.

Remarquons que le déterminant de A est égal à c_L. Ainsi, si $c_L = 1$, la matrice A est inversible et le LFSR ne passe jamais par le registre nul. La condition $R_{i_0} = R_{i_0+T}$ devient $A^{i_0} R_0 = A^{i_0+T} R_0$ mais comme A est inversible, on en déduit $R_0 = A^T R_0 = R_T$ et la suite s est périodique.

Complexité linéaire des LFSRs.

Associons à la suite $(s_n)_{n\geq 0}$ la série génératrice

$$s(X) = \sum_{n\geq 0} s_n X^n.$$

Définition 3 *Soit un LFSR dont la fonction de rétroaction est donnée par la relation*

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

Son **polynôme de rétroaction** f *est le polynôme de* $\mathbf{F}_2[X]$*:*

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

On peut alors écrire le développement en série formelle de la suite $(s_n)_{n\geq 0}$ sous forme d'une fraction rationnelle.

Proposition 2 *La suite $(s_n)_{n\geq 0}$ est produite par un LFSR dont le polynôme de rétroaction est $f(X) = 1 + c_1X + c_2X^2 + \cdots + c_LX^L$ si et seulement si son développement en série formelle $s(X) = \sum_{n\geq 0} s_nX^n$ s'écrit*

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

où g est un polynôme de $F_2[X]$ tel que $\deg(g) < \deg(f)$. En outre, le polynôme g est entièrement déterminé par l'état initial du registre :

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

Démonstration –

On a: $g(X) = s(X)f(X)$. Soit, pour tout $i \geq 0$ $g_i = \sum_{j=0}^{L} s_{i-j}c_j.$

Les formules qui se déduisent de cette équation impliquent que g est un polynôme à partir du rang L.

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 $(s_n)_{n\geq 0}$.

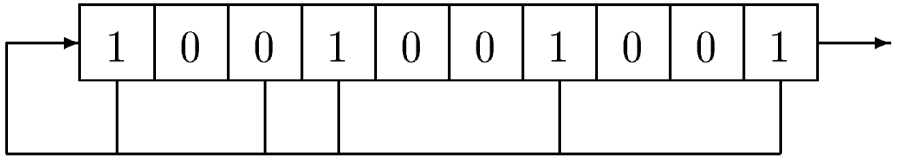
Définition 4 *Soit $(s_n)_{n\geq 0}$ une suite binaire à rétroaction linéaire d'ordre L dont l'état initial est non nul. Son polynôme de rétroaction minimal est l'unique polynôme unitaire f_0 de $F_2[X]$ tel qu'il existe $g_0 \in F_2[X]$, avec $\deg(g_0) < \deg(f_0)$ et $pgcd(g_0, f_0) = 1$, vérifiant*

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

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 Soit $(s_n)_{n\geq 0}$ la suite produite par le LFSR suivant :



Son polynôme de rétroaction est :

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

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 :

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

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.

Proposition 3 *Si le polynôme de rétroaction minimal du LFSR est* **primitif** *et que son état initial est non nul, alors la suite produite* $(s_n)_{n\geq 0}$ *est de* **période maximale** $2^{\Lambda(s)}-1$ *et est dite suite de longueur maximale.*

Démonstration

Soit f le polynôme de rétroaction minimal du LFSR. Un calcul simple montre que le polynôme caractéristique de la matrice $A = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 1 \\ c_L & c_{L-1} & \dots & c_3 & c_2 & c_1 \end{pmatrix}$ est $X^L f(X^{-1}) = \tilde{f}(X)$.

Les valeurs propres de A sont les racines de $\tilde{f}$. Elles sont toutes distinctes, et d'ordre 2^L-1, puisqu'elles sont génératrices du groupe cyclique $\mathbf{F}_{2^L}$.

Donc A est d'ordre 2^L-1, et c'est la période de la suite $(s_n)_{n\geq 0}$.

Critères statistiques

On demande à une suite pseudo-aléatoire, périodique de période N, de satisfaire certaines des propriétés des suites authentiquement aléatoires.

Parmi ces propriétés, les plus classiques sont les trois critères de Golomb, proposés par celui-ci en 1982.

Bien sûr, ces conditions ne garantissent en aucun cas la sécurité cryptographique!

1. Dans chaque période, le nombre de 0 est approximativement égal au nombre de 1:

$$\left| \sum_{i=0}^{N-1} (-1)^{s_i} \right| \leq 1.$$

2. Une série (de 0 ou de 1) est une succession de bits identiques, maximale (i.e. encadrée par des bits opposés). Dans chaque période, soit S l'ensemble des séries; si $2^k \leq |S| < 2^{k+1}$, on trouve $|S|/2$ séries de longueur 1, $|S|/4$ séries de longueur 2, ..., $|S|/2^k$ séries de longueur k, et pour chaque longueur, autant de séries de 0 que de séries de 1.

3. La fonction d'auto-corrélation prend deux valeurs, suivant que $\tau = 0$ ou non:

$$C(\tau) := \sum_{i=0}^{N-1} (-1)^{s_i + s_{i+\tau}}.$$

Proposition 4 *Dans une suite de longueur maximale* $2^L - 1$*, toute suite* $(s_i, s_{i+1}, \ldots, s_{i+L-1})$ *de* L *éléments non tous nuls apparaît une et une seule fois par période.*

Démonstration

Un registre $R_i = (s_i, s_{i+1}, \ldots, s_{i+L-1})$ ne peut apparaître qu'une fois par période. Or il y a $2^L - 1$ registres par période et les registres prennent au plus $2^L - 1$ valeurs. Donc ils prennent toutes les valeurs une fois.

Corollaire 1 *Le premier critère de Golomb est vérifié.*

Démonstration

Tout registre $R_i = (s_i, s_{i+1}, \ldots, s_{i+L-1})$ apparaît une et une seule fois. Son premier élément s_i prendra donc 2^{L-1} fois la valeur 1, et $2^{L-1} - 1$ fois la valeur 0, car il ne peut pas prendre la valeur $(0, \ldots, 0)$.

Corollaire 2 *Le deuxième critère de Golomb est vérifié.*

Démonstration

Comptons le nombre de fois qu'apparaît une suite de exactement k zéros. Cela revient à compter le nombre de fois qu'apparaît une suite formée de $(1, \underbrace{0, \ldots, 0}_{k}, 1)$ au début d'un registre. Comme tous les registres $R_i = (s_i, s_{i+1}, \ldots, s_{i+L-1})$ apparaissent une et une seule fois, un registre composé de

$$(1, \underbrace{0, \ldots, 0}_{k}, 1, s_{i+k+2}, \ldots, s_{i+L-1})$$

apparaît 2^{L-k-2} fois.

Corollaire 3 *Le troisième critère de Golomb est vérifié.*

Démonstration

La suite $(s_i)_{0\leq i\leq N-1}$ est obtenue par une relation de récurrence **linéaire** liée au registre à décalage donné. La suite $(s_{i+\tau})_{0\leq i\leq N-1}$ est aussi obtenue par une relation de récurrence **linéaire** liée au même registre à décalage (obtenue après un décalage de τ).

La relation de récurrence étant **linéaire**, la suite $(s_i + s_{i+\tau})_{0\leq i\leq N-1}$ est encore donnée par le même registre à décalage. Il existe donc une suite $(t_i)_{0\leq i\leq N-1}$ (toujours donnée par le même registre à décalage) telle que

$$t_i = s_i + s_{i+\tau}$$

pour $0 \leq i \leq N-1$. D'après le premier corollaire, on a donc, si $\tau \neq 0$

$$C(\tau) := \sum_{i=0}^{N-1} (-1)^{s_i+s_{i+\tau}} = \sum_{i=0}^{N-1} (-1)^{t_i} = -1.$$

L'algorithme de Berlekamp-Massey

On supposera pour simplifier que **la suite produite par le LFSR est une suite de longueur maximale.**

Un tel générateur n'est pas cryptographiquement sûr, à cause de l'algorithme de Berlekamp-Massey, qui calcule (en temps quadratique) la complexité linéaire et un polynôme engendrant une suite finie.

Le lemme suivant montre qu'il suffit de connaitre $2L$ bits consécutifs de la suite pour la retrouver entièrement.

Lemme 1 *Soit s une suite récurrente de longueur maximale, de complexité linéaire L. Si on connait $2L$ termes consécutifs de cette suite, alors on peut calculer les coefficients de récurrence en inversant un système linéaire de taille $L \times L$.*

Démonstration. — On a la relation linéaire:

$$\begin{pmatrix} s_i & s_{i+1} & s_{i+2}\cdots & s_{i+L-1} & \\ s_{i+1} & s_{i+2} & s_{i+3}\cdots & s_{i+L} & \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ s_{i+L-2} & \vdots & \cdots\cdots & s_{i+2L-3} & \\ s_{i+L-1} & \vdots & \cdots\cdots & s_{i+2L-2} & \end{pmatrix} \begin{pmatrix} c_L \\ c_{L-1} \\ \vdots \\ c_2 \\ c_1 \end{pmatrix} = \begin{pmatrix} s_{i+L} \\ s_{i+L+1} \\ \vdots \\ s_{i+2L-2} \\ s_{i+2L-1} \end{pmatrix}$$

Il suffit de montrer que la matrice S de ce système est inversible. Ses colonnes sont les vecteurs associés aux registres $R_i,\ldots,R_{i+L-1}$. Une combinaison linéaire nulle non triviale équivaut à

$$a_i R_i + \cdots + a_{i+L-1} R_{i+L-1} = 0$$

donc à

$$(a_i + \cdots + a_{i+L-1} A^{L-1}) R_i = 0$$

donc à l'existence d'un polynôme P non nul de degré au plus égal à $L-1$ tel que $P(A)R_i = 0$. Par hypothèse, le polynôme caractéristique de A, égal au polynôme réciproque du polynôme de rétroaction, est irréductible de degré L, donc premier à P, donc $P(A)$ est inversible.

Théorème 1 *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.

Aussi s'attache-t-on généralement à utiliser des LFSRs dont la **complexité linéaire** est égale à leur **longueur**. C'est en particulier possible si **le polynôme de rétroaction f est irréductible;** on est alors assuré qu'il est impossible d'engendrer $(s_n)_{n\geq 0}$ à l'aide d'un LFSR plus court.

5.3 Description de l'algorithme

Dans la pratique cryptographique, l'attaquant connait seulement un morceau de la suite s, et il essaie de deviner les bits suivants. En particulier, il peut chercher si ce bout de suite est le début d'un LFSR.

Dans cette direction, on étend la notion de complexité linéaire aux suites finies:

Définition 5 *Etant donnée une suite* $s = (s_0, \ldots, s_{n-1})$*, et un polynôme*

$$f = 1 + c_1 x + \cdots + c_L x_L$$

(maintenant f n'est plus nécessairement de degré exactement égal à L), *on dit que* (L, f) **engendre** s *si*

$$s_j = \sum_{i=1}^{L} c_i s_{j-i} \quad \textit{pour tout} \quad j = 0, \ldots, n-1.$$

Le plus petit L *convenable s'appelle* **la complexité linéaire de** s *et se note* $\Lambda(s)$.

L'algorithme utilise le lemme suivant

Lemme 2 *Soit L_n la longueur minimale d'un LFSR qui engendre les bits $s_0, s_1, \ldots, s_{n-1}$ mais qui n'engendre pas $s_0, s_1, \ldots, s_n$. Alors la longueur minimale L_{n+1} d'un LFSR engendrant $s_0, s_1, \ldots, s_{n-1}$ vérifie*

$$L_{n+1} \geq \max(n+1-L_n, L_n)$$

L'algorithme de Belekamp-Massey permet de calculer un (L, f) associé à s, avec $L = \Lambda(s)$. En fait l'algorithme calcule un couple (L, f) avec L minimal, successivement pour les suites tronquées $s^{(1)}$, $s^{(2)}, \ldots$, $s^{(k)}, \ldots$, $s = s^{(n)}$, où $s^{(i)} = (s_0, \ldots, s_{i-1})$.

Le temps de calcul de cet algorithme est en n^2.

On peut maintenant décrire l'algorithme.

Théorème 2 *Supposons pour* $k \geq 1$ *avoir calculé un couple* (L, f) *associé à* $s^{(k)}$ *avec* $L = \Lambda(s^{(k)})$. *Soit*

$$d_k := s_k + \sum_{i=0}^{L-1} c_i s_{k-L+i}.$$

• *Si* $d_k = 0$, *alors* $\Lambda(s^{(k+1)}) = L$ *et* (L, f) *engendre* $s^{(k+1)}$.

• *Si* $d_k = 1$, *alors soit* m *le plus grand entier tel que* $m < k$ *et* $\Lambda(s^{(m)}) < L$, *et soit* g *tel que* $(\Lambda(s^{(m)}), g)$ *engendre* $s^{(m)}$. *Alors*

– $\Lambda(s^{(k+1)}) = \max(L, k+1-L)$

– $s^{(k+1)}$ *est engendré par* $f + x^{k-m} g$.

Exemple. Appliquons l'algorithme de Berlekamp-Massey à la suite

$$0110010101$$

de longueur 10. Voici les résultats fournis par l'algorithme :

k	s_k	d_k	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	$1+X^2+X^3$

Le polynôme de rétroaction est donc $1+X^4+X^5$.

Combinaison de plusieurs registres

Présentation

Même lorsque le polynôme de rétroaction du registre est choisi de manière appropriée, **la complexité linéaire de la suite** produite reste généralement **trop faible** pour se prémunir d'une attaque par l'algorithme de Berlekamp-Massey.

Afin de **surmonter cet obstacle** et **d'augmenter la taille de l'espace des clefs**, c'est-à-dire le nombre d'initialisations possibles, on utilise m LFSRs en parallèle, et on combine leurs sorties par une fonction booléenne

$$f : \mathbf{F}_2^m \longrightarrow \mathbf{F}_2$$

appelée fonction de combinaison.

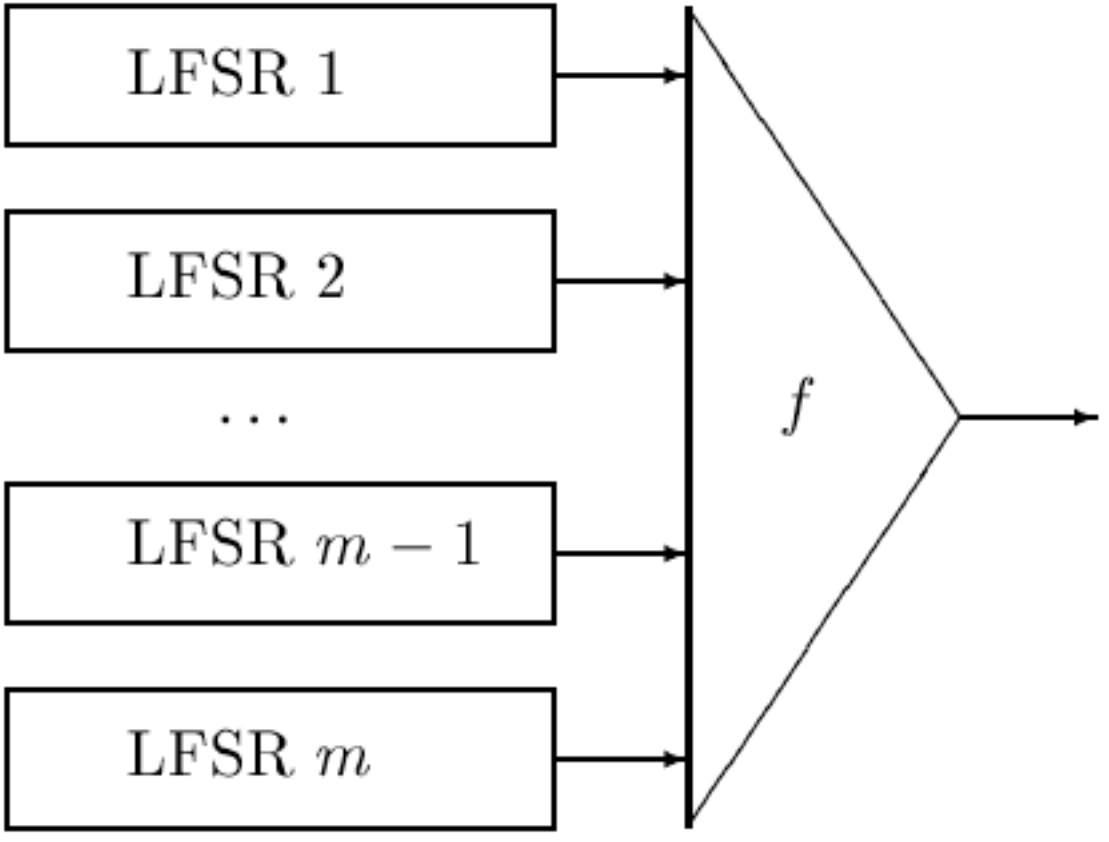


Figure 1 : Combinaison de LFSRs par une fonction booléenne

Définitions

Définition 6 *Une* **fonction booléenne** *à* m *variables est une application de* $\mathbf{F}_2^m$ *dans* $\mathbf{F}_2$*.*

On appelle fonction booléenne vectorielle à m *variables et* n *composantes une application de* $\mathbf{F}_2^m$ *dans* $\mathbf{F}_2^n$*: c'est la juxtaposition de* n *fonctions booléennes à* m *variables.*

Définition 7 *On appelle* **support** *de la fonction booléenne* f *l'ensemble des éléments* u *tels que* $f(u) \neq 0$*, et on le note* $\mathrm{Supp}(f)$*.*

On appelle **poids de** f *le cardinal de son support, et on le note* $wt(f)$*.*

On définit également pour deux fonctions f *et g la* **distance** *qui les sépare par*

$$d(f,g) = wt(f+g).$$

Et de manière générale, à tout vecteur binaire u on associe un poids $wt(u)$ qui est égal au nombre de ses composantes non nulles.

Une fonction booléenne est entièrement caractérisée par sa **table de vérité,** c'est-à-dire la liste de tous les éléments de $\mathbf{F}_2^m$ avec les valeurs qu'elle prend en chacun d'eux.

Exemple

Considérons le cas $m = 3$. Voici la définition d'une fonction f par sa table de vérité :

éléments de $\mathbf{F}_2^3$	valeur de f
$(0,0,0)$	0
$(0,0,1)$	1
$(0,1,0)$	1
$(1,0,0)$	0
$(0,1,1)$	0
$(1,0,1)$	0
$(1,1,0)$	1
$(1,1,1)$	1

On a ici $\mathrm{Supp}(f) = \{(0,0,1),(0,1,0),(1,1,0),(1,1,1)\}$ et $wt(f) = 4$.

Forme Algébrique Normale d'une fonction booléenne f.

Définition 8 *La* **Forme Algébrique Normale** *d'une fonction booléenne* f *à* m *variables est l'unique polynôme* Q_f *de* $\mathbf{F}_2[x_1, \ldots, x_m]/(x_1^2 + x_1, \ldots, x_m^2 + x_m)$ *tel que*

$$f(u_1, \ldots, u_m) = Q_f(u_1, \ldots, u_m)$$

pour tout $(u_1, \ldots, u_m) \in \mathbf{F}_2^m$.

Remarque 1 *Un polynôme Q_f de $\mathbf{F}_2[x_1,\ldots,x_m]/(x_1^2+x_1,\ldots,x_m^2+x_m)$ est souvent représenté par un polynôme de $\mathbf{F}_2[x_1,\ldots,x_m]$ de degrés partiels au plus 1 en les x_i : $\deg_{x_i} \leq 1$.*

Proposition 5 *L'application qui associe à un polynôme de $\mathbf{F}_2[x_1,\ldots,x_m]$ de degrés partiels ≤ 1 son image dans $\mathbf{F}_2[x_1,\ldots,x_m]/(x_1^2+x_1,\ldots,x_m^2+x_m)$ est un isomorphisme d'espaces vectoriel.*

Démonstration –

L'application est **injective**, car un polynôme de degrés partiels ≤ 1 ne peut être réduit par des polynômes $x_i^2+x_i$ de degré partiels égal à 2.

L'application est **surjective**, car un polynôme $\sum_{i_1,\ldots,i_m} a_{i_1,\ldots,i_m} x_1^{i_1}\ldots x_m^{i_m}$ est congru à $\sum_{i_1,\ldots,i_m} a_{i_1,\ldots,i_m} x_1^{j_1}\ldots x_m^{j_m}$ modulo $x_1^2+x_1,\ldots,x_m^2+x_m$

où $j_k = 0$ si $i_k = 0$, $j_k = 1$ sinon.

Définition 9 *Soit f une fonction booléenne à m variables. On définit sa transformée de Möbius comme la fonction*

$$\begin{aligned} f^\circ : \mathbf{F}_2^m &\longrightarrow \mathbf{F}_2 \\ u &\longmapsto \sum_{v \leq u} f(v) \pmod 2 \end{aligned}$$

avec $v \leq u$ si et seulement si pour tout i, $v_i = 1 \Rightarrow u_i = 1$.

Ou encore $v \leq u$ si et seulement si pour tout i, $u_i = 0 \Rightarrow v_i = 0$.

Proposition 6 *La Forme Algébrique Normale d'une fonction booléenne f à m variables est égale à*

$$\sum_{u=(u_1,\ldots,u_m)\in \mathbf{F}_2^m} f^\circ(u) x_1^{u_1} \ldots x_m^{u_m}$$

Démonstration :

On le démontre par récurrence sur m. Les coefficients de la Forme Algébrique Normale de f étant binaires, les sommes constituant ces coefficients sont considérées modulo 2.

1) $m = 1$: on a

$$f(x_1) = f(0)(1 + x_1) + f(1)x_1,$$

et donc

$$f(x_1) = (f(0) + f(1))x_1 + f(0).$$

2) récurrence : par hypothèse, on sait que pour tout $a \in \mathbf{F}_2$ fixé, on a :

$$f(x_1, \ldots, x_{m-1}, a) = \sum_{u \in \mathbf{F}_2^{m-1}} \sum_{v \leq u} f(v_1, \ldots, v_{m-1}, a) x_1^{u_1} \ldots x_{m-1}^{u_{m-1}}$$

Or pour toutes les valeurs de $(v_1, \ldots, v_{m-1}) \in \mathbf{F}_2^{m-1}$ on a

$$\begin{aligned} & f(v_1, \ldots, v_{m-1}, x_m) \\ = \; & f(v_1, \ldots, v_{m-1}, 1) x_m + f(v_1, \ldots, v_{m-1}, 0)(x_m + 1) \\ = \; & (f(v_1, \ldots, v_{m-1}, 0) + f(v_1, \ldots, v_{m-1}, 1)) x_m + f(v_1, \ldots, v_{m-1}, 0) \end{aligned}$$

On a donc bien

$$f(x_1, \ldots, x_m) = \sum_{u \in \mathbf{F}_2^m} \sum_{v \leq u} f(v_1, \ldots, v_m) x_1^{u_1} \ldots x_m^{u_m}$$

Proposition 7 *L'application qui associe à un polynôme Q de $\mathbf{F}_2[x_1, \ldots, x_m]$ de degrés partiels ≤ 1 la fonction booléenne $(u_1, \ldots, u_m) \longmapsto Q(u_1, \ldots, u_m)$ est un isomorphisme d'espaces vectoriels.*

Démonstration –

Cette application est **surjective** puisqu'on a trouvé dans la proposition précédente une Forme Algébrique Normale d'une fonction f.

L'espace vectoriel de ces polynômes a pour base la famille des monômes de degré partiels inférieurs à 1, qui a 2^m éléments. Donc l'ensemble des polynômes Q tels que $\deg_{x_i} Q \leq 1$ a 2^{2^m} éléments.

L'espace des fonctions de $\mathbf{F}_2^m$ dans $\mathbf{F}_2$ a 2^{2^m} éléments.

L'application est surjective, envoie un ensemble dans un ensemble ayant même nombre d'élément: elle est donc bijective.

Dans la suite une fonction booléenne f sera souvent confondue avec sa Forme Algébrique Normale Q_f.

Définition 10 *On appelle degré de f, et on note $\deg(f)$, le degré du polynôme Q_f . Une fonction de degré 1 est dite* **affine**, *et si de plus elle est nulle en 0, elle est* **linéaire**.

Pour combiner des LFSRs en vue d'un chiffrement à flot, on utilise dans l'idéal des fonctions **équilibrées** , i.e. dont la sortie est uniformément distribuée (ces fonctions prennent autant de fois la valeur 0 que la valeur 1), de manière à ce que la suite produite ne soit pas biaisée; on peut éventuellement utiliser une fonction qui ne soit pas tout à fait équilibrée, pour peu que le biais reste inexploitable.

La suite obtenue par combinaison de LFSRs étant également une suite à récurrence linéaire, il est indispensable d'estimer sa **complexité linéaire.**

Puisque f peut être assimilée à un polynôme, la **suite produite** par un générateur sera construite à partir des suites engendrées par les m LFSRs à l'aide de **sommes et de produits terme-à-terme.** Il convient donc d'étudier la complexité linéaire d'une suite résultant de la somme ou du produit d'autres suites.

Proposition 8 *Soient* $(u_n)_{n\geq 0}$ *et* $(v_n)_{n\geq 0}$ *deux suites à récurrence linéaire de polynômes de rétroaction minimaux respectifs* f_u *et* f_v. *Alors leur somme est une* **suite à récurrence linéaire.** *La complexité linéaire de leur somme vérifie*

$$\Lambda(u+v) \leq \Lambda(u) + \Lambda(v)$$

avec égalité si et seulement si $pgcd(f_u, f_v) = 1$.

De plus, dans le cas de l'égalité, la période de leur somme est égale au $ppcm$ *des périodes de* $(u_n)_{n\geq 0}$ *et* $(v_n)_{n\geq 0}$.

Démonstration

La complexité linéaire de la suite $(u_n)_{n\geq 0}$ est le degré de son polynôme de rétroaction minimal, c'est-à-dire de l'unique polynôme unitaire f_u de $F_2[X]$ tel que

il existe $g_u \in F_2[X]$, avec $\deg(g_u) < \deg(f_u)$ et $pgcd(g_u, f_u) = 1$, vérifiant $u(X) = \sum u_n X_n = \frac{g_u(X)}{f_u(X)}$.

De même pour la suite $(v_n)_{n\geq 0}$, et $v(X) = \sum v_n X_n = \frac{g_v(X)}{f_v(X)}$.

Formons $u + v$:

$$\begin{aligned}(u+v)(X) = \sum (u_n + v_n) X_n &= \frac{g_u(X)}{f_u(X)} + \frac{g_v(X)}{f_v(X)} \\ &= \frac{g_u(X) f_v(X) + g_v(X) f_u(X)}{f_u(X) f_v(X)} = \frac{g_{u+v}(X)}{f_{u+v}(X)}\end{aligned}$$

où la dernière fraction est irréductible. On a donc

$$\Lambda(u+v) = \deg f_{u+v} \leq f_u(X) f_v(X) = \Lambda(u) + \Lambda(v)$$

Proposition 9 *Soient* $(u_n)_{n\geq 0}$ *et* $(v_n)_{n\geq 0}$ *deux suites à récurrence linéaire de polynômes de rétroaction minimaux respectifs* f_u *et* f_v*. Alors leur produit est une suite à récurrence linéaire.*

La complexité linéaire de leur produit vérifie

$$\Lambda(uv) \leq \Lambda(u)\Lambda(v)$$

avec égalité si les polynômes f_u *et* f_v *sont primitifs, et si*

$$PGCD(\Lambda(u), \Lambda(v)) = 1.$$

Dans ce cas, la période de $(uv_n)_{n\geq 0}$ *est égale au produit des périodes de* $(u_n)_{n\geq 0}$ *et* $(v_n)_{n\geq 0}$.

Démonstration

On a $u(X) = \sum u_n X_n = \frac{g_u(X)}{f_u(X)}$ et $(v_n)_{n\geq 0}$, et $v(X) = \sum v_n X_n = \frac{g_v(X)}{f_v(X)}$.

Formons uv:

$$(uv)(X) = \sum (u_n v_n) X_n$$

Une fraction rationnelle $h \in \overline{\mathbf{F}_2}(X)$ se met de manière unique sous la forme

$$h(X) = B(X) + \sum_i \frac{A_i(X)}{(X-\alpha_i)^{n_i}} \tag{1}$$

où $A_i(X)$ et $B(X)$ sont des polynômes et $\deg A_i < n_i$.

Supposons que les polynômes f_u et f_v soient sans racines multiples. Alors il existe des éléments $\mu_1, \ldots, \mu_{L_u}$ de $\mathbf{F}_{2^{L_u}}$ tels que

$$\frac{g_u(X)}{f_u(X)} = \sum_{i=1}^{L_u} \frac{\mu_i}{\alpha_i X - 1}$$

d'où

$$u(X) = \sum_n u_n X_n = \frac{g_u(X)}{f_u(X)} = \sum_{i=1}^{L_u} \mu_i \sum_n (\alpha_i X)^n = \sum_n X^n \sum_{i=1}^{L_u} \mu_i (\alpha_i)^n$$

d'où

$$u_n = \sum_{i=1}^{L_u} \mu_i (\alpha_i)^n$$

De même il existe des éléments $\nu_1, \ldots, \nu_{L_v}$ de $\mathbf{F}_{2^{L_v}}$ tels que

$$v_n = \sum_{i=1}^{L_u} \nu_i (\beta)^n$$

où les $\beta_1, \ldots, \beta_{L_v}$ sont les racines inverses de f_v.

Donc on a $u_n v_n = \sum_{i,j} \mu_i \nu_j (\alpha_i \beta_j)^n$ et

$$uv(X) = \sum_n u_n v_n X^n = \sum_n \sum_{i,j} \mu_i \nu_j (\alpha_i \beta_j X)^n = \sum_{i,j} \mu_i \nu_j \sum_n (\alpha_i \beta_j X)^n$$

$$= \sum_{i,j} \frac{\mu_i \nu_j}{1 - \alpha_i \beta_j X} = \frac{g_{uv}(X)}{f_{uv}(X)}$$

avec $f_{uv}(X) = \prod_{i,j} (1 - \alpha_i \beta_j X)$. Ce polynôme est invariant par la transformation $x \longmapsto x^2$. Par conséquent il appartient à $\mathbf{F}_2[X]$. On a donc

$$\Lambda(uv) \leq \deg(f_{uv}) = L_u L_v = \Lambda(u)\Lambda(v)$$

Pour voir si $\Lambda(uv) = L_u L_v$, il faut vérifier que $\frac{g_{uv}(X)}{f_{uv}(X)}$ est irréductible. Comme

$$\sum_{i,j} \frac{\mu_i \nu_j}{1 - \alpha_i \beta_j X} = \frac{g_{uv}(X)}{f_{uv}(X)},$$

cela revient à dire que

– $\mu_i \nu_j \neq 0$;

– les $\alpha_i \beta_j$ sont tous distincts.

Les polynômes f_u et f_v étant des polynômes de rétroaction minimaux, on a $\mu_i \neq 0$ et $\nu_i \neq 0$. Donc $\mu_i \nu_j \neq 0$.

Si les $\alpha_i \beta_j$ ne sont pas tous distincts, il existe i et j tels que $\alpha_i = \beta_j \neq 1$. On a $\alpha_i \in \mathbf{F}_{2^{L_u}}$ et $\beta_j \in \mathbf{F}_{2^{L_v}}$, donc il existe un sous-corps commun à $\mathbf{F}_{2^{L_u}}$ et à $\mathbf{F}_{2^{L_v}}$, ce qui veut dire que $PGCD(L_u, L_v) \neq 1$.

Théorème 3 *Soient m LFSRs binaires dont les polynômes de rétroaction sont* **primitifs** *et de degrés $L_1, \ldots, L_m$ deux-à-deux* **premiers entre eux.** *Alors la complexité linéaire de la suite produite en combinant ces LFSRs par la fonction booléenne f à m variables est égale à*

$$L = f(L_1, \ldots, L_m)$$

où f est vue comme un polynôme de $\mathbf{Z}[x_1, x_2, \ldots, x_m]$ et est évaluée sur les entiers.

On en déduit donc qu'une fonction de combinaison doit non seulement être équilibrée, mais aussi avoir **un degré le plus élevé possible**.

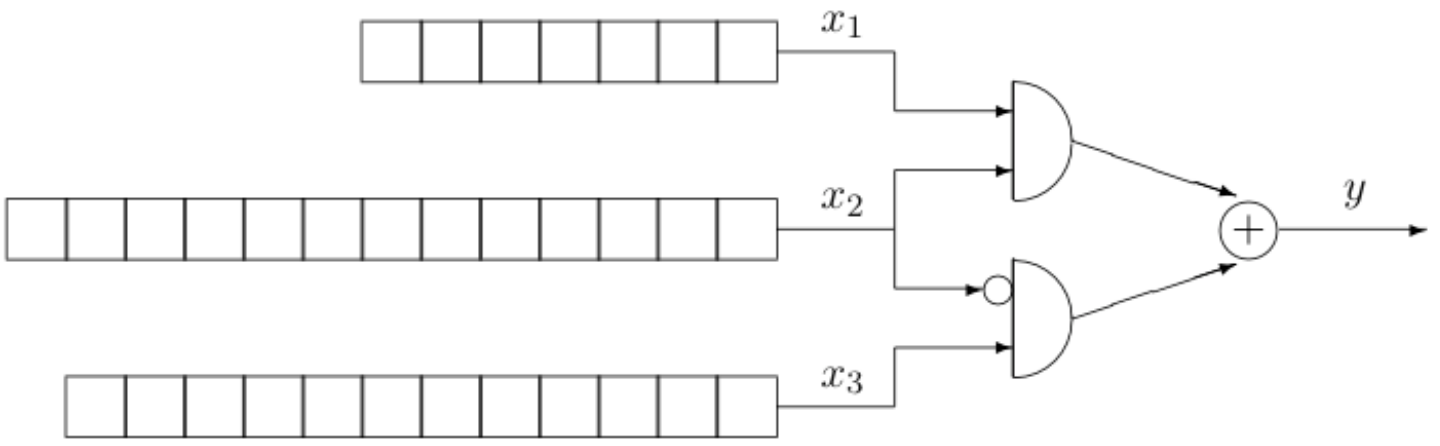
Exemple

Le générateur de Geffe (1973)

Le générateur de Geffe est défini par 3 LFSRs dont les polynômes de rétroaction sont primitifs et de degrés L_1, L_2 et L_3 deux-à-deux premiers entre eux. Ces registres sont assemblés par la fonction booléenne

$$f(x_1, x_2, x_3) = x_1 x_2 + x_2 x_3 + x_3$$

La complexité de la suite produite par ce générateur est donc $L_1 L_2 + L_2 L_3 + L_3$.



Le générateur de Geffe

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

Ce générateur de Geffe a deux avantages importants.

D'une part, son rendement comporte une distribution moyenne égale à 0 et à 1, qui est très rarement le cas quand des opérations de logique du type de produit sont présentées.

D'autre part, pour décoder ce dispositif sans savoir la clef, il serait nécessaire de résoudre un système de n_d équations, avec $n_d = n_a + n_b + n_c$ (n_a, n_b, n_c indiquant le nombre de registre de A, B et C respectivement), mais ces équations sont non-linéaires et la solution du système est difficile.

Attaque par corrélation de Siegenthaler

Cette attaque, développée par T. Siegenthaler est de type "diviser pour mieux régner" : elle consiste à retrouver l'initialisation de chacun des registres indépendemment des autres.

Pour cela, on exploite l'existence d'une **éventuelle corrélation entre la sortie de la fonction de combinaison f et l'une de ses entrées**.

On va regarder un exemple sur le générateur de Geffe.

Problème: pour une suite de sortie $\mathbf{s(t)}$ donnée, retrouver la clef qui a permis de la produire.

$$f(x_1, x_2, x_3) = x_1 x_2 + (1 + x_2) x_3 = x_1 x_2 + x_2 x_3 + x_3 \pmod 2$$

- Probabilité de corrélation de la suite $x_1(t)$ à la suite $s(t)$:

$$\begin{aligned} P(s(t) = x_1(t)) &= P(x_2(t) = 1) + P(x_2(t) = 0) \; P(x_3(t) = x_1(t)) \\ &= \frac{1}{2} + \frac{1}{2} \; \frac{1}{2} \\ &= \frac{3}{4} \end{aligned}$$

- De même, $P(s(t) = x_3(t)) = 3/4$.

- On essaye toutes les initialisations du 1er registre jusqu'à ce que le nombre de coïncidences entre la suite de sortie $\mathbf{s(t)}$ et la sortie $\mathbf{x_1(t)}$ de R_1 soit $\simeq$ probabilité p de corrélation.

Trouver l'état initial de R_1 prendra 2^{L_1} essais.

- En répétant l'opération pour les deux autres registres, l'état initial de chaque LFSR peut être déterminé en environ

$$2^{L_1} + 2^{L_2} + 2^{L_3} \text{ essais}$$

Ce nombre est bien plus petit que le nombre de différentes clefs qui est environ

$$2^{L_1+L_2+L_3}.$$

Théorème 4 *Soit f une fonction booléenne de combinaison à m variables. Pour $1 \leq i \leq m$, on note*

$$p_i = Pr[f(X_1, \ldots, X_m) = X_i]$$

où les X_i sont m variables aléatoires indépendantes uniformément distribuées dans $\mathbf{F}_2$. Soient $c_1, \ldots, c_N$ N bits de la suite chiffrante. Alors la corrélation α_i entre le chiffré et la sortie du i-ème LFSR, définie par

$$\alpha_i = \sum_{j=1}^{N} (-1)^{c_j} (-1)^{s_j^i}$$

est une variable aléatoire de moyenne m_i et de variance σ_i^2, avec

$$m_i = N(2p_i - 1) \quad \text{et} \quad \sigma_i^2 = 4Np_i(1 - p_i)$$

De même, la corrélation α_0 entre la suite chiffrante et une suite aléatoire s_0 indépendante de $s^1, \ldots, s^m$ est une variable aléatoire de moyenne m_0 et de variance σ_0^2, avec

$$m_0 = 0 \quad \text{et} \quad \sigma_0^2 = N.$$

On suppose fixé i. On le supprimera des formules.

On a

$$\alpha_i = \sum_{j=1}^{N} (-1)^{c_j} (-1)^{s_j^i} = \sum_{j=1}^{N} (-1)^{c_j + s_j}$$

Donc l'espérance de α_i est donnée par

$$E(\alpha) = \sum_{j=1}^{N} E\left((-1)^{c_j + s_j}\right)$$

On a

$$P(c_j + s_j = 0) = P(c_j = s_j) = P(f(X_1, \ldots, X_m) = X_i) = p_i$$

D'où

$$E\left((-1)^{c_j + s_j}\right) = 1P(c_j + s_j = 0) + (-1)P(c_j + s_j = \pm 1) = p - (1-p) = 2p - 1$$

et

$$E(\alpha) = N(2p - 1)$$

D'autre part

$$\begin{aligned} E(\alpha^2) &= E\left(\Big(\sum_{j=1}^{N} (-1)^{c_j+s_j} \Big)^2 \right) \\ &= \sum_{j=1}^{N} E\left(\left((-1)^{c_j+s_j} \right)^2 \right) + \sum_{j \neq j'=1}^{N} E\left((-1)^{c_j+s_j} (-1)^{c_{j'}+s_{j'}} \right) \\ &= N + \sum_{j \neq j'=1}^{N} E\left((-1)^{c_j+s_j} \right) E\left((-1)^{c_{j'}+s_{j'}} \right) \\ &= N + N(N-1)(2p-1)^2. \end{aligned}$$

La fonction f est indépendante du temps (donc de l'indice j). Par conséquent les événements $c_j + s_j$ et $c_{j'} + s_{j'}$ sont indépendants.

La variance est égale à

$$\begin{aligned} E(\alpha^2) - E^2(\alpha) &= N + N(N-1)(2p-1)^2 - (N(2p-1))^2 \\ &= 4Np(1-p). \end{aligned}$$

On voudrait mener une attaque à clair connu : on connaît un certain nombre de bits de la suite chiffrante et on voudrait retrouver l'état d'initialisation des registres.

La fonction de combinaison f et les polynômes de rétroaction des n LFSRs étant supposés connus, l'idée consiste à déterminer l'initialisation de chaque LFSR indépendamment les uns des autres.

Pour cela, il suffit d'utiliser habilement une éventuelle corrélation entre la sortie (σ) d'un registre et la sortie (s) de la suite chiffrante: d'où une attaque par corrélation.

Si l'initialisation du registre i n'est pas correcte, la suite (σ) sera complètement décorrélée de la suite (s) et α_i aura une distribution normale **de moyenne nulle** et de variance N.

Par contre, si l'initialisation du registre est correcte, la corrélation α_i entre la suite s et σ suit une loi normale **de moyenne non nulle.**

L'algorithme est alors le suivant :

- Calculer p_i,
- pour chacune des $(2^{L_i} - 1)$ initialisations possibles du i ème registre,
- calculer la corrélation entre la suite s et σ produite sur N bits,
- si $\alpha_i \simeq N(1-2p_i)$ alors l'initialisation est correcte sinon l'initialisation n'est pas correcte.

L'attaque par corrélation de Siegenthaler permet donc de retrouver l'initialisation des registres en

$$\prod_{\mathbf{j,p_j=0,5}} (2^{L_j} - 1) + \sum_{\mathbf{j,p_j \neq 0,5}} (2^{L_j} - 1) \text{ essais,}$$

alors qu'une recherche exhaustive nécessite

$$\prod_{\mathbf{j=1}}^{\mathbf{m}} (2^{L_j} - 1) \text{ essais,}$$

Des raffinements de cette attaque ont été étudiés par W. Meier et O. Staffelbach, ainsi que V. Chepyzhov et B. Smeets.

Dans un système combinant m LFSR on peut éviter ce type d'attaque en choisissant f non corrélée à l'ordre k pour k petit.

Définition 11 *On dit que f est* **non-corrélée à l'ordre k** *si, pour $X_1, \ldots, X_m$ des variables aléatoires binaires équidistribuées indépendantes, la variable aléatoire $f(X_1, \ldots, X_m)$ est indépendante des variables $\sum_{i \in I} X_i$, où I est un sous-ensemble de $\{1, \ldots, m\}$ de cardinal au plus égal à k.*

On dit que f est **k-résiliente** *si f est non-corrélée à l'ordre k, et équilibrée.*

Cela revient à dire que

$$d_H\left(f, \sum_{i \in I} x_i\right) = 2^{m-1},$$

pour tout sous-ensemble I de cardinal au plus égal à k.