Les représentations équivalentes d'un LFSR et leur impact en cryptanalyse

Yann Rotella

Les représentations équivalentes d'un LFSR et leur impact en cryptanalyse

Yann Rotella, encadré par Anne Canteaut
INRIA équipe SECRET
yann.rotella@inria.fr

Mars - Août 2015

Fiche de synthèse

Contexte général. Le principe du chiffrement à flot est de générer une séquence pseudo-aléatoire de bits, avec laquelle on va additionner le message clair par un XOR bit à bit, ce qui nous donnera le chiffré. Le chiffrement à flot se fonde donc sur le principe du chiffrement de Vernam (one-time-pad) [1]. Cependant, dans le chiffrement de Vernam, la clef secrète est une suite aléatoire à usage unique et doit être longue que le message. C'est cependant beaucoup trop lourd à mettre en place en pratique. Il est donc nécessaire de concevoir des générateurs pseudo-aléatoires. Une des solutions est d'utiliser des registres à rétroaction linéaire (LFSR : Linear Feedback Shift Register) pour faire évoluer l'état interne du générateur, qui coûtent peu en temps de calcul et en nombre de portes logiques et qui produisent des suites ayant de bonnes propriétés statistiques. Les systèmes de chiffrement à flot utilisant des LFSR sont encore utilisés dans beaucoup de protocoles, autant dans des environnements logiciels que matériels. On peut citer par exemple : E0 (Bluetooth), A5/1 (GSM), SNOW (réseau 3G). En revanche, les LFSR peuvent pas être utilisés seuls, car ils sont très faciles à attaquer. L'utilisation de fonctions de filtrage sur un ou plusieurs LFSR peut être une des solutions pour complexifier le système et le rendre plus sûr. Ainsi le LFSR filtré est un modèle de chiffrement à flot à coût réduit, qui est encore actuellement très utilisé dans des systèmes réels soit tout seul [2] soit comme partie d'un générateur plus complexe.

Position du problème. De nombreuses attaques existent déjà sur ce type de systèmes comme les attaques par corrélation ou les attaques algébriques. Dans [3], Rønjom et Cid ont constaté que, dans certains cas, les représentations équivalentes des LFSR filtrés obtenues par un changement de racine primitive pouvaient aboutir à un nouveau type de cryptanalyse. Ils ont aussi décrit une attaque qui s'applique quand la fonction de filtrage est une composante d'une fonction monôme $x \mapsto \mathsf{Tr}(\lambda x^r)$ sur le corps fini $\mathbb{F}_{2^n}$ où n est la longueur du LFSR. Beaucoup de questions se posent en conséquence de cet article : la possibilité de généraliser cette attaque à d'autres fonctions de filtrage et la recherche d'autres attaques exploitant ces représentations équivalentes.

Contribution proposée. L'utilisation des représentations équivalentes introduites dans [3] m'a permis de mettre en évidence trois nouveaux types de faiblesses potentielles pour les LFSR filtrés. Alors que Rønjom & Cid avaient montré que si la représentation trace de la fonction de filtrage ne comporte qu'un seul terme, alors le LFSR filtré est équivalent à un unique LFSR de même taille, j'ai montré que dans le cas d'une fonction formée par la somme de deux termes, il est équivalent à un LFSR filtré par une fonction de degré au plus $\frac{n}{2}$ quel que soit le degré de la fonction d'origine. Plus généralement, j'ai donné une preuve simple du fait que la complexité linéaire d'un LFSR filtré était entièrement déterminée par les termes apparaissant dans la forme trace de la

fonction de filtrage. En particulier, la sécurité augmente avec le nombre de termes de la fonction. Enfin, j'ai proposé une nouvelle attaque qui exploite la corrélation entre le LFSR filtré et la sortie du même LFSR filtré par une fonction ayant un seul terme trace. Cette attaque généralise les attaques par corrélation classiques, et montre que la notion de non-linéarité généralisée introduite dans [4] est pertinente dans ce contexte. Toutefois, nous avons démontré que cette notion n'était pas suffisante et qu'il fallait considérer la distance de f à un plus grand nombre de fonctions.

Arguments. Les résultats présents dans ce document montrent que les représentations équivalentes des LFSR filtrés induisent beaucoup plus d'attaques que celles déjà connues et que les quelques exemples présentés dans [3]. Dans ces conditions, le travail réalisé impacte réellement les problèmes de sécurité cryptographique des LFSR filtrés.

Bilan. Nous avons exhibé de nouvelles attaques sur les LFSR filtrés. Cependant de nombreux problèmes sont apparus et nous avons mis en exergue quelques questions ouvertes intéressantes. Le calcul de la forme trace d'une fonction booléenne à partir de sa forme normale algébrique ou de sa table de vérité sur un nombre élevé de variables est pour l'instant hors de portée. La structure des poids des exposants des fonctions monômes pose encore beaucoup de questions intéressantes aujourd'hui sans réponse et il serait intéressant de continuer à travailler sur le sujet. Finalement, de nouveaux critères sont à prendre en compte pour améliorer la sécurité de ces systèmes, mais ces critères sont encore très difficiles à quantifier. Ce seront les premières questions que je me poserai dans le cadre de ma thèse que je poursuis à l'issue de ce stage de recherche.

Résumé du document. Dans un premier temps, un état de l'art sur les registres filtrés et les attaques existantes sera présenté. Dans un deuxième temps, nous décrirons les travaux réalisés et les pistes de recherche à développer. Nous avons travaillé sur des cas particuliers de fonctions booléennes et plus précisément sur la structure des exposants des fonctions booléennes. Nous avons aussi pu donner une preuve du calcul exact de la complexité linéaire de la séquence générée par un registre filtré. Finalement, nous avons conçu une nouvelle attaque basée sur les attaques par corrélation sur les registres filtrés en élargissant les représentations équivalentes possibles d'un LFSR filtré en considérant des exposants non premiers avec $2^n - 1$, ce qui n'a pas été fait en [3].

Table des matières

1 Introduction à la cryptographie

Chiffrement des données. La cryptographie est la science qui permet de chiffrer des données sur un canal de communication non sécurisé. Elle répond aux besoins de communiquer de manière confidentielle. On différencie deux formes de cryptographie : symétrique et asymétrique. La cryptographie asymétrique permet en particulier de partager un secret entre deux personnes sur un canal non sécurisé, sans que celles-ci ne partagent de secret commun. Alors que la cryptographie symétrique permet à deux personnes partageant un secret (la clef) de communiquer sans permettre à un adversaire d'apprendre des informations ou sur la clef ou sur le message initial (message clair).

Chiffrements symétriques. Il existe deux grandes familles de chiffrements symétriques : les chiffrements par blocs et les chiffrements à flots. le principe du chiffrement par bloc est de découper le message clair en blocs de taille fixe, et de chiffrer séparément ou non ces blocs (AES), alors que le chiffrement par flot a pour but de générer au moyen de générateurs pseudo aléatoires une suite chiffrante, qui sera additionnée bit à bit au message clair afin d'obtenir le message chiffré. De manière générale, les chiffrements à flot sont utilisés dans des implémentations hardware pour leur faible coût d'utilisation et leur rapidité.

Chiffrement de Vernam. Les chiffrements à flot se fondent sur le chiffrement de Vernam (one time pad) dont le principe est le suivant :

1. La clef est une suite binaire aussi longue que le message.
2. Les bits de clef sont choisis aléatoirement.
3. Chaque clef ne doit être utilisée qu'une seule fois.
4. Puis on additionne modulo 2 la clef et le message clair bit par bit.

Ce chiffrement offre une sécurité inconditionnelle absolue. En revanche, générer une telle suite coûte très cher, et c'est d'autant plus embêtant qu'elle ne peut être utilisée qu'une seule fois. Dans ces conditions, nous cherchons alors à générer des suites pseudo aléatoires, au moyen d'une clef secrète, afin que l'on puisse réutiliser la clef plusieurs fois, et que la taille de la suite chiffrante soit la plus grande possible, tout cela en essayant de garder une bonne sécurité.

Chiffrements basés sur les registres à décalage. Nous devons générer des suites qui semblent aléatoires. Pour répondre à ce problème, nous utilisons des registres à décalages linéaires (LFSR - Linear Feedback Shift Register). En revanche, utilisés tous seuls, ceux ci ne sont pas une bonne solution puisqu'il est possible de retrouver la clef facilement car ils sont entièrement linéaires. Dans ces conditions, ils sont utilisés dans plusieurs chiffrements à flot mais pas seuls, notamment combinés à des fonctions booléennes.
Notre travail a donc consisté à étudier plus en détail les registres à décalage linéaires filtrés.

2 Définitions et généralités

2.1 LFSR

Remarque 2.1. *Dans tout le document, nous pourrons identifier l'espace $\mathbb{F}_2^n$ au corps fini $\mathbb{F}_{2^n}$.*

Définition 2.1 (Linear Feedback Shift Register)**.** *Un LFSR de taille L sur le corps $\mathbb{F}_q$ est un automate fini qui produit une séquence d'éléments $s = (s_t)_{t\geq 0}$ du corps $\mathbb{F}_q$ satisfaisant une relation de récurrence de degré L sur $\mathbb{F}_q$.*

$$s_{t+L} = \sum_{i=1}^{L} c_i s_{t+L-i},\ \forall t \geq 0$$

Les L coefficients $c_1, \ldots, c_L$ sont les coefficients de rétroaction du LFSR.
Le registre du LFSR contient L cellules, chacune étant constituée d'un élément de $\mathbb{F}_q$. L'ensemble contenu dans ces cellules forme l'état du LFSR à un instant t. De plus le LFSR est initialisé par $s_0, \ldots, s_{L-1}$: c'est l'état initial (non nul) du LFSR.

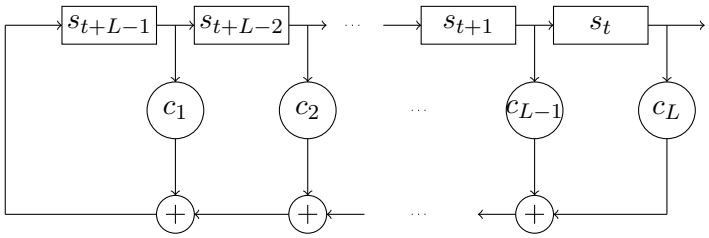

FIGURE 1 – Représentation d'un LFSR

La séquence produite par un LFSR est donc déterminée de manière unique par ses coefficients de rétroaction et son état initial. Les coefficients de rétroaction définissent alors le polynôme de rétroaction du LFSR.

$$P(X) = 1 - \sum_{i=1}^{L} c_i X^i$$

D'un autre côté, nous pouvons aussi définir le polynôme caractéristique du LFSR comme étant le polynôme réciproque du polynôme de rétroaction :

$$P^\star(X) = X^L P(1/X)$$

La caractérisation des séquences engendrées par les LFSR [5] permet alors de se restreindre aux registres à décalage dont le polynôme caractéristique est un polynôme primitif (le polynôme minimal de la séquence produite). Si en revanche le polynôme n'est pas primitif, alors la période de la séquence générée ne sera pas maximale. L'ordre du polynôme minimal du LFSR détermine la période de la séquence produite.

LFSR et éléments du corps fini

Proposition 2.1. *[6] Soit $P^\star$ un polynôme irréductible dans $\mathbb{F}_q[X]$ de degré L. Soit $\alpha \in \mathbb{F}_{q^L}$ une racine de $P^\star$ et $\beta_0, \ldots, \beta_{L-1}$ la base duale de $1, \alpha, \ldots, \alpha^{L-1}$.*
Alors l'état du LFSR de polynôme caractéristique $P^\star$ à l'instant $(t+1)$ est l'état du LFSR à l'instant t multiplié par α, où l'on identifie les vecteurs de $\mathbb{F}_q$ aux éléments de $\mathbb{F}_{q^L}$ en décomposant ces vecteurs dans la base duale $\beta_0, \ldots, \beta_{L-1}$.

Ainsi, si X_0 est l'état initial d'un LFSR de polynôme caractéristique $P^\star$ définissant une racine α, alors à l'instant t, l'état du LFSR est l'élément $X_0\alpha^t$ du corps fini $\mathbb{F}_{q^L}$.

Algorithme de Berlekamp-Massey. L'algorithme de Berlekamp-Massey permet de calculer la complexité linéaire (voir section 4) d'une séquence et son polynôme minimal. En 1969, J. Massey [7] a montré que l'algorithme proposé par Berlekamp [8] pouvait être utilisé pour retrouver le polynôme minimal d'une suite uniquement à partir de ses $2L$ premiers bits où L est la complexité linéaire de la séquence.

Dans ces conditions, on ne peut pas utiliser un registre seul pour générer une suite chiffrante seule. Plusieurs propositions existent pour complexifier les systèmes, comme la combinaison de générateurs ou les registres filtrés. Nous nous intéresserons dans la suite au bloc *"registre filtré"*.

Registres filtrés. Les registres filtrés sont encore très souvent utilisés seuls ou comme blocs dans d'autres systèmes cryptographiques. Une étude plus approfondie des registres filtrés est donc nécessaire.

Définition 2.2. *Un registre filtré (ou générateur filtré) est un générateur pseudo-aléatoire composé d'un unique LFSR de taille L dont le contenu est filtré par une fonction booléenne f (voir la figure 2). La suite pseudo-aléatoire générée ainsi satisfait la relation suivante :*

$$s_t = f(u_{t+\gamma_1}, \cdots, u_{t+\gamma_n})$$

où $(u_t)_{t\leq 0}$ est la suite produite par le LFSR et $(\gamma_i)_{1\leq i\leq n}$ une suite décroissante d'entiers de $\{0, \cdots, L-1\}$.

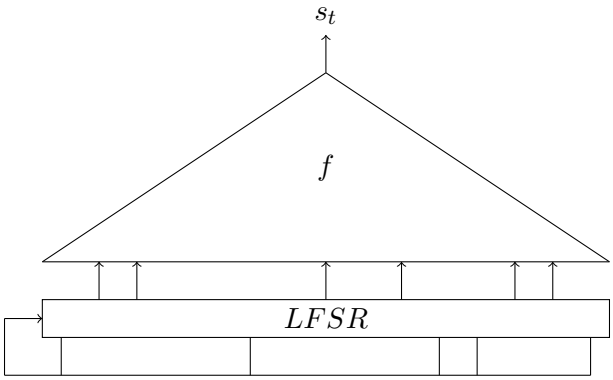

FIGURE 2 – Registre filtré

Il est à noter qu'en pratique, la fonction de filtrage f ne prend en entrée qu'une dizaine de bits du LFSR alors que ce dernier est généralement de taille 128 ou 256 bits.
Dans la suite, on considérera parfois la fonction de filtrage comme une fonction dont le nombre de variables est égal à la taille du LFSR (même si certaines entrées n'influencent pas la sortie de la fonction). Dans ce cas, les deux paramètres L et n seront notés identiquement n.

2.2 Fonctions booléennes

Définition 2.3. *Une fonction booléenne est une fonction de $\mathbb{F}_2^n$ à valeurs dans $\mathbb{F}_2$.*

Ainsi une fonction booléenne est entièrement définie par sa table de vérité (de taille 2^n).

Notation 2.1. *Dans toute la suite on notera le poids de Hamming d'une fonction booléennne f ou d'un mot binaire m : $\mathsf{w}_H(f)$ ou $\mathsf{w}_H(m)$.*

Définition 2.4 (Forme algébrique normale)**.** *La forme algébrique normale (ANF) est la représentation unique d'une fonction booléenne comme polynôme multivarié sur $\mathbb{F}_2$:*

$$f(x) = \sum_{I\in P(N)} a_I x^I$$

où $P(N)$ désigne l'ensemble des parties de $\{1, \ldots, N\}$.

Il est à noter que chaque exposant de x_i prendra uniquement les valeurs 0 ou 1 puisque l'on se situe dans le corps de caractéristique 2.

Définition 2.5. *Pour chaque entier k et r qui divise k, on définit classiquement la fonction trace de $\mathbb{F}_{2^k}$ dans $\mathbb{F}_{2^r}$, notée par $\mathsf{Tr}_r^k(\cdot)$ de la manière suivante :*

$$\mathsf{Tr}_r^k(x) = \sum_{i=0}^{\frac{k}{r}-1} x^{2^{ir}} = x + x^{2^r} + x^{2^{2r}} + \cdots + x^{2^{k-r}}$$

La plupart du temps, on utilisera la fonction trace sur $\mathbb{F}_2$, nous noterons alors Tr^n et quand le contexte sera suffisant pour la compréhension, nous écrirons seulement Tr.

Propriété 2.1 (de la fonction trace)**.** *La fonction trace vérifie les quatre propriétés suivantes :*

1. *La fonction trace est surjective ;*
2. $\mathsf{Tr}_k^n(ax + by) = a\mathsf{Tr}_k^n(x) + b\mathsf{Tr}_k^n(y)$ *pour* $a, b \in \mathbb{F}_{2^k}$ *et* $x, y \in \mathbb{F}_{2^n}$ *;*
3. $\mathsf{Tr}_k^n(x^{2^k}) = \mathsf{Tr}_k^n(x)$ *pour* $x \in \mathbb{F}_{2^n}$ *;*
4. *Si* $\mathbb{F}_{2^k} \subset \mathbb{F}_{2^r} \subset \mathbb{F}_{2^n}$*, alors la fonction trace est transitive :* $\mathsf{Tr}_k^n = \mathsf{Tr}_k^r \circ \mathsf{Tr}_r^n$

Définition 2.6 (Transformée de Walsh-Hadamard)**.** *Dans le cas particulier des fonctions booléennes, le coefficient de Walsh en $a \in \mathbb{F}_2^n$ de la fonction f est*

$$\widehat{f}(a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+a\cdot x}$$

de plus, $\mathsf{w}_H(f) = 2^{n-1} - \frac{1}{2}\widehat{f}(0)$.

Propriété 2.2 (lien entre les coefficients de Walsh et la distance aux fonctions affines)**.** *Soit f une fonction booléenne à n variables, $a \in \mathbb{F}_2^n$, $b \in \mathbb{F}_2$, et $\phi_{a,b}$ l'application affine définie par $\phi_{a,b}(x) = a \cdot x + b$, $x \in \mathbb{F}_2^n$, alors*

$$d(f, \phi_{a,b}) = 2^{n-1} - \frac{(-1)^b}{2}\widehat{f}(a)$$

où $d(f, g)$ est la distance entre les fonctions booléennes f et g.

Nous voulons aussi que la suite chiffrante sortant de notre système de chiffrement à flot ait les mêmes propriétés qu'une suite aléatoire. Dans ces conditions, il est nécessaire en particulier d'utiliser des fonctions booléennes équilibrées.

Définition 2.7. *Une fonction booléenne f à n variables est dite équilibrée si* $\mathsf{w}_H(f) = 2^{n-1}$*, ou de façon équivalente* $\widehat{f}(0) = 0$.

Proposition 2.2 (Représentation univariée des fonctions booléennes)**.** *On identifie tout d'abord $\mathbb{F}_2^n$ avec $\mathbb{F}_{2^n}$ et l'on considère que la fonction booléenne a ses entrées dans $\mathbb{F}_{2^n}$. Dans ces conditions, il existe une unique représentation univariée de la fonction booléenne de la forme suivante :*

$$f(x) = \sum_{j \in \Gamma_n} \mathsf{Tr}^{C_j}(a_j x^j) + \varepsilon(1 + x^{2^n-1})$$

avec :

1. *Γ_n est l'ensemble des entiers obtenus en choisissant un élément dans chaque classe cyclotomique modulo $2^n - 1$. On utilisera souvent le plus petit représentant.*
2. *C_j est la taille de la classe cyclotomique contenant j.*
3. $a_j \in \mathbb{F}_{2^{C_j}}$
4. $\varepsilon = \mathsf{w}_H(f) \mod 2$

2.3 Cryptanalyse

De très nombreuses attaques existent déjà pour des générateurs pseudo-aléatoires basés sur les LFSR. Dans ce chapitre, nous donnerons un rapide état de l'art de ce qui existe déjà en terme d'attaques.

2.3.1 Attaques algébriques

Le principe de ces attaques est d'écrire des équations de petit degré entre les bits de sortie de la fonction de chiffrement et les bits de la clef utilisée pour le chiffrement (ou dans notre cas l'état initial du LFSR) afin de retrouver ces derniers. Cependant, la sortie n'est pas linéaire en les bits inconnus. Dans ces conditions nous écrivons tout d'abord un système d'équations algébriques que l'on résoud par exemple en le linéarisant i.e. en rajoutant une nouvelle variable indépendante pour chaque monôme présent dans le système d'équations. Ainsi, dans le cas des registres filtrés de taille L, si la fonction est de degré d, alors nous aurons un système avec

$$D = \sum_{k=0}^{d} \binom{L}{k}$$

équations. Donc, pour réussir une attaque algébrique classique, il est nécessaire de connaître au minimum D bits de la suite chiffrante ; de plus, il faut ensuite résoudre un système linéaire, qui coûte en complexité $O(D^3)$ avec un algorithme classique. Cette complexité peut être améliorée par l'utilisation d'algorithmes de résolution de systèmes polynomiaux comme par exemple à l'aide des bases de Gröbner. Des attaques algébriques plus performantes ont également été développées afin d'améliorer sensiblement la complexité de ces attaques [9, 10].

2.3.2 Immunité algébrique

Au lieu d'écrire des équations sur la fonction filtrante f, nous pouvons aussi écrire des équations sur une fonction annulatrice de f. En effet, si h est telle que $\forall x,\ f(x)h(x) = 0$, alors lorsque $f(x) = 1$, on a $h(x) = 0$. Ainsi, nous pouvons écrire nos équations sur h et non sur f. Si la fonction h est de petit degré, alors cela devient intéressant. En effet, dans l'évaluation de complexité précédente, d correspond au degré de h et non plus à celui de f.

Définition 2.8. *On définit l'immunité algébrique d'une fonction f par la quantité suivante :*

$$\mathsf{AI}(f) = \min(\{\deg(g) : g \neq 0, fg = 0\} \cup \{\deg(h), h \neq 0, (1+f)h = 0\})$$

Nous avons de plus les propriétés suivantes :

Propriété 2.3. *L'immunité algébrique d'une fonction booléenne à n variables vérifie :*

1. $\mathsf{AI}(f) \leq \deg(f)$
2. $\mathsf{AI}(f) \leq \lceil \frac{n}{2} \rceil$

D'autres résultats plus précis existent sur l'immunité algébrique des fonctions booléennes, notamment une borne en fonction du nombre de plages de 1 dans les exposants de la représentation trace des fonctions booléennes [11].

2.3.3 Attaques par corrélation

Cette famille d'attaques s'applique à n'importe quel générateur pseudo-aléatoire produisant une suite chiffrante, si la séquence sortante est fortement corrélée avec la séquence sortant d'un automate dont l'état initial dépend d'une partie des bits de la clef. Dans ces conditions, il est

possible de retrouver une partie des bits de la clef par une attaque par corrélation [12]. De plus, si cet automate a une transition linéaire (typiquement un LFSR), on peut utiliser une attaque par corrélation rapide [13].
Dans notre cas particulier de registres filtrés, le critère de non-linéarité des fonctions booléennes nous assure une certaine résistance à ce type d'attaque. La non-linéarité d'une fonction booléenne correspond à la distance de celle-ci à toutes les fonctions affines et se calcule rapidement à l'aide des coefficients de la transformée de Walsh.

Définition 2.9. *On appelle non-linéarité d'une fonction booléenne f à n variables la distance qui la sépare de l'ensemble des fonctions affines à n variables. On notera cette valeur*

$$\mathsf{NL}(f) = \min_{h \text{ affine}} d(f,h)$$

Propriété 2.4. *Soit f une fonction booléenne à n variables, alors les propriétés suivantes se vérifient.*

1. $\mathsf{NL}(f) = 2^{n-1} - \frac{1}{2}\max_{a\in\mathbb{F}_2^n} |\widehat{f}(a)|$
2. $\mathsf{NL}(f) \leq 2^{n-1} - 2^{n/2-1}$

Définition 2.10. *On dira qu'une fonction booléenne est courbe si $\mathsf{NL}(f) = 2^{n-1} - 2^{n/2-1}$. Les fonctions courbes n'existant naturellement que si n est pair.*

Proposition 2.3 (Lien entre immunité algébrique et non-linéarité)**.** *Une faible non-linéarité implique nécessairement une faible immunité algébrique.*

$$\textit{Si } \mathsf{NL}(f) < \sum_{i=0}^{d} \binom{m}{i} \textit{ alors } \mathsf{AI}(f) \leq d+1$$

En revanche une immunité algébrique maximale $\lceil \frac{n}{2} \rceil$ n'implique pas nécessairement que la non linéarité soit aussi maximale : $2^{n-1} - 2^{n/2-1}$.

Nous n'expliciterons pas en détail les différentes techniques d'attaques par corrélation [14] [15], cependant, nous retiendrons les résultats de complexité suivants.

2.3.4 Attaques par corrélation rapide

Considérons un LFSR filtré par f à n variables, d'état initial X_0, générant une suite s, où s est fortement corrélée avec σ, suite sortant d'un LFSR non filtré d'état initial X'_0 de taille L. Alors les complexités en nombre de bits (N) nécessaires de la suite chiffrante, en précalcul (P) et en temps (T) sont les suivantes.

$$N = 2^{\frac{L}{d-1}} \left(\frac{1}{2\varepsilon}\right)^{\frac{2(d-2)}{d-1}} \qquad P = \frac{N^{d-2}}{(d-2)!} \qquad T = 2^{\frac{L}{d-1}} \left(\frac{1}{2\epsilon}\right)^{\frac{2d(d-2)}{d-1}} \tag{1}$$

où d est un entier $d \geq 3$ qui correspond au nombre de termes des équations de parité utilisées dans l'attaque [16] et ε est le biais entre s et σ : $\Pr[s_t = \sigma_t] = \frac{1}{2} + \varepsilon$
Notre travail consistera à regarder plus en détail les registres filtrés, utiliser des représentations équivalentes afin d'en déduire un impact sur la cryptanalyse de ces systèmes.

3 Générateurs équivalents à un LFSR filtré

3.1 LFSR filtrés équivalents

Définition 3.1. *On dira qu'un système de chiffrement à flot S' est équivalent à un autre système S si S' peut produire la même séquence (in-)finie que le système S et ce pour tout état initial de S et réciproquement.*

Dans la suite, on identifiera l'état interne d'un LFSR de taille n à un élément du corps $\mathbb{F}_2^n$ au sens de la prop. 2.1, et la fonction de filtrage est vue comme une fonction de $\mathbb{F}_{2^n}$ dans $\mathbb{F}_2$.

Lemme 3.1 (Représentation équivalente linéaire)**.** *Soit un LFSR de taille n filtré par une fonction booléenne f et d'état initial X_0. Alors le même LFSR, filtré par la fonction f' définie par $f'(x) = f(\lambda x)$ avec $\lambda \in \mathbb{F}_{2^n}, \lambda \neq 0$ et d'état initial $\lambda^{-1}X_0$ produit la même séquence.*

Remarque 3.1. *Ceci n'est d'aucun intérêt pour la cryptanalyse puisque les notions classiques de degré, d'immunité algébrique et de non-linéarité (attaques par corrélation) sont naturellement invariantes par la transformation précédente.*

Cependant, des représentations équivalentes non linéaires existent [3]. On considère un générateur pseudo-aléatoire filtré de polynôme primitif P de degré n définissant ainsi un élément primitif α sur le corps fini $\mathbb{F}_{2^n}$ et muni d'une fonction de filtrage f.
L'équivalence proposée consiste à considérer les LFSR filtrés de même taille, produisant les mêmes séquences. Comme la période des séquences est $2^n - 1$, nous devons prendre tous les polynômes primitifs de degré n. Ceux-ci définissent chacun un élément primitif β qui peut s'écrire sous la forme $\beta = \alpha^k$ où $\text{pgcd}(k, 2^n - 1) = 1$. Ainsi, en considérant les représentants qui sont premiers avec $2^n - 1$ des classes cyclotomiques, on obtient tous les LFSR primitifs de taille n (en effet, les éléments α^{2^i} ont même polynôme minimal donc il suffit de se restreindre aux représentants des classes cyclotomiques). De plus, on ne s'intéresse qu'aux éléments générateurs du groupe multiplicatif de $\mathbb{F}_{2^n}$, sinon le nouveau LFSR filtré ne pourrait pas produire le même ensemble de séquences.
Soit une transformation qui associe l'état interne du LFSR d'origine à celui du LFSR équivalent :

$$\begin{array}{rccc} \Psi_\beta & : & \mathbb{F}_{2^n} & \to & \mathbb{F}_{2^n} \\ & & X & \mapsto & Y \end{array} \tag{2}$$

où X et Y sont respectivement les états internes possibles non nuls des deux LFSR filtrés i.e. les éléments non nuls du corps $\mathbb{F}_{2^n}$. Nous allons maintenant déterminer l'expression des transformations Ψ_β. Nous voulons que la transformation Ψ_β vérifie la condition suivante :

$$\forall 0 \leq t \leq 2^n - 2, \Psi_\beta(X\alpha^t) = Y\beta^t \tag{3}$$

afin que chaque état interne à l'instant t du premier LFSR corresponde à l'état interne au même instant t du second générateur. Avec la condition précédente, on en déduit donc que la transformation Ψ_β est entièrement définie par l'image d'un seul élément non nul X_0. Notons $Y_0 = \Psi_\beta(X_0)$. Posons alors $X_0 = \alpha^i$. Alors il existe $\lambda \in \mathbb{F}_{2^n}, \lambda \neq 0$ tel que $Y_0 = \lambda\beta^i$. Avec ces notations et la condition (3) que doit vérifier notre transformation, nous en déduisons que les transformations possibles entre nos deux LFSR différents sont des monômes, c'est-à-dire qu'elles sont de la forme :

$$\begin{array}{rccc} \Psi_{\beta,\lambda} & : & \mathbb{F}_{2^n} & \to & \mathbb{F}_{2^n} \\ & & X & \mapsto & \lambda X^k \end{array}$$

où k est tel que $\beta = \alpha^k$. Ou encore de manière équivalente :

$$\begin{array}{rccc} \Psi_{\beta,\lambda} & : & \mathbb{F}_{2^n} & \to & \mathbb{F}_{2^n} \\ & & \alpha^i & \mapsto & \lambda\beta^i \end{array}$$

où $\lambda \in \mathbb{F}_{2^n}, \lambda \neq 0$. Or, d'après la remarque 3.1, on peut se restreindre sans perdre de généralité aux fonctions $\Psi_{\beta,\lambda=1} = \Psi_\beta$. Finalement, afin de récupérer la même séquence en sortie du LFSR filtré, la nouvelle fonction de filtrage booléenne associée est la suivante : $f' = f \circ \Psi_\beta^{-1}$. Or comme $\beta = \alpha^k$ pour $\text{pgcd}(k, 2^n - 1) = 1$, les Ψ_β^{-1} sont les $\Psi_{\beta'}$ avec $\beta' = \alpha^{k'}$ et $kk' = 1 \mod (2^n - 1)$. Le nombre de ces LFSR filtrés équivalents est

$$\frac{\Phi(2^n - 1)}{n}$$

où Φ est la fonction indicatrice d'Euler. En effet, on a une transformation différente pour chaque k, représentant d'une classe cyclotomique (de taille n) et premier avec $2^n - 1$.
Par cette transformation, on peut créer des générateurs équivalents, ainsi, si l'on doit mesurer la sécurité de ces systèmes, il faut regarder la sécurité de l'ensemble de ces systèmes équivalents. En effet, l'attaquant a accès à la séquence sortant du LFSR filtré et veut réaliser une attaque (afin de retrouver l'état initial du LFSR). Si une représentation équivalente est plus facile à attaquer, l'attaquant va réaliser son attaque (par exemple une attaque algébrique), récupérer l'état initial du système équivalent, puis en appliquant Ψ_β^{-1}, l'attaquant peut donc retrouver l'état initial du système. Ainsi le niveau de sécurité d'un LFSR filtré est le niveau de sécurité minimal pour un générateur de sa classe d'équivalence.

3.2 Quand la fonction de filtrage est creuse

3.2.1 Fonction composante d'un monôme

Considérons maintenant que nous avons un LFSR filtré de taille n, d'état initial X_0, de polynôme primitif P définissant une racine primitive α et de fonction de filtrage $f(x) = \mathsf{Tr}(Ax^r)$, avec $A \in \mathbb{F}_{2^n}, A \neq 0$. Nous allons alors regarder en particulier l'impact des représentations équivalentes décrites à la section 3.1 sur le degré algébrique des fonctions booléennes de filtrage. Le cas où $\text{pgcd}(r, 2^n - 1) = 1$, correspondant au premier cas traité ci-dessous est celui qui avait été mis en évidence par Rønjom et Cid dans [3]. Nous allons ensuite montrer comment il se généralise à toutes les valeurs de r.

Premier cas. On suppose que $\text{pgcd}(r, 2^n - 1) = 1$. On utilise alors notre transformation Ψ_{α^k} en prenant k tel que $kr \equiv 1 \mod 2^n - 1$, . On a alors $\Psi_{\alpha^r}^{-1} = \Psi_{\alpha^k}$. Ainsi, on peut construire un LFSR filtré équivalent, dont la fonction de filtrage est maintenant :

$$f' = f \circ \Psi_{\alpha^k} = \mathsf{Tr}^n \left(A\Psi_{\alpha^k}(x^r)\right) = \mathsf{Tr}^n \left(Ax^{kr}\right) = \mathsf{Tr}^n (Ax)$$

La nouvelle fonction de filtrage est maintenant linéaire, et il est facile de retrouver l'état initial d'un tel système (voir l'algorithme de Berlekamp Massey). En réappliquant $\Psi_{\alpha^r}^{-1}$, on obtient alors aisément l'état initial du LFSR filtré de départ.

Exemple. Soit le LFSR de taille 6, dont le polynôme caractéristique est $P(X) = X^6 + X + 1$, définissant une racine primitive α. Ce registre est filtré par la fonction de filtrage suivante : $f(x) = \mathsf{Tr}(x^{23})$.
Cette fonction booléenne est de degré 4, d'immunité algébrique 2 et de non-linéarité 20.
D'après ce qui précède, nous pouvons assurer que ce système est équivalent au LFSR de taille 6 dont le polynôme caractéristique est le polynôme minimal de α^{23} qui est $P_{\alpha^{23}}(X) = X^6 + X^5 + X^4 + X + 1$. On associera bien sûr l'état initial X_0 à X_0^{23}. Si l'on observe par exemple en sortie $2 \times 6 = 12$ bits de la séquence suivante produite par le LFSR filtré :

$$(1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0)$$

alors l'algorithme de Berlekamp-Massey nous renvoie le polynôme $X^6 + X^5 + X^4 + X + 1$, et on obtient bien que $X_0^{23} = 1 + \beta^2 + \beta^3 + \beta^5$ où $\beta = \alpha^{23}$. Alors on retrouve que $X_0 = (X_0^{23})^{11} = 1 + \alpha + \alpha^2$.

Deuxième cas. On suppose maintenant que $\text{pgcd}(r, 2^n - 1) > 1$. En effet, il n'est pas nécessaire que x^r soit une permutation pour que la fonction $f(x) = \mathsf{Tr}(Ax^r)$ soit équilibrée pour certaines valeurs de A. Dans ces conditions, les éléments X^r où $X \in \mathbb{F}_{2^n}$ ne décrivent pas nécessairement un sous-corps de $\mathbb{F}_{2^n}$. Notons τ l'ordre de l'élément α^r ; nous savons que $\tau < 2^n - 1$ (sinon α^r serait

un élément primitif de $\mathbb{F}_{2^n}$, ce qui signifierait que $\text{pgcd}(r, 2^n - 1) = 1$). Nous pouvons affirmer alors que le système est équivalent à un LFSR de taille n mais dont le polynôme caractéristique noté P_r qui est le polynôme minimal de α^r n'est plus nécessairement primitif. Dans certains cas, il est possible que les éléments X^r où $X \in \mathbb{F}_{2^n}$ définissent un sous-corps de $\mathbb{F}_{2^n}$. La taille de ce sous-corps est le plus petit d tel que $2^d \equiv 1 \mod \tau$. Dans le cas particulier où $\tau < 2^n - 1$, le LFSR filtré initial est équivalent à un LFSR de taille $d \leq n$, d'état initial X_0^r et dont le polynôme de rétroaction est maintenant un polynôme (toujours non-nécessairement primitif) de $\mathbb{F}_{2^d}$ d'ordre τ. Dans tous les cas, la séquence sortant du système sera de période τ, et il existe un LFSR non filtré équivalent à notre LFSR filtré initial. Ainsi l'algorithme de Berlekamp Massey nous permettra de retrouver l'état initial du système équivalent X_0^r. La connaissance de X_0^r nous fournit donc $\log_2 \tau$ bits d'information sur X_0.

3.2.2 Somme de deux monômes

A présent, on s'intéresse aux fonctions de filtrage de la forme :

$$f(X) = \mathsf{Tr}(A_1 X^{k_1}) + \mathsf{Tr}(A_2 X^{k_2})$$

avec k_1 premier avec $2^n - 1$.
En utilisant les transformations possibles précédentes, on sait maintenant que l'on peut créer des LFSR filtrés équivalents dont les fonctions de filtrage sont de la forme :

$$f'_r(X) = \mathsf{Tr}(A'_1 X^{k_1 r}) + \mathsf{Tr}(A'_2 X^{k_2 r}) \qquad \forall r, \text{pgcd}(r, 2^n - 1) = 1$$

On va s'intéresser à réduire le degré de cette nouvelle fonction. On sait déjà que, lorsque l'on écrit une fonction booléenne sous sa représentation Trace, le degré de la fonction vaut :

$$\deg(f) = \max_k(\mathsf{w}_H(k))$$

quand k décrit l'ensemble des exposants présents dans la représentation trace. Pour notre problème particulier, nous allons donc montrer que l'on peut diviser le degré maximum de la fonction par deux. Comme on a supposé k_1 premier avec $2^n - 1$, en posant $r' = k_1^{-1} r$, notre problème appliqué à $f'_{r'}$ revient à minimiser sur tous les r premiers avec $2^n - 1$ la quantité $\max(\mathsf{w}_H(r), \mathsf{w}_H(kr))$ où $k = k_1^{-1} k_2$.

Proposition 3.1. *Soit $n \geq 2$. Pour tout entier $k \in [0, 2^n - 2]$ alors*

$$\min_{0<r<2^n-1, \text{pgcd}(r,2^n-1)=1} [\max(\mathsf{w}_H(r), \mathsf{w}_H(kr))] \leq \left\lceil \frac{n}{2} \right\rceil$$

et cette borne est atteinte si $\mathsf{w}_H(k) = n - 1$

Preuve. On suppose $\text{pgcd}(k, 2^n - 1) = 1$. On note de plus $X = \{x \in [0, 2^n - 2] : \text{pgcd}(x, 2^n - 1) = 1\}$. L'application

$$\begin{array}{rccc} \psi & : & X & \to & X \\ & & x & \mapsto & kx \end{array}$$

est bijective, et le nombre d'entiers premiers avec $2^n - 1$ de poids inférieur ou égal à $\lceil \frac{n}{2} \rceil$ est strictement supérieur au nombre de mots premiers de poids strictement supérieur à $\lceil \frac{n}{2} \rceil$.
En effet, il suffit de constater que si x est premier avec $2^n - 1$ et de poids t, alors $-x = 2^n - 1 - x$ est aussi premier avec $2^n - 1$ et l'on a $\mathsf{w}_H(x) = n - t$. Donc, comme ψ est bijective, il n'est pas possible que l'image par ψ des mots de poids inférieurs ou égaux à $\lceil \frac{n}{2} \rceil$ soit toujours de poids strictement supérieur à $\lceil \frac{n}{2} \rceil$. Ainsi, $\exists x$ avec $\text{pgcd}(x, 2^n - 1) = 1$ tel que $\mathsf{w}_H(x) \leq \lceil \frac{n}{2} \rceil$ et $\mathsf{w}_H(\psi(x)) \leq \lceil \frac{n}{2} \rceil$.
Il suffit de montrer que la borne est atteinte pour $k = 2^n - 2$ car $\forall i, w_H(2^i k) = w_H(k)$.

Or, $\forall x, (2^n-2)x = (2^n-1)x - x \equiv -x \mod (2^n-1) \equiv 2^n - 1 - x$
Ainsi, on obtient donc que si $\mathsf{w}_H(k) = n-1$, alors $\forall x, \mathsf{w}_H(kx \mod (2^n-1)) = n - \mathsf{w}_H(x)$. Donc, pour tout x, $\max(\mathsf{w}_H(x), \mathsf{w}_H(kx \mod (2^n-1))) = \max(\mathsf{w}_H(x), n - \mathsf{w}_H(x)) \geq \lceil \frac{n}{2} \rceil$.
Si $\text{pgcd}(k, 2^n-1) > 1$, alors l'application n'est plus bijective dans l'ensemble considéré précédemment. Cependant, la propriété reste vérifiée.
En effet, soit x un nombre premier avec 2^n-1 de poids exactement $\lceil \frac{n}{2} \rceil$ (on va montrer qu'il en existe sauf pour $n=4$), alors $-x$ est de poids inférieur à $\lceil \frac{n}{2} \rceil$ (égal si n est pair, strictement inférieur sinon). Alors l'image par ψ de x est de poids t, et l'image par ψ de $-x$ est de poids $n-t$. En choisissant x ou $-x$ selon la valeur de t, on peut conclure qu'il existe x avec $\text{pgcd}(x, 2^n-1) = 1$ tel que $\mathsf{w}_H(x) \leq \lceil \frac{n}{2} \rceil$ et $\mathsf{w}_H(\psi(x)) \leq \lceil \frac{n}{2} \rceil$.

Il reste à montrer que, pour tout $n > 4$, il existe bien des nombres premiers avec 2^n-1 de poids $\frac{n}{2}$ si n est pair et $\frac{n+1}{2}$ si n est impair. En effet, si n est impair, alors le nombre $2^{\frac{n+1}{2}} - 1$ convient car il est de poids $\frac{n+1}{2}$ et $\text{pgcd}(2^{\frac{n+1}{2}} - 1, 2^n - 1) = 2^{\text{pgcd}(\frac{n+1}{2}, n)} - 1$, or n est impair donc $\text{pgcd}(\frac{n+1}{2}, n) = \text{pgcd}(n+1, n) = 1$ donc $2^{\frac{n+1}{2}} - 1$ est premier avec $2^n - 1$.
Si n est pair, on note $t = \frac{n}{2}$ et on remarque que le nombre $A = 2^{2t-2} + 2^t + (2^{t-2} - 1)$ convient lui aussi, il est de poids $\frac{n}{2}$ si $n > 4$ et il est premier avec $2^n - 1$. En effet, $2^n - 1 = (2^t+1)(2^t-1)$ et on sait que $2^t + 1$ et $2^t - 1$ sont premiers entre eux. De plus A est premier avec $2^t - 1$ et $2^t + 1$ puisqu'il peut s'écrire sous la forme $A = 2^{t-2}(2^t+1) + (2^t - 1)$. Si d divise $2^t - 1$ et A, alors d divise $2^{t-2}(2^t+1)$ donc $d = 1$ puisque 2^{t-2} et $2^t + 1$ sont premiers avec $2^t - 1$ (idem en choisissant d' un diviseur de A et $2^t + 1$). On obtient donc que A est premier avec $2^n - 1$.

Pour $n = 4$, le nombre A défini ci-dessus ne convient pas puisque $2t - 2 = t$ et $t - 2 = 0$. On remarque alors par une recherche exhaustive qu'il n'existe pas de nombres premiers de poids 2 premiers avec $2^4 - 1 = 15$. Cependant une autre recherche exhaustive montre que la proposition 3.1 reste vérifiée pour $n = 4$ (et est triviale pour $n = 2$ et $n = 3$).

□

De plus, nous pouvons assurer que d'autres k apparaissent pour lesquels la borne est atteinte en regardant les travaux réalisés par S. Mesnager sur les fonctions hypercourbes [17]. Nous détaillerons ce point dans la section 5.

Exemple. Supposons que notre fonction de filtrage soit $f(x) = \mathsf{Tr}^6(x^{31}) + \mathsf{Tr}^6(x^3)$ sur $\mathbb{F}_{2^6}$. f est de degré algébrique 5, et $31^{-1} = 61 \mod 63$ et $3 \times 61 = 57 \mod 63$. On va donc chercher maintenant à trouver r premier avec 63 tel que $\max(\mathsf{w}_H(r), \mathsf{w}_H(57 \times r))$ soit minimal. On trouve alors que pour $r = 5$, $57 \times 5 = 12 \mod 63$ donc $\max(\mathsf{w}_H(r), \mathsf{w}_H(57 \times r)) = 2$.
Ainsi, en réalisant la transformation avec r un représentant de la classe cyclotomique de $(31 \times 5)^{-1}$ par exemple $r = 11$, on obtient un LFSR filtré équivalent de taille 6 dont la fonction de filtrage est maintenant $f'(x) = f(x^{11^{-1}}) = f(x^{23}) = \mathsf{Tr}^6(x^{20}) + \mathsf{Tr}^6(x^6) = \mathsf{Tr}^6(x^5) + \mathsf{Tr}^6(x^3)$.

Approfondissement : Il serait intéressant d'essayer de trouver une borne de ce problème qui dépende de k et non de n, mais aussi de comprendre ce que l'on peut faire lorsque k_1 et k_2 sont tous les deux non premiers avec $2^n - 1$, cependant la structure des exposants est complexe [18], ainsi il est difficile de conclure plus précisément sur ce problème.

3.3 Equivalence avec une somme de LFSR filtrés

Proposition 3.2. *Soit un LFSR de taille L, de polynôme caractéristique P_α, α élément primitif, filtré par une fonction booléenne f. Alors ce LFSR filtré est équivalent à la combinaison par XOR de deux générateurs aléatoires ayant chacun un état initial de taille L, mais dont les deux fonctions filtrantes g_1 et g_2 sont de degré d_1 et d_2, avec $d_i \leq \lceil \frac{n}{2} \rceil$.*

Preuve. On commence par écrire f dans sa représentation Trace :

$$f(x) = \sum_{k \in \Gamma_n} \mathsf{Tr}^{C_k}(A_k x^k)$$

où Γ_n est l'ensemble des représentants des classes cyclotomiques, C_k est la taille de la classe cyclotomique de k, $A_k \in \mathbb{F}_{2^{C_k}}$.
Ensuite, il est clair que $\text{pgcd}(2^n - 2, 2^n - 1) = 1$ et que $\forall r, \mathsf{w}_H((2^n - 2)r) = n - \mathsf{w}_H(r)$ car $(2^n - 2)r \equiv 2^n - 1 - r \mod (2^n - 1)$. Ainsi, on sépare les monômes présents dans l'écriture de notre fonction f en deux parties comme ceci :

$$f(x) = g_1(x) + g_2(x)$$

avec

$$g_1(x) = \sum_{k \in \Gamma_n, wt(k) \leqslant \lceil \frac{n}{2} \rceil} \mathsf{Tr}(A_k x^k) \text{ et } g_2(x) = \sum_{k \in \Gamma_n, wt(k) > \lceil \frac{n}{2} \rceil} \mathsf{Tr}(A_k x^k)$$

On peut donc représenter notre LFSR filtré comme combinaison de deux LFSR filtrés (voir figure 3) :

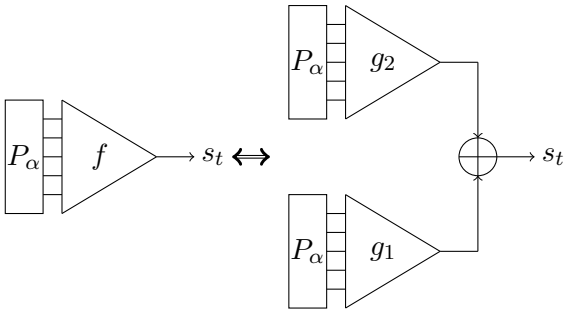

FIGURE 3 – Combinaison de deux LFSR filtrés

En réalisant maintenant notre transformation Ψ_β dans le LFSR filtré par g_2 avec $\beta = \alpha^{2^n-2}$, la nouvelle fonction g_2' est alors de degré plus petit que $\lceil \frac{n}{2} \rceil$ d'après la proposition 3.1. □

Cette nouvelle représentation équivalente ne semble pas directement exploitable, cependant, on peut s'inspirer de cette représentation pour en concevoir d'autres plus intéressantes.

4 Complexité linéaire

4.1 Rappels

Définition 4.1. *[19] La complexité linéaire d'une séquence semi infinie $s = (s_t)_{t \geq 0}$ à valeurs dans $\mathbb{F}_q$, notée $\Lambda(s)$, est le plus petit entier Λ tel que s peut être générée par un LFSR de taille Λ sur $\mathbb{F}_q$. Si aucun LFSR n'existe on pose $\Lambda = \infty$. La séquence nulle a pour complexité linéaire 0. La complexité linéaire d'une séquence récurrente linéaire est le degré de son polynôme minimal. La séquence peut aussi être finie de taille n.*

L'algorithme de Berlekamp Massey permet d'avoir accès à la complexité linéaire d'une séquence puisqu'il détermine le polynôme minimal et l'état initial d'un LFSR qui génèrerait cette séquence.

Combinaisons de générateurs Si l'on considère deux séquences linéairement récursives u et v sur $\mathbb{F}_q$ de complexité linéaire $\Lambda(u)$ et $\Lambda(v)$, alors on a les résultats suivants :

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

avec égalité si et seulement si les polynômes minimaux de u et de v sont premiers entre eux. Par ailleurs la période de $u+v$ est au moins un multiple commun des périodes de u et de v. De même,

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

avec égalité en particulier si les polynômes minimaux de u et v sont primitifs et si $\Lambda(u)$ et $\Lambda(v)$ sont distincts et plus grands que 2.

LFSR filtré En considérant maintenant un LFSR filtré par une fonction booléenne f. On a une borne très connue sur la complexité linéaire de la séquence produite [20]. Pour un LFSR binaire de polynôme primitif de degré L, de fonction de filtrage f de degré algébrique d on a :

$$\Lambda(s) \leq \sum_{k=1}^{d} \binom{L}{k}$$

Dans ce système de LFSR filtré, il faudra donc faire attention à avoir L et d suffisamment grands afin que l'attaquant ne puisse pas réaliser une attaque algébrique sur le système. En effet la complexité linéaire donne en réalité le nombre d'équations linéaires que l'attaquant a à résoudre lors d'une attaque algébrique afin de retrouver l'état initial. Ainsi, la taille de la séquence produite devra évidemment être bien plus petite que la complexité linéaire du système.

4.2 Calcul de la complexité linéaire d'un LFSR filtré

Proposition 4.1. *Soit un LFSR de taille n filtré par une fonction booléenne f que l'on écrit sous sa forme univariée :*

$$f(x) = \sum_{k \in \Gamma_n} \mathsf{Tr}^{C_k}(A_k x^k) + \varepsilon(1 + x^{2^n-1})$$

avec :

1. $A_k \in \mathbb{F}_{2^{C_k}}$
2. Γ_n *est l'ensemble des représentants des classes cyclotomiques modulo* $2^n - 1$.
3. C_k *est la taille de la classe cyclotomique contenant* k.

Alors la complexité linéaire de la séquence engendrée par un tel système est donnée par la relation suivante :

$$\Lambda(s) = \sum_{k \in \Gamma_n, A_k \neq 0} C_k$$

Ce résultat est mentionné en 2011 dans [21], nous en donnons une preuve en annexe A.

Exemple : Si l'on considère maintenant la fonction booléenne à 6 variables suivante : $f(x) = \mathsf{Tr}^6(x^3) + \mathsf{Tr}^6(x^{23}) + \mathsf{Tr}^3(x^9)$. Cette fonction est de degré algébrique 4, d'immunité algébrique 3 et de non-linéarité 24 (sachant que la non-linéarité maximale pour $n = 6$ vaut 28). Cette fonction booléenne possède d'assez bonnes propriétés cryptographiques. En revanche, d'après ce qui précède, si elle est utilisée pour filtrer un LFSR de taille 6, alors il est possible de retrouver l'état initial du registre si l'on a accès à $6 + 6 + 3 = 15$ bits de la suite chiffrante en résolvant simplement un système linéaire de taille 15 (ou avec l'algorithme de Berlekamp Masssey [8]) en

représentant notre système comme un LFSR non-filtré de taille 15 comme ci-dessus. Or, avec un tel système de chiffrement, on voudrait pouvoir générer une suite dont la complexité linéaire est de l'ordre de $\sum_{k=1}^{4} \binom{k}{6} = 56$ bits.
Dans ces conditions, il est nécessaire qu'il y ait beaucoup de termes non nuls dans la représentation univariée de la fonction de filtrage utilisée dans les registres filtrés.

5 D'autres représentations équivalentes

5.1 Attaque par corrélation généralisée

Au lieu de séparer la moitié des monômes dans la représentation trace de f comme il a été fait à la section 3.3, nous pouvons isoler par exemple un seul monôme, où l'exposant de ce monôme est premier avec $2^n - 1$.
Soit un LFSR de taille n de racine primitive α et d'état initial X_0, filtré par une fonction booléenne f. On suppose maintenant qu'il existe $\lambda \in \mathbb{F}_{2^n} \backslash \{0\}$ et k premier avec $2^n - 1$ tel que $d(f, \mathsf{Tr}^n(\lambda x^k)) = N$.
On note alors $g(x) = \mathsf{Tr}(\lambda x^k)$ et f' la fonction booléenne définie par $f'(x) = f(x) + g(x)$. Alors le système représenté sur la figure 4 de gauche est équivalent au système initial, puisqu'il produit la même séquence si on initialise les LFSR avec l'état initial X_0. Comme $\text{pgcd}(k, 2^n - 1) = 1$,

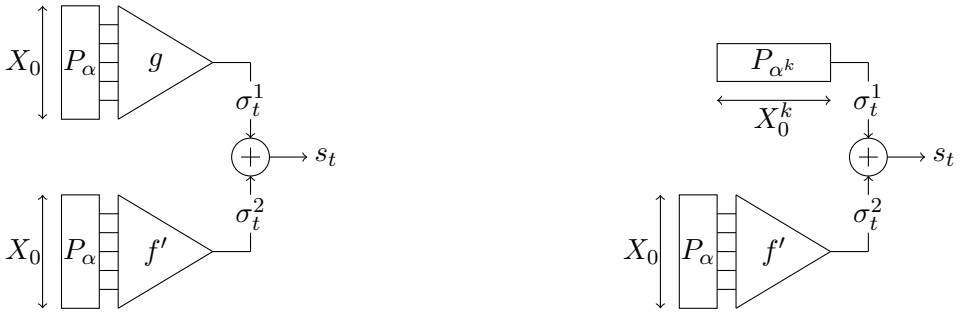

FIGURE 4 – Attaque par corrélation généralisée

nous pouvons réaliser la transformation décrite à la section 3.1 sur le LFSR filtré par la fonction g. Dans ces conditions, on obtient le système équivalent représenté figure 4 de droite :

Nous pouvons maintenant réaliser une attaque par corrélation visant à retrouver l'état initial du registre dont le polynôme de rétroaction est P_{α^k}. La probabilité d'erreur du canal binaire symétrique associé est de :

$$\Pr[s_t \neq \sigma_t^1] = \Pr[\sigma_t^2 = 1] = \frac{N}{2^n}$$

Nous retrouvons ainsi l'état initial X_0^k du LFSR non filtré de racine primitive α^k avec la complexité d'une attaque par corrélation classique si la non-linéarité vaut N. Or, comme $\text{pgcd}(k, 2^n - 1) = 1$, on en déduit alors directement l'état initial du LFSR filtré initial X_0. La complexité de notre attaque est la même que celle d'une attaque par corrélation classique, sauf qu'on peut s'autoriser maintenant à approximer f par n'importe quelle fonction monômiale de la forme $\mathsf{Tr}^n(\lambda x^k)$ où k est premier avec $2^n - 1$, sans perdre en complexité d'attaque. Dans ces conditions, il est donc nécessaire d'avoir recours à des fonctions très éloignées en distance de toutes les fonctions monômes $\mathsf{Tr}(\lambda x^k)$ où $\text{pgcd}(k, 2^n - 1) = 1$.
Ce critère a été énoncé une première fois en 2001 [4] mais sans être motivé par l'existence d'une attaque concrète. Ici, nous venons de montrer avec notre nouvelle attaque pourquoi ce critère est pertinent dans le cas des fonctions booléennes employées dans les LFSR filtrés. Depuis 2001, beaucoup de travaux ont été réalisés en vue de caractériser les fonctions hypercourbes, i.e. optimales pour ce critère.

5.2 Fonctions hypercourbes

Définition 5.1 (Transformée de Walsh étendue [4])**.** *Soit f une fonction booléenne, alors sa transformée de Walsh étendue est la fonction*

$$\widehat{f}(\lambda, k) = \sum_{x \in \mathbb{F}_{2^n}} (-1)^{f(x)+\mathsf{Tr}(\lambda x^k)}$$

où $\lambda \in \mathbb{F}_{2^n}$ et $\mathrm{pgcd}(k, 2^n - 1) = 1$.

Dans ces conditions, nous pouvons définir la non-linéarité généralisée :

$$\mathsf{NLG}(f) = 2^{n-1} - \frac{1}{2} \max_{\substack{\lambda \in \mathbb{F}_{2^n} \\ k:\mathrm{pgcd}(k,2^n-1)=1}} |\widehat{f}(\lambda, k)|$$

Définition 5.2. *On dit que f est hypercourbe si et seulement si $\widehat{f}(\lambda, k) = \pm 2^n$ pour tout $\lambda \in \mathbb{F}_{2^n}$ et pour tout k tel que* $\mathrm{pgcd}(k, 2^n - 1) = 1$.
De manière équivalente, f est hypercourbe si et seulement si $f(x^k)$ est courbe pour tout k tel que $\mathrm{pgcd}(k, 2^n - 1) = 1$.

Cependant, un premier problème pratique se pose. Les fonctions de filtrage sont classiquement des fonctions booléennes à n bits en entrée, où n est très significativement inférieur à L la taille du LFSR. Ainsi, il est assez aisé de calculer la non-linéarité puisque l'on peut se restreindre à l'espace $\mathbb{F}_{2^n}$ pour la calculer. En revanche, calculer la non-linéarité généralisée devient plus ardu puisque le degré d'extension L de l'ordre de 256 bits rend le calcul de la représentation trace de f vue comme fonction booléenne à L variables très complexe. De même, trouver une fonction monôme proche de f n'est a priori pas évident et la recherche exhaustive est entièrement hors de portée étant données les tailles en pratique du problème considéré.
Il est à noter que la recherche d'une fonction monôme proche d'une fonction booléenne donnée est, dans notre cas, une part de précalcul. Ce n'est donc pas très grave si ce calcul est long, cependant pour $n = 256$, cela donnerait une complexité au moins de l'ordre de 2^{256} opérations, ce qui se rapproche de l'ordre de grandeur du nombre d'atomes dans l'univers ! Ainsi même en considérant qu'il s'agit de précalcul, il nous est nécessaire de comprendre mieux la structure algébrique des éléments du corps $\mathbb{F}_{2^n}$. Cela fera évidemment partie de mes futurs travaux.

Complexité linéaire et fonctions hypercourbes. Afin de résister aux attaques décrites ci-dessus, il est nécessaire de construire des fonctions hypercourbes. En revanche, beaucoup de ces fonctions ont été exhibées sous la forme $\mathsf{Tr}(\lambda x^d)$ avec évidemment $\mathrm{pgcd}(d, 2^n - 1) > 1$ [22]. Or, d'après tout le travail précédent, il est clair qu'il ne faut pas non plus utiliser des fonctions booléennes de ce type, car la complexité linéaire de la séquence générée par un LFSR filtré par une fonction de ce type serait très petite, de plus la période serait de taille strictement inférieure à $2^n - 1$. Dans ces conditions, il est nécessaire d'exhiber aussi des fonctions hypercourbes, dont le nombre de termes dans la représentation trace est important [23]. En 2006, il a été démontré le résultat [24] suivant :

Théorème 5.1. *Si f est une fonction hypercourbe à n variables (n pair), alors sa représentation trace est de la forme :*

$$f(x) = \sum_{k \in M(\lambda)} \mathsf{Tr}(A_k x^k)$$

où l'ensemble $M(\lambda)$ est caractérisé de la façon suivante :

$$M(\lambda) = \{k : \forall r : \mathrm{pgcd}(r, 2^n - 1) = 1, \mathsf{w}_H(rk) = \frac{n}{2}\}$$

Dans ces conditions, le nombre de termes dans la représentation trace semble limité, ce qui dans notre cas nous donnera une suite de complexité linéaire limitée.

Remarque 5.1. *Cependant nous retombons sur le problème que nous avions eu au tout début, c'est-à-dire le lien entre le poids de deux entiers et le poids du produit de deux entiers quand on considère la multiplication modulo* $2^n - 1$. *Il serait aussi intéressant de calculer une borne sur le nombre d'exposants possibles différents dans la représentation trace des fonctions hypercourbes. Cela semble être fortement corrélé au nombre de diviseurs de* $2^n - 1$.

Par ailleurs, du point de vue cryptographique, les fonctions hypercourbes ne sont pas très intéressantes dans notre contexte dans la mesure où, comme toute fonction courbe, elles ne sont pas équilibrées. Il faudrait donc utiliser des fonctions équilibrées dont la non-linéarité généralisée soit presque optimale.

5.3 D'autres corrélations

Pour l'instant, nous nous sommes bornés à regarder ce qu'il se passait lorsque $\text{pgcd}(k, 2^n-1) = 1$. Maintenant, nous allons nous intéresser aux différents cas où notre fonction booléenne est proche d'une fonction monomiale dont l'exposant n'est pas nécessairement premier avec $2^n - 1$.

Considérons toujours un LFSR de taille n, de racine primitive α et d'état initial X_0, filtré par une fonction booléenne f. On suppose maintenant qu'il existe $\lambda \in \mathbb{F}_{2^n}\backslash\{0\}$ et k non nécessairement premier avec $2^n - 1$ tel que $d(f, \mathsf{Tr}^{C_k}(\lambda x^k)) = N$ où C_k est la taille de la classe cyclotomique de k.
Réutilisons les notations précédentes : f' la fonction booléenne définie par $f'(x) = f(x) + g(x)$ et $g(x) = \mathsf{Tr}(\lambda x^k)$. Alors le système représenté figure 4 à droite est équivalent au système initial, puisqu'il produit la même séquence si on initialise les LFSR respectivement avec l'état initial X_0 et X_0^k.

En revanche, comme k n'est pas nécessairement premier avec $2^n - 1$, le polynôme minimal de α^k qui définit le LFSR non filtré du dessus n'est plus nécessairement primitif.

Notation 5.1. *On notera* $\tau(x)$ *l'ordre de l'élément* x, τ_k *l'ordre de l'élément* α^k *où* α *est une racine primitive et* $D(x)$ *l'ordre de* 2 *modulo* $\tau(x)$, *i.e. le plus petit entier* d *tel que* $2^d \equiv 1 \mod \tau(x)$, *de même on notera* D_k *l'ordre de* 2 *modulo* τ_k.

Avec ces notations, le polynôme minimal de α^k qui définit cet LFSR est de degré D_k. Une attaque par corrélation classique visant à retrouver l'état initial de ce LFSR coûterait

$$O\left(\frac{\tau_k \log(\tau_k)}{\varepsilon^2}\right) \text{ en temps et } O\left(\frac{\log(\tau_k)}{\varepsilon^2}\right) \text{ en données}$$

où ε est le biais entre la fonction f et la fonction g, alors qu'une attaque par corrélation rapide nous coûterait la complexité donnée par la formule (1) page 9 avec $L = D_k$. Pour aller plus loin, il serait aussi intéressant de regarder si l'on ne peut pas mieux exploiter la structure afin d'améliorer la complexité de ces attaques.
Dans tous les cas, nous récupérons la valeur de l'élément X_0^k. Or, comme k n'est pas premier avec $2^n - 1$, nous ne pouvons pas retrouver immédiatement la valeur de X_0.

Lemme 5.1. *La connaissance de* X_0^k *nous donne* $\log_2(\tau_k)$ *bits d'informations sur* X_0.

Preuve. X_0 est un élément non nul du corps $\mathbb{F}_{2^n}$ et α en est une racine primitive du corps. Nous pouvons donc assurer qu'il existe un unique $i \in [0, 2^n - 2]$ tel que $X_0 = \alpha^i$.
Soit k un entier compris entre 1 et $2^n - 2$. On réalise alors la division euclidienne de i par τ_k. Alors, il existe un unique couple d'entiers (q, r) tels que :

$$X_0 = \alpha^{q\tau_k}\alpha^r$$

avec $r < \tau_k$. Donc

$$X_0^k = \alpha^{qk\tau_k}\alpha^{rk}$$

Or par définition de τ_k, nous avons $\alpha^{k\tau_k} = 1$, donc

$$X_0^k = \alpha^{rk}$$

Donc nous connaissons r puisqu'il n'existe qu'un seul $r \in [0, \tau_k - 1]$ tel que $X_0^k = \alpha^{rk}$. En effet s'il existe r_1 et r_2, $r_1 \geq r_2$ tel que $\alpha^{r_1 k} = \alpha^{r_2 k}$, alors $\alpha^{(r_2 - r_1)k} = 1$. Par définition de τ_k (le plus petit j tel que $\alpha^{kj} = 1$), on a τ_k qui divise $r_2 - r_1$, or $r_2 - r_1 \in [0, \tau_k - 1]$ donc $r_2 = r_1$.
Donc en écrivant $X_0 = \alpha^i$, la connaissance de X_0^k nous donne le reste de la division euclidienne de i par τ_k. Dans ces conditions, nous avons $\log_2(\tau_k)$ bits d'information sur X_0. □

Pour conclure, s'il existe $\lambda \in \mathbb{F}_{2^n}\backslash\{0\}$ et k tel que $d(f, \mathsf{Tr}^{C_k}(\lambda x^k)) = N$, alors avec le coût d'une attaque par corrélation classique en $O\left(\frac{\tau_k}{\varepsilon^2}\right)$ ou d'une corrélation rapide avec $L = D_k$, nous retrouvons $\log_2(\tau_k)$ bits d'information sur X_0.
De plus, une recherche exhaustive sur les quotients possibles de la division euclidienne nous donnera l'état initial du LFSR. Cette recherche exhaustive a un coût de l'ordre de

$$\frac{2^n - 1}{\tau_k}$$

opérations. Finalement, le coût total de l'attaque nous permettant de retrouver X_0 est de :

$$\frac{2^n - 1}{\tau_k} + \frac{\tau_k \log(\tau_k)}{\varepsilon^2}$$

sachant qu'il est possible d'améliorer le terme de droite si l'on peut utiliser une attaque par corrélation rapide.

Combinaison d'attaques par corrélation. On suppose maintenant que nous avons réalisé deux attaques par corrélation distinctes comme précédemment. Plus précisément, on suppose qu'il existe λ_1 et λ_2 dans $\mathbb{F}_{2^n}\backslash\{0\}$ et k_1 et k_2 dans $[1, 2^n - 2]$ tels que $d(f, \mathsf{Tr}^{C_{k_1}}(\lambda_1 x^{k_1})) = N_1$ et $\mathsf{Tr}^{C_{k_2}}(\lambda_2 x^{k_2}) = N_2$.
Alors, par deux attaques par corrélation, il est possible de retrouver $X_0^{k_1}$ et $X_0^{k_2}$.
Nous écrivons X_0 de la manière suivante :

$$X_0 = \alpha^i = \alpha^{q_1\tau_{k_1}}\alpha^{r_1} \text{ et } X_0 = \alpha^i = \alpha^{q_2\tau_{k_2}}\alpha^{r_2}$$

c'est à dire :

$$i \equiv r_1 \mod \tau_{k_1} \text{ et } i \equiv r_2 \mod \tau_{k_2}$$

avec k_1,k_2,r_1 et r_2 connus suite à la réalisation des attaques par corrélation. Il est clair qu'une solution à ce problème existe, et dans ces conditions, par le théorème des restes chinois, nous pouvons affirmer que nous connaissons le reste dans la division euclidienne de i par $\text{ppcm}(k_1, k_2)$, le meilleur cas étant lorsque k_1 et k_2 sont premiers entre eux.
En effet, en supposant que $\text{pgcd}(k_1, k_2) = 1$, nous récupérons alors $\log_2(k_1 k_2)$ bits d'information sur X_0, avec un coût total de l'attaque qui est la somme du coût de deux attaques par corrélation.
Ce résultat se généralise facilement à autant de k_j que l'on veut, on essaiera de les prendre premiers entre eux deux à deux afin de ne pas recouper l'information que l'on pourrait avoir en commun dans les différentes attaques par corrélation. De plus, on priviligera les k_j tels que $\tau_{k_j} = 2^d - 1$ où nécessairement $d|n$, afin de pouvoir réaliser des attaques par corrélation rapides. Si on n'est pas dans ce cas là, alors le LFSR sur lequel on réalise notre corrélation aura un polynôme non primitif. Une attaque par corrélation classique visera alors à lister tous les états internes possibles générés par ce LFSR, alors qu'une attaque par corrélation rapide n'exploitera pas le fait que le polynôme de rétroaction n'est pas primitif et que donc il y a moins d'états internes possibles.

6 Conclusion

Nous avons exhibé de nouveaux critères à considérer lorsque l'on choisit la fonction booléenne qui filtre les registres dans les chiffrements à flots.
Premièrement, il faut que la fonction possède beaucoup de termes dans l'expression de sa représentation trace, afin que la complexité linéaire soit élevée, à la fois pour éviter de distinguer la suite d'une suite aléatoire, mais aussi afin d'éviter des attaques décrites en [21].
De plus, nous avons proposé une nouvelle attaque qui montre qu'il faut que la fonction booléenne soit loin en distance, à la fois des fonctions monomiales dont l'exposant est premier avec 2^n-1, ce qui nous impose d'utiliser des fonctions proches des fonctions hypercourbes. Cependant,il semble que le nombre de termes dans l'expression de ces fonctions soit très limité, ce qui est alors un problème au vu de la complexité linéaire des séquences.
De plus, il est aussi nécessaire que la fonction soit loin de toutes les autres fonctions booléennes monomiales. Dans ces conditions, un équilibre reste à trouver, afin que toutes les attaques aient environ la plus grande complexité possible.
Nous avons de plus soulevé un certain nombre de problèmes ouverts suite à ce travail :

- La relation entre les poids des exposants et les poids des exposants multiplié par des nombres premiers avec 2^n-1 semble complexe. Cependant, il serait très intéressant de travailler sur cette question, afin d'améliorer la borne trouvée à la section 3.2.2, mais aussi afin de mieux caractériser les fonctions hypercourbes.
- Lorsqu'un système est publié, la fonction booléenne est très souvent décrite par sa forme normale algébrique ou sa forme univariée, mais prenant en entrée un corps beaucoup plus petit. Ainsi, le calcul de la représentation trace de la fonction booléenne vue comme une fonction agissant sur le corps tout entier semble très complexe, et la taille rend toute recherche exhaustive hors de portée. Dans ces conditions, il faut continuer à travailler sur ce problème là, c'est à dire comment trouver la racine primitive qui rendra la fonction booléenne vulnérable en un temps de précalcul raisonnable.
- Les nouveaux critères à prendre en compte sont complexes. Il serait donc intéressant de généraliser ce qui a été fait, afin de simplifier les critères à prendre en compte pour donner une "bonne" fonction booléenne. Par exemple, les complexités des dernières attaques dépendent de beaucoup de paramètres : à la fois τ_k, la distance entre la fonction et la fonction monôme, mais aussi la taille du sous corps. De plus, il est clair que si 2^n-1 possède beaucoup de facteurs dans sa décomposition en facteurs premiers, alors nous pourrons utiliser cela afin de réaliser la dernière attaque décrite. En revanche, si 2^n-1 possède peu de facteurs premiers, alors ces dernières attaques ne seront que peu efficaces, mais en revanche le nombre de fonctions appropriées semble être encore plus limité.

Références

[1] Claude E. Shannon. Communication theory of secrecy systems. *Bell system Technical Journal*, 28, 1949.

[2] An Braeken, Joseph Lano, Nele Mentens, Bart Preneel, and Ingrid Verbauwhede. Sfinks : A synchronous stream cipher for restricted hardware environments. *submission to eSTREAM*, 2008.

[3] Sondre Rønjom and Carlos Cid. Nonlinear equivalence of stream ciphers. In *Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers*, pages 40–54, 2010.

[4] Amr M. Youssef and Guang Gong. Hyper-bent functions. In *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*, pages 406–419, 2001.

[5] Neal Zierler. Linear recurring sequences. *J. Soc. Indus. Appl. Math., 7 :31-48*, 1959.

[6] Robert J. McEliece. *finite Fields for Computer Scientists and Engineers*, volume 23 of *SECS*. Kluwer Academic Publishers, 1987.

[7] James Lee Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, vol. 15 :122–127, Janvier 1969.

[8] E. R. Berlekamp. Factoring polynomials over finite fields. *Bell Syst. Tech. J.*, vol. 46 :1853–1859, 1967.

[9] Nicolas Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, pages 176–194, 2003.

[10] Nicolas Courtois and Willi Meier. Algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, pages 345–359, 2003.

[11] Yassir Nawaz, Guang Gong, and Kishan Chand Gupta. Upper bounds on algebraic immunity of boolean power functions. In *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers*, pages 375–389, 2006.

[12] Thomas Siegenthaler. Decrypting a class of stream ciphers using ciphertext only. *IEEE Trans. Computers*, 34(1) :81–85, 1985.

[13] Willi Meier and Othmar Staffelbach. Fast correlation attacks on stream ciphers. In *Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings*, pages 301–314, 1988.

[14] Anne Canteaut. Fast correlation attacks against stream ciphers and related open problems. In *Proceedings of the 2005 IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security - ITW 2005*, pages 49–54. IEEE Press, 2005.

[15] Philippe Chose, Antoine Joux, and Michel Mitton. Fast correlation attacks : An algorithmic point of view. In *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, pages 209–221, 2002.

[16] Anne Canteaut and Michaël Trabbia. Improved fast correlation attacks using parity-check equations of weight 4 and 5. In *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, pages 573–588, 2000.

[17] Sihem Mesnager. Bent and hyper-bent functions in polynomial form and their link with some exponential sums and Dickson polynomials. *IEEE Transactions on Information Theory*, 57(9) :5996–6009, 2011.

[18] Gohar M. M. Kyureghyan and Valentin Suder. On inversion in F_{2^n-1}. *Finite Fields and Their Applications*, 25 :234–254, 2014.

[19] Anne Canteaut. *Encyclopedia of Cryptography and Security, 2nd Ed, page 720.* Springer, 2011.

[20] Edwin L. Key. An analysis of the structure and complexity of nonlinear binary sequence generators. *IEEE Transactions on Information Theory*, 22(6) :732–736, 1976.

[21] Guang Gong, Sondre Rønjom, Tor Helleseth, and Honggang Hu. Fast discrete Fourier spectra attacks on stream ciphers. *IEEE Transactions on Information Theory*, 57(8) :5555–5565, 2011.

[22] Pascale Charpin and Guang Gong. Hyperbent functions, Kloosterman sums, and Dickson polynomials. *IEEE Transactions on Information Theory*, 54(9) :4230–4238, 2008.

[23] Sihem Mesnager. Hyper-bent Boolean functions with multiple trace terms. In *Arithmetic of Finite Fields, Third International Workshop, WAIFI 2010, Istanbul, Turkey, June 27-30, 2010. Proceedings*, pages 97–113, 2010.

[24] A. A. Nechaev A. S. Kuzmin, V. T. Markov and A. B. Shishkov. Approximation of boolean functions by monomial ones. *Discrete Math. Appl.*, 16(1) :7–28, 2006.

Annexe A : Calcul de la complexité linéaire

Nous donnons ici la preuve de la proposition 4.1.

Soit une séquence $(s_t)_{t\geq 0}$ de période $2^n - 1$. Alors elle est générée par un LFSR filtré de taille n de polynôme primitif P (définissant un élément primitif α) et de fonction de filtrage f. On notera dans la suite ce LFSR : $LFSR_0$ (par opposition aux transformations que l'on fera ultérieurement). On commence par écrire la fonction de filtrage sous sa représentation Trace :

$$f(x) = \sum_{k\in\Gamma_n} \mathsf{Tr}_1^{C_k}(A_k x^k)$$

où Γ_n est l'ensemble des représentants des classes cyclotomiques, C_k est la taille de la classe cyclotomique de k, $A_j \in \mathbb{F}_{2^{C_k}}$.

On isole maintenant chaque terme monomial dans l'écriture de la fonction et on considère que la séquence est produite par la composition de ℓ LFSR filtrés chacun de polynôme P, et de fonction de filtrage $\mathsf{Tr}_1^{C_{k_i}}(A_{k_i}x^{k_i})$ notées chacunes $(f_i)_{i\in[\![1,\ell]\!]}$ où ℓ est le nombre d'éléments présents dans l'écriture de la fonction. Donc la séquence est produite par la composition de ces générateurs :

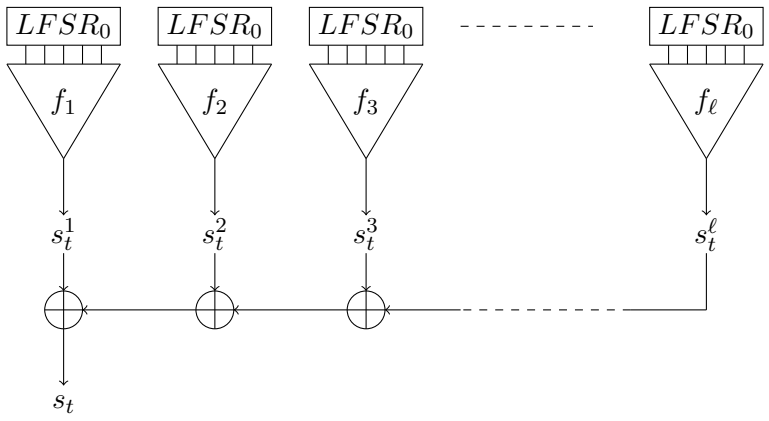

FIGURE 5 – Registre filtré vu comme combinaison de registres filtrés par un monôme

Or, il est clair que, vu l'expression de chaque fonction f_i, chaque suite $(s_t^i)_{t\geq 0}$ peut être produite par le LFSR (non filtré) de polynôme de rétroaction le polynôme minimal (non nécessairement primitif) de l'élément α^{k_i} noté $LFSR_i$.

En effet, la suite $(s_t^i)_{t\geq 0}$ est définie par la relation suivante :

$$s_t^i = \mathsf{Tr}_1^{C_{k_i}}(A_{k_i}x^{k_i t})$$

D'après le lemme 3.1, nous pouvons considérer que $A_{k_i} = 1$. Avec ce qui a été dit en 3.2.1, nous pouvons assurer que le bloc "LFSR filtré" est équivalent à un LFSR seul dont le polynôme de rétroaction est le polynôme minimal de α^{k_i}.

Lemme A.1. *La taille de $LFSR_i$ est la taille de la classe cyclotomique de k_i.*

Preuve. La taille du sous-corps auquel appartiennent les éléments $(\alpha^{k_i t})_{t\geq 0}$ produits par $LFSR_i$ est le plus petit entier d tel que $2^d \equiv 1 \mod \tau$ où τ est l'ordre de α^{k_i}. Or C_{k_i} est la taille de la classe cyclotomique de k_i et est donc par définition le plus petit entier r tel que $\alpha^{k_i 2^r} = \alpha^{k_i}$ i.e. $2^r \equiv 1 \mod \tau$. Donc la taille de la classe cyclotomique de k_i donne la taille du sous-corps auquel appartiennent les éléments $(\alpha^{k_i t})_{t\geq 0}$. Autrement dit, C_{k_i} est la taille du plus petit LFSR qui génère la séquence $(s_t^i)_{t\geq 0}$. □

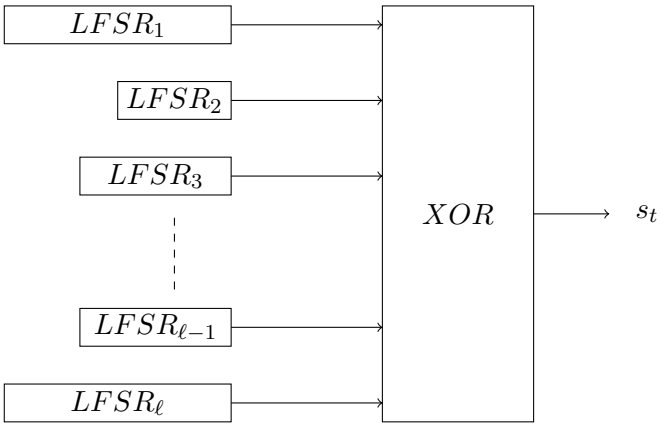

FIGURE 6 – Système de chiffrement à flot vu comme combinaison de LFSR

Ainsi, la suite initiale $(s_t)_{t\geq 0}$ est produite par le système représenté à la figure 6.
Cependant, comme tous les k_i sont les représentants distincts des classes cyclotomiques, il en découle que les polynômes minimaux qui définissent nos ℓ LFSR sont premiers entre eux deux à deux. En effet, le polynôme minimal de α^{k_i} est le polynôme

$$P_{\alpha^{k_i}}(X) = \prod_{j=0}^{C_{k_i}-1} (X - \alpha^{k_i 2^j})$$

Soient i et i', $i \neq i'$. On suppose que $P_{\alpha^{k_i}}$ et $P_{\alpha^{k_{i'}}}$ ont un facteur commun, i.e. il existe j et j' tels que $k_i 2^j = k_{i'} 2^{j'}$, on arrive à une contradiction puisque k_i et $k_{i'}$ sont dans deux classes cyclotomiques distinctes. Donc les ℓ LFSR sont premiers entre eux deux à deux.
De plus, pour tout $1 \geq i \geq \ell$ chaque $LFSR_i$ est le plus petit LFSR qui génère la séquence $(s_t^i)_{t\geq 0}$, donc $\Lambda((s_t^i)_{t\geq 0}) = C_{k_i}$. Ainsi, d'après les propriétés de la complexité linéaire sur la combinaison de générateurs (section 4), on en déduit une expression de la complexité linéaire de notre suite $(s_t)_{t\geq 0}$:

$$\Lambda(s) = \sum_{k\in\Gamma_n, A_k\neq 0} C_k$$

où Γ_n est l'ensemble des représentants des classes cyclotomiques, C_k est la taille de la classe cyclotomique de k, et A_k est le coefficient dans la représentation trace (unique) de f. □