Ciphertext only Reconstruction of LFSR-based Stream Ciphers

Anne Canteaut, Eric Filiol

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Ciphertext Only Reconstruction of LFSR-based Stream Ciphers

Anne Canteaut et Eric Filiol

No 3887

Février 2000

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--3887--FR+ENG

Ciphertext Only Reconstruction of LFSR-based Stream Ciphers

Anne Canteaut* et Eric Filiol†

Thème 2 — Génie logiciel
et calcul symbolique
Projet Codes

Rapport de recherche n ° 3887 — Février 2000 — 27 pages

Abstract: This report presents an operational reconstruction technique of most stream ciphers. We primarily expose it for systems where several linear feedback shift registers (LFSR) are combined by a nonlinear Boolean function. With only short ciphertexts, it is shown how to completely recover the different feedback polynomials and the combining function, when the algorithm is totally unknown. Estimates of necessary cipherlength and experimental results are detailed.

Key-words: stream cipher, Boolean functions, correlation, linear feedback shift register, ciphertext only reconstruction, unknown algorithm

(Résumé : tsvp)

* Projet Codes - Anne.Canteaut@inria.fr
† Ecoles Militaires de Saint-Cyr Coëtquidan, DGER/CRECSC, 56 381 GUER Cédex, efiliol@mailhost.esm-stcyr.terre.defense.gouv.fr

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : 01 39 63 55 11 - International : +33 1 39 63 55 11
Télécopie : (33) 01 39 63 53 30 - International : +33 1 39 63 53 30

Reconstruction à chiffré seul des systèmes de chiffrement à flots

Résumé : Ce document présente une technique opérationnelle de reconstruction pour la plupart des systèmes de chiffrement à flot. La démonstration est essentiellement faite sur la classe des systèmes constitués de plusieurs registres à décalage combinés par une fonction booléenne. En utilisant seulement des messages chiffrés relativement courts, il est montré comment complètement reconstruire les différents polynômes de rebouclage et la fonction de combinaison, quand le système est totalement inconnu. Des résultats expérimentaux et de complexité sont donnés.

Mots-clé : attaque à chiffré seul, chiffrements à flot, correlation, fonctions booléennes, registres à décalage, systèmes inconnus

1 Introduction

Stream ciphers are an important class of cipher systems. They are widely used by the world's militaries [18] and governmental offices. They also are very often used in industrial encryption products. The success of stream ciphers comes from the fact that they are very easy to build (LFSR are in fact simple arrays of bit memories and xor gates). They need only few logic gates in VLSI circuitry. They finally offer a very high security level (suitable for governmental applications) at low prices. They are particularly well suited as soon as encryption is needed in embedded systems (satellite for example) or in systems for which maintenance is either impossible or very difficult. All this implies that (due to the cost of the design), these systems, like many ciphers systems, have a very long life (at least twenty years, even more in some cases). Moreover, their use is particularly well suited when errors may occur during the transmission because they avoid error propagation.Most practical designs center around *linear feedback shift register* (LFSR) combined by a nonlinear Boolean function. In fact, different variants can be found (clock-controlled systems, filter generator, multiplexed systems...[16]) but almost all can be proved more or less equivalent (by algorithm transformation) to the most common class of combination generators. Consequently, we will focus on this very generic class.

The other very important aspect is that the designs are often secret [18] and contrary to block ciphers, generally no public evaluation is possible. In other words, it is possible to suspect "biased evaluation" (for reasons ranging from insufficient experience in stream ciphers design to backdoor implementation like a correlated combining function, for example, or irreducible but not primitive polynomials).

The problem of attacking the cipher becomes quite impossible without the algorithm. During World War II, US cryptanalysts had to face this problem with the Japanese PURPLE machine [10]: they reconstructed it before cryptanalysing it. This paper presents a similar approach and a reconstruction technique of stream ciphers allowing, from ciphertexts only, complete recovering of the unknown algorithm. By algebraic and statistical results, we show how to recover all the cryptographic primitives (LFSRs and the combining function) constituting the system, thus allowing evaluation or cryptanalysis. Recovering only the LFSRs is already in itself of very high importance since an

attack has been shown possible [15] without the knowledge of the combining function.

The reconstruction has been conducted on the following basis and assumptions:

- We use only ciphertexts, possibly generated from different secret keys. Each of them, however, must be of a realistic length.
- We accept very long computing time since work is done only once (and for all) and as long as it remains far lower than the life of the algorithm itself.
- We know the plaintext encoding (or at least some of its statistical parameters) and the linguistic group of the plaintext language.
- The system is a combination generator: Most practical designs use combining functions with up to 5 or 7 variables (*i.e.* registers). In this paper we will only consider additive stream ciphers but generalization to other combining functions can be envisaged with suitable modifications.

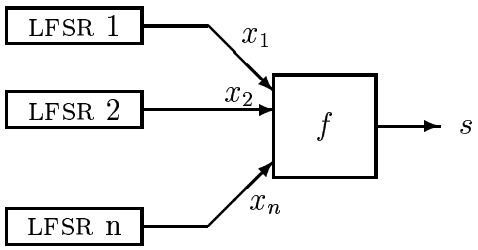

Figure 1: Nonlinear combination generator

This paper is organized as follows. Section 2 presents the theoretical tools we use in the reconstruction. Section 3 shows how to recover the LFSR and gives experimental results, bounds on complexity and ciphertext length. Section 4 deals with the combining functions recovering. Section 5 outlines generalization to other kind of LFSR-based stream ciphers.

2 Theoretical background

2.1 Linear Feedback Shift Register sequences

A linear feedback shift register of length L is characterized by L binary connection coefficients $(p_i)_{1\leq i\leq L}$. It associates to any L-bit initialization $(s_t)_{1\leq t\leq L}$ a sequence $(s_t)_{t>0}$ defined by the L-th order linear recurrence relation (see Figure 2)

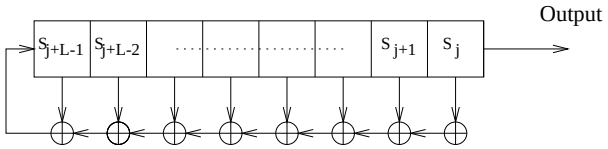

Figure 2: Linear Feedback Shift Register

$$s_{t+L} = \sum_{i=1}^{L} p_i s_{t+L-i}, \quad t \geq 0 \ .$$

The connection coefficients are usually represented by a univariate polynomial P over $\mathbf{F}_2$, called *the feedback polynomial*:

$$P(X) = 1 + \sum_{i=1}^{L} p_i X^i \ .$$

Most applications use a primitive feedback polynomial since this ensures that the periods of all sequences produced by the LFSR are maximal.

We now recall some well-known properties on LFSR sequences. In the following, $\mathcal{S}(P)$ denotes the set of all sequences produced by the LFSR with feedback polynomial P.

Proposition 1 *[7, 17, 19] Let P and Q be two non constant polynomials over $\mathbf{F}_2$. Then we have*

- $\{(u_t + v_t)_{t>0},\ u \in \mathcal{S}(P), v \in \mathcal{S}(Q)\} = \mathcal{S}(R)$ *where R is the least common multiple of P and Q.*

- $\{(u_t v_t)_{t>0},\ u \in \mathcal{S}(P), v \in \mathcal{S}(Q)\} = \mathcal{S}(R)$ *where* $\deg(R) \leq \deg(P)\deg(Q)$. *Equality holds if and only if at least one of the polynomials P and Q has only simple roots and all products $\alpha\beta$ are distinct for all α and β such that $P(\alpha) = 0$ and $Q(\beta) = 0$ in a common splitting field. This condition is notably satisfied if P and Q have coprime orders.*

A lower bound on the degree of R can also be deduced from the multiplicities of the roots of P and Q and from the number of distinct products $\alpha\beta$ [5].

Proposition 2 [12, Th. 8.53] *Let P and Q be two non constant polynomials over $\mathbf{F}_2$. Then $\mathcal{S}(P)$ is a subset of $\mathcal{S}(Q)$ if and only if P divides Q.*

This proposition implies that if a sequence s is generated by a LFSR with feedback polynomial P, then it satisfies the recurrence relations (also called parity-check equations) corresponding to PQ for any $Q \in \mathbf{F}_2[X]$.

For a given feedback polynomial P of degree L, we focus on all multiples of P of weight d where d is small. This approach is similar to fast correlation techniques [1, 8, 14]. The following formula (see e.g. [1]) provides an approximation of the average number $m(d)$ of multiples Q of P which have weight d and degree at most D $Q(X) = 1 + \sum_{j=1}^{d-1} X^{i_j}$:

$$m(d) \simeq \frac{D^{d-1}}{(d-1)!2^L} \ . \tag{1}$$

2.2 Boolean functions for stream ciphers

A Boolean function with n variables is a function from the set of n-bit words, $\mathbf{F}_2^n$, into $\mathbf{F}_2$. Such a function can be expressed as a unique polynomial in x_1, $\ldots$, x_n, called its *Algebraic Normal Form* (ANF):

$$f(x_1, \ldots, x_n) = \sum_{u \in \mathbf{F}_2^n} a_u x^u, \quad a_u \in \mathbf{F}_2$$

where $u = (u_1, \ldots, u_n)$ and $x^u = x_1^{u_1} x_2^{u_2} \ldots x_n^{u_n}$. The coefficients a_u of the ANF can be obtained from the Möbius transform of f [13]:

$$a_u = \bigoplus_{x \preceq u} f(x) \tag{2}$$

where $\alpha \preceq \beta$ describes the partial ordering on the Boolean lattice. This means that $\alpha \preceq \beta$ if and only if $\alpha_i \leq \beta_i$ for all $1 \leq i \leq n$.

The *Walsh-Hadamard transform* of a Boolean function f refers to the Fourier transform of the corresponding sign function, $x \mapsto (-1)^{f(x)}$:

$$\forall u \in \mathbf{F}_2^n, \quad \widehat{\chi}_f(u) = \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)} (-1)^{u \cdot x}$$

where $u \cdot x$ denotes the usual scalar product. The Walsh coefficient $\widehat{\chi}_f(u)$ then estimates the Hamming distance between f and the affine function $u \cdot x + \varepsilon$, $\varepsilon \in \mathbf{F}_2$, both seen as Reed-Muller codewords [13]:

$$d_H(f, u \cdot x + \varepsilon) = 2^{n-1} - \frac{(-1)^\varepsilon}{2} \widehat{\chi}_f(u) \ .$$

A Boolean function is obviously completely characterized by its Walsh spectrum. The coefficients of the algebraic normal form of f can then be computed from its Walsh coefficients as follows.

Proposition 3 *Let f be a Boolean function with n variables and let $(a_u)_{u \in \mathbf{F}_2^n}$ denote the coefficients of its algebraic normal form, i.e.,*

$$f(x_1, \ldots, x_n) = \sum_{u \in \mathbf{F}_2^n} a_u x^u \ .$$

Then we have for all $u \in \mathbf{F}_2^n$,

$$a_u = 2^{wt(u)-1} \left(1 - \frac{1}{2^n} \sum_{v \preceq \bar{u}} \widehat{\chi}_f(v) \right) \bmod 2$$

where $\bar{u}$ denotes the bitwise completion to 1 and $wt(u)$ denotes the Hamming weight of u, i.e., the number of its non-zero components.

Proof.
From Equation (2) we have for any $u \in \mathbf{F}_2^n$

$$\begin{aligned} a_u &= \sum_{x \preceq u} f(x) \bmod 2 = \sum_{x \preceq u} \frac{1}{2} \left(1 - (-1)^{f(x)}\right) \bmod 2 \\ &= 2^{wt(u)-1} - \frac{1}{2} \sum_{x \preceq u} (-1)^{f(x)} \bmod 2 \end{aligned}$$

Since the normalized Fourier transform is involutive, we have

$$\forall x \in \mathbf{F}_2^n, \quad (-1)^{f(x)} = 2^{-n} \sum_{v \in \mathbf{F}_2^n} \widehat{\chi}_f(v)(-1)^{v \cdot x} \ .$$

By combining these relations, we deduce that

$$\begin{aligned} a_u &= 2^{wt(u)-1} - 2^{-n-1} \sum_{x \preceq u} \sum_{v \in \mathbf{F}_2^n} \widehat{\chi}_f(v)(-1)^{v \cdot x} \bmod 2 \\ &= 2^{wt(u)-1} - 2^{-n-1} \sum_{v \in \mathbf{F}_2^n} \widehat{\chi}_f(v) \left(\sum_{x \preceq u} (-1)^{v \cdot x} \right) \ . \end{aligned}$$

The set $E_u = \{x \in \mathbf{F}_2^n, \ x \preceq u\}$ is a linear subspace of $\mathbf{F}_2^n$ of dimension $wt(u)$. Its orthogonal, $E_u^{\perp}$, satisfies $E_u^{\perp} = E_{\bar{u}}$. It follows that

$$\sum_{x \preceq u} (-1)^{v \cdot x} = \begin{cases} 2^{wt(u)} & \text{if } v \in E_{\bar{u}}, \\ 0 & \text{otherwise.} \end{cases}$$

We then obtain that, for all $u \in \mathbf{F}_2^n$,

$$a_u = 2^{wt(u)-1} - 2^{-n-1+wt(u)} \sum_{v \preceq \bar{u}} \widehat{\chi}_f(v) \bmod 2 \ .$$

□

This proposition will be used in the attack for recovering the algebraic normal form of the combining function.

It is well-known that a combining function must fulfill some criteria to yield a cryptographically secure combination generator (see e.g. [3]). A first obvious requirement is that the output of the combining function f be uniformly distributed. This corresponds to *balancedness*. Note that a Boolean function f is balanced if and only if $\widehat{\chi}_f(0) = 0$.

In order to increase the linear complexity of the produced pseudo-random sequence, the algebraic normal of f should have a high degree.

Another usual criterion is that f should be far from all affine functions regarding Hamming distance. The existence of a good approximation of f by an

affine function makes fast correlation attacks feasible [1, 8, 9]. The Hamming distance between f and the set of affine functions, called the *nonlinearity* of f, is given by

$$\mathcal{NL}(f) = 2^{n-1} - \frac{1}{2} \max_{u \in \mathbf{F}_2^n} |\widehat{\chi}_f(u)| \ .$$

Finally, the combination generator is vulnerable to correlation attacks [21] if the output of the combining function statistically depends on one of its inputs. More generally, Siegenthaler [20] introduced the following criterion:

Definition 1 *A Boolean function is* t-th order correlation-immune *if the probability distribution of its output is unaltered when any t input variables are fixed.*

This property equivalently asserts that the output of f is statistically independent of any linear combination of t input variables. The correlation-immunity order of a function can be characterized by its Walsh spectrum:

Proposition 4 [23] *A Boolean function f is t-th order correlation-immune if and only if*

$$\forall u \in \mathbf{F}_2^n,\ 1 \leq wt(u) \leq t,\ \ \widehat{\chi}_f(u) = 0 \ .$$

Since any t-th order correlation-immune function is k-th order correlation-immune for any $k \leq t$, we call correlation-immunity order of a function f the highest integer t such that f is t-th order correlation-immune.

Note that the correlation-immunity order of a function with n variables can not exceed $(n-1)$. This comes from Parseval's relation:

$$\sum_{u \in \mathbf{F}_2^n} \left(\widehat{\chi}_f(u)\right)^2 = 2^{2n} \ .$$

This equality also points out the existence of a trade-off between the correlation-immunity order and the non-linearity of a function. The higher is the correlation-immunity order, the lower may be the nonlinearity. The correlation-immunity order t of a Boolean function f with n variables also provides an upper bound on its degree [20]: $\deg(f) \leq n - t$. Moreover, if f is balanced, we have $\deg(f) \leq n - t - 1$.

3 Recovering the LFSRs

We now show how the key-stream generator depicted in Figure 3 can be reconstructed from the knowledge of some ciphertext bits. In the rest of the paper

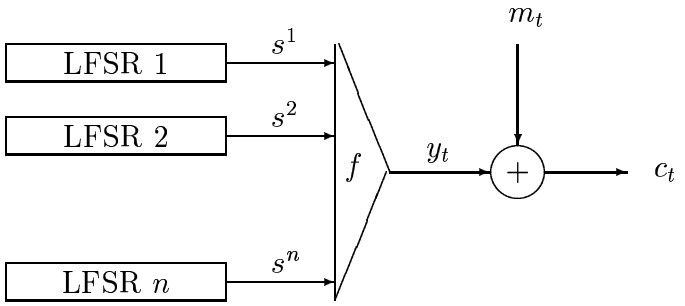

Figure 3: Additive stream cipher using a combination generator

we use the following notation. n denotes the number of constituent LFSRs. L_i and P_i denote the length and the feedback polynomial of the i-th LFSR and s^i refers to the generated sequence. The sequences y, m and c respectively correspond to the key-stream, to the plaintext and to the ciphertext. When dealing bitwise, we use t as index time.

The plaintext is assumed to be the output of a binary memoryless source with $P[m_t = 0] = p_0 \neq \frac{1}{2}$. All commonly used coding scheme (ASCII, Murray, CCITTx ...) satisfy this hypothesis. Moreover, the value of p_0 is supposed to be known. Practical overall values of p_0 are usually greater than 0.6 and lower than 0.7.

The first step of the reconstruction consists in recovering the feedback polynomials of the constituent LFSRs.

3.1 Statistical model

We first point out that the knowledge of a sequence s which is correlated with the ciphertext sequence provides some information on the feedback polynomials of the constituent LFSRs.

Proposition 5 *Let s be a binary sequence. If $P[c_t = s_t] \neq 1/2$ then there exists a Boolean function g with n variables such that $s = g(s^1, \ldots, s^n)$. Moreover, we have*

$$P[c_t = s_t] = 1 - p_0 - p_g + 2p_0p_g$$

where $p_g = P[f(x_1, \ldots, x_n) = g(x_1, \ldots, x_n)]$.

Proof.
We obviously have

$$\begin{aligned} P[c_t = s_t] &= P[y_t = s_t]P[m_t = 0] + P[y_t = s_t \oplus 1]P[m_t = 1] \\ &= 1 - p_0 - P[y_t = s_t] + 2p_0P[y_t = s_t] \ . \end{aligned}$$

By hypothesis, $p_0 \neq 1/2$. Thus $P[c_t = s_t] \neq 1/2$ implies that $P[y_t = s_t] \neq 1/2$. Since $y = f(s^1, \ldots, s^n)$, the sequences y and s are statistically independent if s is statistically independent of $(s^1, \ldots, s^n)$. It follows that $P[y_t = s_t] = 1/2$ unless $s = g(s^1, \ldots, s^n)$ for some Boolean function g. In this case, we have

$$P[y_t = c_t] = P[f(x_1, \ldots, x_n) = g(x_1, \ldots, x_n)] \ .$$

□

Note that some variables may not appear in the algebraic normal form of g.

If s is such that $P[c_t = s_t] \neq 1/2$ we deduce from the previous proposition and from Proposition 1 that the feedback polynomial of s is related to the feedback polynomials $P_1, \ldots, P_n$.

Corollary 1 *Let $\mathcal{S}(Q)$ denote the set of all sequences generated by $Q \in \mathbf{F}_2[X]$. If there exists $s \in \mathcal{S}(Q)$ such that $P[c_t = s_t] \neq 1/2$, then there exists a divisor Q' of Q and a Boolean function g such that Q' is derived from $P_1, \ldots, P_n$ and g as described in Proposition 1.*

This result leads to the following algorithm for recovering some information on $P_1, \ldots, P_n$. Let $\mathcal{Q}$ be a subset of $\mathbf{F}_2[X]$. For each $Q \in \mathcal{Q}$, we determine whether $\mathcal{S}(Q)$ contains a sequence which is correlated with the ciphertext. If such a sequence exists, Q provides some information on $P_1, \ldots, P_n$.

We here choose for $\mathcal{Q}$ the set of all polynomials of $\mathbf{F}_2[X]$ of weight d and of degree at most D having the following form $Q(X) = 1 + \sum_{j=1}^{d-1} X^{i_j}$.

Recall that the degree of the feedback polynomial of the product of two sequences s^i and s^j is roughly $L_i L_j$. It is then usually much higher than the degree of the feedback polynomial of $s^i + s^j$. If the upper-bound D on the degree of the examined polynomials is well-chosen, the polynomials Q detected by the algorithm correspond to the case where the combining function g is linear. For $g(x) = u \cdot x$, any feedback polynomial of $s = g(s^1, \ldots, s^n)$ is a multiple of $\mathrm{lcm}_{i \in supp(u)} P_i$ where $supp(u) = \{i, u_i = 1\}$. Since all feedback polynomials are usually primitive, we have $\mathrm{lcm}_{i \in supp(u)} P_i = \prod_{i \in supp(u)} P_i$ in most practical situations. Moreover, we have

$$P[c_t = s_t] = \frac{1}{2} + \frac{(2p_0 - 1)}{2^{n+1}} \widehat{\chi}_f(u) \ . \tag{3}$$

Example *We consider the combination generator described by Geffe [4]. This generator consists of three LFSRs combined by the Boolean function f:*

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

Assume that the feedback polynomials of the constituent LFSRs are randomly chosen primitive polynomials and that their lengths are respectively $L_1 = 15$, $L_2 = 17$ and $L_3 = 23$. Let c be the ciphertext sequence obtained by adding the output of the Geffe generator to a plaintext with $p_0 \neq 0.5$. Let $\mathcal{Q}$ be the set of all polynomials of weight 4 and of degree at most 10000. For all $Q \in \mathcal{Q}$, we determine whether $\mathcal{S}(Q)$ contains a sequence which is correlated with c. Let P be a randomly chosen polynomial of degree L. We deduce from Formula (1) that $\mathcal{Q}$ contains a multiple of P of weight 4 if $L \leq 37$. Our algorithm is then expected to detect multiples of P_1, P_2, P_3 and $P_1 P_2$. Note that P_2 can not be detected by the algorithm since the Walsh coefficient $\widehat{\chi}_f(010)$ vanishes.

A simple method for determining whether $\mathcal{S}(Q)$ contains a sequence which is correlated with c consists in computing the parity-check equation corresponding to Q for the ciphertext bits. The efficiency of this procedure strongly depends on the weight of Q.

Theorem 1 *Let Q be a polynomial in $\mathbf{F}_2[X]$ of weight d having the following form*

$$Q(X) = 1 + \sum_{j=1}^{d-1} X^{i_j} \text{ with } i_1 < i_2 < \ldots < i_{d-1} \ .$$

For a given ciphertext subsequence $(c_t)_{t<N}$ we consider the binary sequence $(z_t)_{i_{d-1} \le t < N}$ defined by $z_t = c_t \oplus \bigoplus_{j=1}^{d-1} c_{t-i_j}$. Then the random variable $Z = \sum_{t=i_{d-1}}^{N-1} (-1)^{z_t}$ has a Gaussian distribution with mean value

$$M = \pm (N - i_{d-1})(2\varepsilon)^d$$

and with variance

$$\sigma^2 = (N - i_{d-1})(1 - (2\varepsilon)^{2d})$$

where $\varepsilon = \max_{s \in \mathcal{S}(Q)} |P[c_t = s_t] - \frac{1}{2}|$.

Proof.
Let $s \in \mathcal{S}(Q)$ be such that $|P[c_t = s_t] - \frac{1}{2}|$ is maximal. Let $p = P[c_t = s_t]$. For all t, we decompose c_t as $c_t = s_t \oplus e_t$ where $P[e_t = 1] = 1 - p$. Then we have

$$P[z_t = 1] = P[c_t \oplus \bigoplus_{j=1}^{d-1} c_{t-i_j} = 1] = P[e_t \oplus \bigoplus_{j=1}^{d-1} e_{t-i_j} = 1]$$

since s satisfies the parity-check equation because $s \in \mathcal{S}(Q)$. This implies that $z_t = 1$ if and only if the number of indexes $i \in \{t, t - i_1, \ldots, t - i_{d-1}\}$ such that $e_i = 1$ is odd. Therefore we have

$$\begin{aligned} P[z_t = 1] &= \sum_{\ell=0, \ell \text{ odd}}^{d} \binom{d}{\ell} (1-p)^\ell p^{d-\ell} \\ &= \frac{1}{2} \left[\sum_{\ell=0}^{d} \binom{d}{\ell} (1-p)^\ell p^{d-\ell} - \sum_{\ell=0}^{d} \binom{d}{\ell} (p-1)^\ell p^{d-\ell} \right] \\ &= \frac{1}{2} \left[1 - (2p-1)^d \right] \ . \end{aligned}$$

The random variable Z can now be expressed as $Z = (N - i_{d-1}) - 2\sum_{t=i_{d-1}}^{N} z_t$. All random variables z_t are independent and identically distributed. Due to the central limit theorem [2], the random variable $\sum_{t=i_{d-1}}^{N} z_t$ for large values of $(N - i_{d-1})$ can be assumed to have a Gaussian distribution with mean value $(N - i_{d-1})P[z_t = 1]$ and variance $(N - i_{d-1})P[z_t = 1]P[z_t = 0]$. It follows that Z has a Gaussian distribution with mean value

$$M = (N - i_{d-1})(1 - 2P[z_t = 1]) = (N - i_{d-1})(2p - 1)^d$$

and with variance

$$\begin{aligned} \sigma^2 &= 4(N - i_{d-1})P[z_t = 1]P[z_t = 0] \\ &= (N - i_{d-1})(1 - (2p-1)^d)(1 + (2p-1)^d) \\ &= (N - i_{d-1})(1 - (2p-1)^{2d}) \ . \end{aligned}$$

□

If all sequences in $\mathcal{S}(Q)$ are statistically independent of c, Z has Gaussian distribution with mean value 0 and variance $(N - i_{d-1})$ since $\varepsilon = 0$ in this case.

We now want to distinguish between two hypotheses:

- $\mathcal{H}_0$: for all $s \in \mathcal{S}(Q)$, $P[c_t = s_t] = \frac{1}{2}$.
- $\mathcal{H}_1$: there exists $s \in \mathcal{S}(Q)$ such that $P[s_t = c_t] \neq \frac{1}{2}$.

From the previous theorem, we have

$$\begin{aligned} P[Z = x \mid \mathcal{H}_0] &= \frac{1}{\sqrt{2\pi(N - i_{d-1})}} \exp\left(-\frac{x^2}{2(N - i_{d-1})}\right) \\ P[Z = x \mid \mathcal{H}_1] &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - M)^2}{2\sigma^2}\right) \end{aligned}$$

We use a decision threshold T, $T > 0$, for discriminating hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$. If $|Z| < T$, $\mathcal{H}_0$ is kept; if $|Z| \geq T$, $\mathcal{H}_1$ is accepted. The minimum number of required ciphertext bits, N, depends on the number of wrong decisions that we allow. This number corresponds to the probability for a false alarm, $P_f =$

$P[|Z| \geq T \mid \mathcal{H}_0]$. The decision threshold is determined by the probability for a non-detection, $P_n = P[|Z| < T \mid \mathcal{H}_1]$. Let Φ denotes the normal distribution function,

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{x^2}{2}\right) dx \ .$$

Then we have

$$\begin{aligned} P_f &= P[|Z| \geq T \mid \mathcal{H}_0] = 2 \int_{-\infty}^{-T} P[Z = x \mid \mathcal{H}_0] dx \\ &= \frac{2}{\sqrt{2\pi(N - i_{d-1})}} \int_{-\infty}^{-T} \exp\left(-\frac{x^2}{2(N - i_{d-1})}\right) dx = 2\Phi\left(\frac{-T}{\sqrt{N - i_{d-1}}}\right) \end{aligned}$$

Similarly the probability for a non-detection is given by

$$\begin{aligned} P_n &= P[|Z| < T \mid \mathcal{H}_1] = \frac{1}{\sqrt{2\pi}\sigma} \int_{-T}^{T} \exp\left(-\frac{(x - M)^2}{2\sigma^2}\right) dx \\ &= \frac{1}{\sqrt{2\pi}} \int_{\frac{-T-M}{\sigma}}^{\frac{T-M}{\sigma}} \exp\left(-\frac{x^2}{2}\right) dx = \Phi\left(\frac{T - M}{\sigma}\right) - \Phi\left(\frac{-T - M}{\sigma}\right) \\ &= \Phi\left(\frac{T - |M|}{\sigma}\right) - \Phi\left(\frac{-T - |M|}{\sigma}\right) \end{aligned}$$

since M is not necessarily positive. In most cases, $\Phi(\frac{-T-|M|}{\sigma})$ is much smaller than P_n and than $\Phi(\frac{T-|M|}{\sigma})$. Then this latter will approximate P_n. The predetermined value of P_n fixes the choice for the threshold:

$$T = |M| + \Phi^{-1}(P_n)\sigma = (N - i_{d-1})(2\varepsilon)^d + \Phi^{-1}(P_n)\sqrt{(N - i_{d-1})(1 - (2\varepsilon)^{2d})} \ .$$

Similarly, the predetermined probability for a false alarm gives the minimum value of$(N - i_{d-1})$:

$$N - i_{d-1} = \left(\frac{T}{\Phi^{-1}(1 - \frac{P_f}{2})}\right)^2 \ .$$

After different attempts to tune up the best values for P_f and P_n, we choose $P_f = 2^{-20}$ and $P_n = 10^{-3}$. In practical situations the known ciphertext

sequence does not consist of a large number of consecutive bits. The attacker has access to some ciphertext blocks of reasonable lengths. These ciphertexts may be produced with different keys, i.e., with different LFSR initializations. Theorem 1 can nevertheless be adapted to this more realistic situation.

Corollary 2 *Let Q be a polynomial in $\mathbf{F}_2[X]$ of weight d having the following form*

$$Q(X) = 1 + \sum_{j=1}^{d-1} X^{i_j} \text{ with } i_1 < i_2 < \ldots < i_{d-1} \ .$$

For n_c ciphertexts c^k, $1 \leq k \leq n_c$, of respective lengths $LC(k)$, we consider the binary sequence $(z_t^k)_{i_{d-1} \leq t < LC(k), 1 \leq k \leq n_c}$ defined by

$$z_t^k = c_t^k \oplus \bigoplus_{j=1}^{d-1} c_{t-i_j}^k \ .$$

Then the random variable $Z = \sum_{k=1}^{n_c} \sum_{t=i_{d-1}}^{LC(k)-1} (-1)^{z_t^k}$ has a Gaussian distribution with mean value $M = \pm(2\varepsilon)^d \sum_{k=1}^{n_c} (LC(k) - i_{d-1})$ and with variance $\sigma^2 = (1 - (2\varepsilon)^{2d}) \sum_{k=1}^{n_c} (LC(k) - i_{d-1})$ where $\varepsilon = \max_{s \in \mathcal{S}(Q)} |P[c_t = s_t] - \frac{1}{2}|$.

The following algorithm then examines all polynomials of degree at most D and of weight d, and it detects all polynomials Q in this set such that there exists $s \in \mathcal{S}(Q)$ with $|P[s_t = c_t] - 1/2| \geq \varepsilon_{\min}$.

Algorithm

For each $(d-1)$-tuples $(i_1, \ldots, i_{d-1})$ such that $0 < i_1 < \ldots < i_{d-1} < D$

$N \leftarrow \sum_{k=1}^{n_c} (LC(k) - i_{d-1})$.

$T \leftarrow N(2\varepsilon_{\min})^d - 3\sqrt{N(1 - (2\varepsilon_{\min})^{2d})}$.

$Z \leftarrow 0$.

For each ciphertext block $(c_t^k)_{0 \leq t < LC(k)}$ where $LC(k) > i_{d-1}$

for each t from i_{d-1} to $LC(k) - 1$

$z \leftarrow c_t^k \oplus \bigoplus_{j=1}^{d-1} c_{t-i_j}^k$.

$Z \leftarrow Z + (-1)^z$.

If $|Z| \geq T$, factor $1 + \sum_{j=1}^{d-1} X^{i_j}$ and store all its primitive factors. Store Z.

The primitive factors which are detected several times by the algorithm are expected to be the feedback polynomials of the constituent LFSRs.

3.2 Complexity analysis

We now discuss the choice of the input parameters d, D and $\varepsilon_{\min}$.

Recall that we aim at recovering multiples of polynomials $\prod_{i\in T} P_i$, $T \subset \{1,\ldots,n\}$ such that $|P[c_t = \bigoplus_{i\in T} s_t^i] - 1/2| \geq \varepsilon_{\min}$. According to Formula (3), these subsets T are characterized by

$$\frac{|2p_0 - 1|}{2^{n+1}} |\widehat{\chi}_f(1_T)| \geq \varepsilon_{\min}$$

where the i-th component of 1_T equals 1 if and only if $i \in T$. It is well-known that all Walsh coefficients of a Boolean function f with n variables are divisible by 4, unless f has degree n. This case is here dismissed since such a function cannot be balanced. Choosing

$$\varepsilon_{\min} = \frac{|2p_0 - 1|}{2^{n-1}} \tag{4}$$

then ensures to detect all polynomials $\prod_{i\in T} P_i$ such that $\widehat{\chi}_f(1_T) \neq 0$. In most practical situations, the number of variables n does not exceed 7.

We now assume that our search can be restricted to all products $\prod_{i\in T} P_i$ of degree at most $L_{\max}$. This means that we suppose that all feedback polynomials $P_1, \ldots, P_n$ can be recovered from all products $\prod_{i\in T} P_i$ such that $\widehat{\chi}_f(1_T) \neq 0$ and $\sum_{i\in T} L_i \leq L_{\max}$. Note that $L_{\max}$ should obviously be greater than the maximum length of all constituent LFSRs. A polynomial of degree $L_{\max}$ is then recovered by our algorithm if it divides at least one polynomial of weight d and of degree at most D. We deduce from Formula (1) that the minimum possible value for D is approximatively

$$D = (d-1)!^{\frac{1}{d-1}} 2^{\frac{L_{\max}}{d-1}} \ . \tag{5}$$

This also implies that the attack can only use ciphertext blocks of length at least LC with

$$LC \geq (d-1)!^{\frac{1}{d-1}} 2^{\frac{L_{\max}}{d-1}} \ . \tag{6}$$

Moreover, we want the probability for a false alarm in the algorithm to be less than 2^{-20}. This implies that $(\sum_{k=1}^{n_c} LC(k)) - n_c D \geq \left(\frac{T}{5}\right)^2$. By replacing T by its value, we obtain the following condition

$$N_t - n_c D \geq \frac{1}{25}\left((N_t - n_c D)(2\varepsilon_{\min})^d - 3\sqrt{(N_t - n_c D)(1 - (2\varepsilon_{\min})^{2d})}\right)^2$$

where $N_t = \sum_{k=1}^{n_c} LC(k)$ is the total ciphertext length. We deduce that

$$N_t - n_c D \geq \frac{\left(5 + 3\sqrt{1 - (2\varepsilon_{\min})^{2d}}\right)^2}{(2\varepsilon_{\min})^{2d}} \ . \tag{7}$$

It finally follows that the total ciphertext length should satisfy

$$N_t \geq n_c (d-1)! ^{\frac{1}{d-1}} 2^{\frac{L_{\max}}{d-1}} + \frac{\left(5 + 3\sqrt{1 - (2\varepsilon_{\min})^{2d}}\right)^2}{(2\varepsilon_{\min})^{2d}} \ . \tag{8}$$

This value is minimal if $n_c = 1$, i.e., if all known ciphertext bits are consecutive. In this case, the minimum length of the ciphertext sequence required by the reconstruction is

$$N_t = \min_d \left[(d-1)! ^{\frac{1}{d-1}} 2^{\frac{L_{\max}}{d-1}} + \frac{\left(5 + 3\sqrt{1 - (2\varepsilon_{\min})^{2d}}\right)^2}{(2\varepsilon_{\min})^{2d}} \right] \ . \tag{9}$$

Figure **??** shows, for different values of $L_{\max}$, how N_t and the optimal value of d vary with $\varepsilon_{\min}$.

‘ In most practical situations, all ciphertext blocks have roughly the same length LC. The number n_c of such ciphertext blocks required by the reconstruction is then

$$n_c \geq \frac{\left(5 + 3\sqrt{1 - (2\varepsilon_{\min})^{2d}}\right)^2}{(2\varepsilon_{\min})^{2d}(LC - (d-1)^{\frac{1}{d-1}} 2^{\frac{L_{\max}}{d-1}})} \ .$$

We then use the algorithm with the value of d which minimizes this formula.

Example *Suppose that* $\varepsilon_{\min} = 0.1$ *and* $L_{\max} = 35$. *We use ciphertext blocks of length* $LC = 10,000$ *bits (i.e., about 1,200 ASCII characters). Condition (6) imposes that* $d \geq 4$. *The number of required ciphertext blocks is* $n_c = 6,109$ *for* $d = 4$ *and* $n_c = 69,083$ *for* $d = 5$. *Our algorithm examines all polynomials of weight 4 and of degree at most* $D = 5,910$.

The number of operations performed by the algorithm (without the factorization step) is roughly

$$\frac{D^{d-1}}{(d-1)!} d(N_t - n_c D) \ .$$

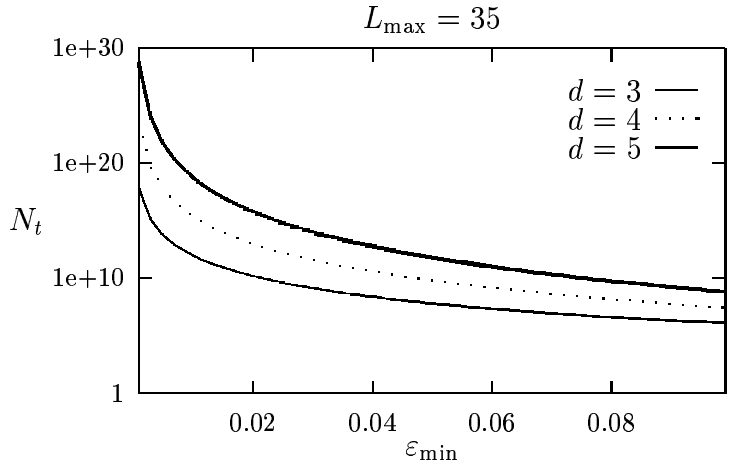

Figure 4: Minimum ciphertext length required for $L_{\max} = 35$

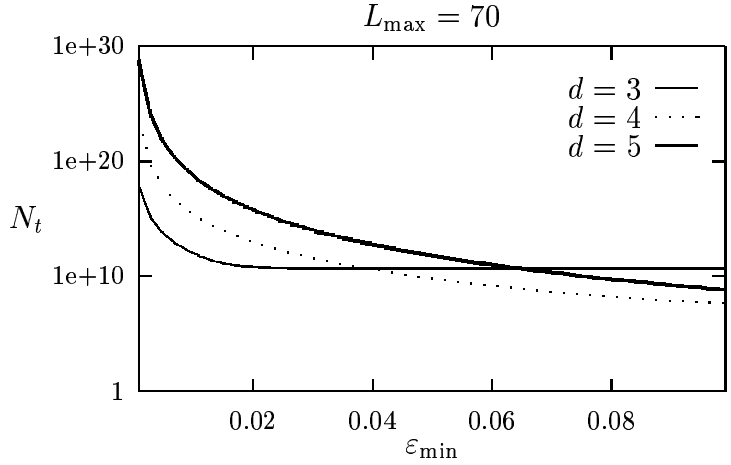

Figure 5: Minimum ciphertext length required for $L_{\max} = 70$

Using equations (5) and (8), we obtain the following complexity

$$\frac{d2^{L_{\max}}\left(5+3\sqrt{1-(2\varepsilon_{\min})^{2d}}\right)^2}{(2\varepsilon_{\min})^{2d}}\ .$$

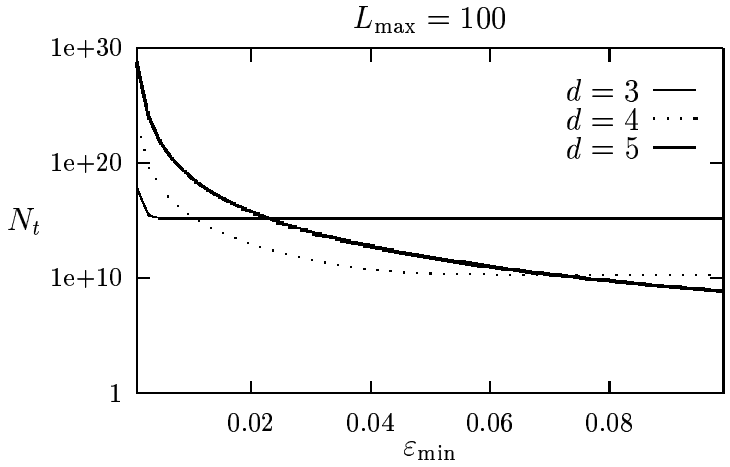

Figure 6: Minimum ciphertext length required for $L_{\max} = 100$

Another method for recovering the feedback polynomials of the LFSRs consists in examining all polynomials of degree at most $L_{\max}$ and in computing the corresponding parity-check equations on the ciphertext sequence. A similar analysis applies to this attack. We here have to choose $D = L_{\max}$ and $d \simeq L_{\max}/2$ since the average weight of a polynomial of degree $L_{\max}$ is roughly $L_{\max}/2$. With these parameters, Formula (7) provides the minimum ciphertext length required by this second attack:

$$N'_t = L_{\max} + \frac{\left(5 + 3\sqrt{1 - (2\varepsilon_{\min})^{L_{\max}}}\right)^2}{(2\varepsilon_{\min})^{L_{\max}}} \ .$$

Figure 7 shows that this number is much larger that the number of ciphertext bits required by our attack (see Formula (8)). Moreover, the number of operations performed by this second attack is roughly

$$\frac{d2^{L_{\max}}\left(5 + 3\sqrt{1 - (2\varepsilon_{\min})^{L_{\max}}}\right)^2}{2(2\varepsilon_{\min})^{L_{\max}}} \ .$$

Our attack is then much more efficient than the enumeration of all polynomials of degree $L_{\max}$.

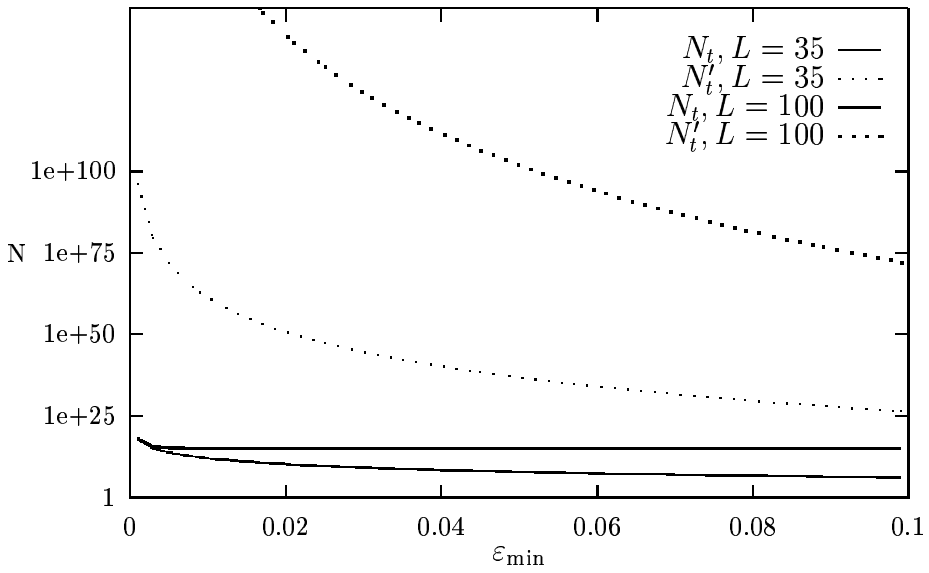

Figure 7: Minimum number of ciphertext bits required by both attacks for $L_{\max} = 35$ and 100

3.3 Simulation results

We considered the following toy example of combination generator. Three LFSRs are combined by the majority function

$$f(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3 \ .$$

The feedback polynomials are respectively

$$\begin{aligned} P_1(x) &= 1 + x^2 + x^4 + x^5 + x^6 + x^8 + x^9 + x^{10} + x^{11} + x^{13} + \\ &\quad x^{14} + x^{15} + x^{17} \\ P_2(x) &= 1 + x + x^7 + x^8 + x^{10} + x^{12} + x^{15} + x^{16} + x^{17} + x^{20} + x^{21} + \\ &= x^{22} + x^{23} \\ P_3(x) &= 1 + x + x^3 + x^6 + x^7 + x^8 + x^{13} + x^{16} + x^{19} + x^{20} + x^{25} + \\ &\quad x^{26} + x^{27} + x^{28} + x^{29} + x^{31} + x^{33} \end{aligned}$$

The output of this combination generator is used for encrypting a plaintext with $p_0 = 0.70$. We used 6,109 ciphertext blocks of length 10,000. We apply our algorithm on these ciphertext blocks with parameters $D = 5,910$ and $d = 3$ and 4. We take $\varepsilon_{\min} = 0.1$; this value corresponds to Formula (4) with $n = 3$. Table 1 gives all primitive polynomials detected by the algorithm for $d = 3$ and $d = 4$. The number of obtained multiples for this polynomial is compared with the theoretical values given by Formula (1). The theoretical number of

	d	Nb detected	P_1	P_2	P_3
simulations	3	126	123	3	0
theory	3	139	137	2	0
simulations	4	266,191	262,043	4,146	3
theory	4	266,589	262,483	4,102	4

Table 1: Detected polynomials for the toy example

operations performed by the algorithm is 2^{50} for $d = 3$ and 2^{61} for $d = 4$. Our simulation required 4 days to test all the genuine d-multiples and 0.5% of all the possible d-multiples on a Linux Pentium II 400. For each detected $P_i(x)$ we compute the 5-nomials on $D = 5,910$ bits for each product $P_i(x)P_j(x)$. Only $P_1(x)P_2(x)$ was potentially detectable according to Formula (1). It was not.

4 Recovering of the Combining Function

In [15] reconstruction of the combining function of such scheme has been exposed. But in fact, yet interesting, the approach suffers from severe limitations:

- recovering the function required to first recover all the initializations of the registers, that is to say to cryptanalyze the scheme.
- they use Siegenthaler attack [20] requiring exponential complexity.
- the function recovering explores the 2^n inputs and thus is of exponential complexity. Moreover it requires a consequent amount of ciphertexts.

We now present how to bypass these limitations and to practically reconstruct the combining function. From previous step, we obtained the following information on f: The number of variables of the combining function is derived from the previous step of our attack. Moreover, the previous analysis also provides an estimation of some Walsh coefficients of the combining function. Suppose that some multiples of weight d of $\prod_{i\in T} P_i$, $T \subset \{1,\ldots,n\}$, have been detected by our algorithm. For any such multiple, the mean value of the estimator Z equals $N(2p-1)^d$, where $p = P[c_t = s_t]$ with $s = g(s^1,\ldots,s^n)$ and $g(x) = 1_T \cdot x$. The values of Z obtained for all detected multiples of $\prod_{i\in T} P_i$ therefore provides an estimation of probability p. Using Formula (3), we can then compute the value of the corresponding Walsh coefficient, $\widehat{\chi}_f(1_T)$. This value is rounded to the closest multiple of 4, since all the Walsh coefficients are divisible by 4 for balanced functions.

If $\prod_{i\in T} P_i$ has degree L greater than $L_{\max}$, no multiple was detected by the algorithm. We then choose a higher value of d satisfying

$$(d-1)!^{\frac{1}{d-1}} 2^{\frac{L}{d-1}} \leq LC \ .$$

We then compute all multiples of $\prod_{i\in T} P_i$ of weight d and degree at most LC, and the corresponding values of Z. We deduce the involved Walsh coefficient as previously seen.

Example *In the toy example, the average values of the estimator Z (e.g. 12,384 for $P_1(x)$) obtained for each multiple of weight 3 of P_1 provide exactly*

$$P[z_t = 0] = 0.5005 \quad \textit{and } P[s_t = y_t] = 0.55.$$

Formula (3) gives the result: $\widehat{\chi_f}(1,0,0) = 4.00$. Similarly, we obtain the following information during the first step:

$$\widehat{\chi_f}(0,1,0) = \widehat{\chi_f}(1,0,0) = 4 \quad \widehat{\chi_f}(0,0,0) = 0 \ .$$

For each detected P_i we compute some multiples of weight 5 and of degree at most 10,000 for each product P_iP_j. Although all of these products were potentially detectable, no one was detected; we then deduce that

$$\widehat{\chi_f}(1,1,0) = \widehat{\chi_f}(1,0,1) = \widehat{\chi_f}(0,1,1) = 0 \ .$$

Similar simulations for $d = 7$ allow to find the remaining coefficient:

$$\widehat{\chi_f}(1,1,1) = -4 \ .$$

5 Generalization to Complex Stream Ciphers

More complex LFSR-based stream ciphers are essentially designed on the same following structure.

- The pseudo-random engine generates n linear pseudo-random sequences. The LFSRs can be simple, clock-controlled, decimated, filtered,...
- Since the LFSRs are inherently linear, their output is linear too. They absolutly cannot be used directly as valid cryptopraphic primitive (otherwise powerful cryptanalyses are possible [24, 25]). That is why efficient practical designs include a module, breaking this linearity and insuring diffusion, propagation and confusion (NDPC module). In a combination generator, it is a simple yet strong Boolean function but it can be far more complex structures including multiplexers, permutators, ... However we can always reduce it to a simple Boolean function (from the black box point of view) whose inputs are the LFSRs outputs and outputing one bit of running key.
- Combination module combines one plaintext bit with one running key bit to produce one ciphertext bit. Generally it consists in bitwise xor. Once again different complex modules, always can be seen as simple Boolean functions which moreover are suitable to be included in the previous combining function.

Then all the problem of scheme equivalence to combination generators lies on the kind of LFSR constituent of the pseudo-random engine. Clock-controlled LFSRs have been shown equivalent to non clock-controlled ones [16, chapter 6]. We have the same equivalence for filtered LFSRs [21]. The decimated LFSRs remains an open problem. Almost all the others schemes can be reduced to the combination generator. Particularly, nonlinear binary sequence generators (i.e. when the feedback function is nonlinear) have been proven equivalent to combination generators [6, 11], increasing the scope of our technique.

6 Conclusion

We have described an operational reconstruction technique of LFSR-based stream ciphers, from only not too long ciphertexts. This technique requires a lot of computation time but far less than the expected life of the algorithm itself. Dedicated parallel computer should dramatically decrease the computation time. The main consequence is that hiding the algorithm (national home-made ciphers, commercial products,...) does not give more security since it can be reconstructed, except for decimated generators. This case is under investigation.

It is worth noting that even partial reconstruction (for example LFSR being missing) are interesting since attack can be performed on the obtained "noisy" equivalent to the original complete algorithm. Fast-correlation attack may in this case be successful.

Acknowledgement

This paper is dedicated to Colonel Max Mayneris, a very enthusiastic signal officer, who very early understood the importance of working on such subjects. We will have an everlasting thought for his influence. We hope he now enjoys his retirement in his beloved Pyrénées mountains.

References

[1] A. Canteaut, M. Trabbia, Improved Fast Correlation Attacks using Parity-check Equations of weight 4 and 5. Submitted.

[2] W. Feller, *An Introduction to Probability Theory*, Wiley, 1966

[3] E. Filiol, C. Fontaine, Highly Nonlinear Balanced Boolean Functions with a Good Correlation-Immunity, Advances in Cryptology - EUROCRYPT'98, LNCS 1403, Springer Verlag, 1998.

[4] P.R. Geffe How to protect Data with Ciphers that are Really Hard to Break, Electronics, pp 99–101, 1973.

[5] R. Göttfert, H. Niederrreiter, On the Minimal Polynomial of the Product of Linear Recurring Sequences, Finite Fields and Their Applications, 1(2), pp 204–218, 1995.

[6] E. J. Groth, Generation of Binary Sequences with Controllable Complexity, IEEE Transactions on Information Theory, vol. 17, pp 288-296, 1971.

[7] T. Herlestam, On Functions of Linear Shift Register Sequences, Advances in Cryptology - EUROCRYPT'85, LNCS 219 pp 119–129, Springer Verlag, 1986.

[8] T. Johansson, F. Jönsson, Improved Fast Correlation Attack on stream Ciphers via Convolutional Codes. Advances in Cryptology - EUROCRYPT'99, LNCS 1592, pp 347–362, Springer Verlag, 1999

[9] T. Johansson, F. Jönsson, Fast Correlation Attack based on Turbo Codes Techniques, Advances in Cryptology - EUROCRYPT'99, LNCS 1592, pp 347–362, Springer Verlag, 1999

[10] D. Kahn, *The Codebreakers: The Story of Secret Writings*, Macmillan Publishing Co, 1967

[11] E. L. Key, An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators, IEEE Transactions on Information Theory, Vol. 22(6), pp 732–736, 1976.

[12] R. Lidl, H. Niederreiter, *Finite Fields*, Encyclopedia of Mathematics and its Applications, Cambridge University Press, 1997.

[13] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Mathematical Library, North-Holland, 1977.

[14] W. Meier, O. Staffelbach, Fast Correlation Attack on certain Stream Ciphers, J. of Cryptology, pp 159–176, 1989.

[15] S. Palit, B. Roy, Cryptanalysis of LFSR-encrypted Codes with Unknown Combining Function, ASIACRYPT'99, Singapore, to appear in LNCS Series, Springer

[16] R.A. Rueppel, *Analysis and Design of Stream Ciphers*, Springer Verlag, 1986.

[17] R.A. Rueppel, O. Staffelbach, Products of Linear Recurring Sequences with Maximum Complexity, IEEE Transactions on Information Theory, Vol. 33(1), pp 124–131, 1987.

[18] B. Schneier, *Applied Cryptography*, 2nd edition, Wiley, 1996.

[19] E.S. Selmer, *Linear Recurrence Relation over Finite Fields*, Ph. D Thesis, University of Bergen, Norway, 1966.

[20] T. Siegenthaler, Correlation Immunity of Nonlinear Combining functions for Cryptographic Applications, IEEE Transactions on Information Theory, Vol. 35 Nr 5, September 1984, pp 776–780.

[21] T. Siegenthaler, Decrypting a Class of Stream Ciphers using Ciphertext Only, IEEE Transactions on Computers, C-34, 1, pp 81–84, 1895.

[22] T. Siegenthaler, Cryptanalysts Representation of Nonlinearly Filtered ML-Sequences, Advances in Cryptology - EUROCRYPT'85, LNCS 219, pp 103–110, Springer Verlag, 1986.

[23] G. Xiao, J. Massey, A Spectral Characterization of Correlation Immune Functions, IEEE Transactions on Information Theory, Vol 34, pp 569 – 571, may 1988.

[24] K. Zheng, M. Huang, On the Linear Syndrome Method in Cryptanalysis, Advances in Cryptology - Crypto'88, LNCS 405, pp 469–478, Springer Verlag, 1990.

[25] K. Zeng, C.H. Yang, T.R. Rao, An Improved Linear Syndrome Algorithm in Cryptanalysis with Applications, Advances in Cryptology - Crypto'90, LNCS 537, pp 34–47, 1991.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399