INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

An Algebraic Cryptanalysis of Nonlinear Filter Generators using Gröbner bases

Jean-Charles Faugère

— Gwénolé Ars

N° 4739

Février 2003

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--4739--FR+ENG

An Algebraic Cryptanalysis of Nonlinear Filter Generators using Gröbner bases

Jean-Charles Faugère*†
, Gwénolé Ars ‡

Thème 2 — Génie logiciel
et calcul symbolique
Projets SPACES

Rapport de recherche n° 4739 — Février 2003 — 21 pages

Abstract: This paper presents an algebraic cryptanalysis of nonlinear filter generator. A linear shift register of length L filtered by a non linear boolean function f of degree $deg(f)$ is equivalently described by a set of algebraic equations. More precisely, if N is the size of given output bits then we have a system of N algebraic equations of total degree $deg(f)$ in L variables. By solving this system of equations we can recover all the possible initial state (the secret key) of the device . Gröbner is precisely an efficient tool for solving algebraic systems. Recently, very efficient algorithms (F_5 [12]) have been proposed which are several order of magnitude faster than the historical Buchberger algorithm. We show that with only a polynomial number of output bits we can recover in polynomial time the initial state. More precisely we can show that is enough to have $\mathcal{O}(L^d)$ output bits with $d \leq \lfloor \frac{k+1}{2} \rfloor$ where k is the number of variables of the filtering function. Surprisingly, for all the stream ciphers satisfying Golič's design criteria and filtering functions found in literature we found that d is much less than the predicted bound: for instance the Lili is of degree 6 but a simple Gröbner computations shows that it behaves like a degree 4 function. Even more surprisingly, we show experimentally that for some examples we can recover the initial state in polynomial time with only $L + \epsilon$ output bits. Different attacks have been implemented, and we give a list of timing experimented on many real size size ($L \approx 80$ bits) stream ciphers

Key-words: Hidden Field Equations (HFE), Multivariate polynomial equations, Gröbner bases, Algebraic Cryptanalysis, Computer Algebra.

* LIP6/LORIA CNRS/UPMC/INRIA
† Projet SPACES
‡ LIP6 DGA/UPMC/Université Rennes 1

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy (France)
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

Cryptanalyse algébrique des registres filtrés par les bases de Gröbner

Résumé : Cet article présente une cryptanalyse algébrique des registres filtrés par une fonction booléene non linéaire. On peut modéliser un registre linéaire de taille L filtré par une fonction booléene f de degré $deg(f)$ par un système d'équations algébriques. Plus exactement, si N est le nombre de bits de la séquence de sortie on obtient un système de N équations de degré total $deg(f)$ en L variables. En résolvant ce système on retrouve l'état initial du registre c'est à dire la clé secrete. Les bases de Gröbner sont un outil très efficace pour résoudre les systèmes algébriques. Récemment, un algorithme (F_5) de calcul de bases de Gröbner a été proposé dont l'efficacité est de plusieurs ordre de grandeur supérieure à celle de l'algorithme historique de Buchberger. On montre qu'il suffit d'un nombre polynomial de bits pour retrouver l'état initial en temps polynomial. Plus précisément, on peut montrer qu'il suffit de $\mathcal{O}(L^d)$ bits de sortie avec $d \leq \lfloor \frac{k+1}{2} \rfloor$ où k est le nombre de variables de la fonction de filtrage. De manière surprenante, pour tous les generateurs pseudo aléoires et satisfaisant les critères de Golič que nous avons testés, nous avons trouvé un degré d encore plus petit que cette borne: par exemple pour Lili-128 on trouve des fonctions de degré 4. Encore plus surprenant, nous montrons expérimentallement que pour certains exemples nous pouvons retrouver efficacement l'état initial avec seulement $L + \epsilon$ bits de sortie. Différente attaques ont été implantées et nous donnons une liste de benchmarks avec des tailles réelles ($L \approx 80$ bits).

Mots-clés : Polynômes multivariés, Bases de Gröbner, Cryptanalyse algébrique, Calcul Formel.

1 Introduction

Stream ciphers are generally faster than block ciphers in hardware, and need a low power consumption. Furthermore, buffering is limited and in situations where transmission errors can occur the error propagation is limited. Consider a synchronous stream cipher in which the key-stream, the plain-text, and the cipher-text are sequences of binary digits: the output sequence of the key-stream generator, $(z_i)_{i\geq 0}$ is added bitwise to the plain-text sequence $(m_i)_{i\geq 0}$, producing the cipher-text $(c_i)_{i\geq 0}$. The key-stream generator is initialized through a secret key K, and hence, each key K will correspond to an output sequence. Since the key is shared between the transmitter and the receiver, the receiver can decrypt by adding the output of the key-stream generator to the cipher-text, obtaining the message sequence.

The goal is to efficiently produce random-looking sequences that are as "indistinguishable" as possible from truly random sequences. Also, from a cryptanalysis point of view, a good stream cipher should be resistant against different kind of attacks as a *known-plaintext attack*. For a synchronous stream cipher, a known-plain-text attack is equivalent to the problem of finding the key K that produce a given output key-stream $z_0, z_1, \ldots, z_{N-1}$. A common methodology for producing random-like sequences is to use linear feedback shift registers, LFSRs, with length L as building blocks in different ways and a filtering functions f to break the linearity of LFSRs. Furthermore, the secret key K is often chosen to be the initial state of the registers. We can describe of such a cryptosystem when N bits of the running-key is given by an algebraic system of N equations of degree $deg(f)$. In the rest of the paper $\mathcal{S}(f, N)$ denote this algebraic system. A first and obvious result is that if $N \approx L^{deg(f)}$, then the resolution of the cryptosystem is polynomial in $L^{deg(f)}$ (more precisely $\mathcal{O}(L^{\omega deg(f)})$ where $\omega \leq 3$ is the exponent of the complexity of linear algebra).

The main attacks on stream cipher are Correlation and Fast Correlation attacks [6, 16]. The essential principle of the attack by correlation is to assimilate the research of the initial state of a register to a problem of correction of errors. Hence we imagine that the output of the nonlinear filter generator results from the transmission produced by a register through a noisy channel. The errors which occur during this transmission come in fact from the filter function. The probability of error is thus higher if the two sequences, the output one and the register one, are slightly correlated. As the sequence generated by only one register is strongly redundant, we can reconstitute it using an algorithm of decoding which corrects the errors of transmission of a code $\mathcal{C}$.

At this time, to the best of the authors's knoweldge, the only other published and similar attack was presented by Courtois [8]. The idea is to *approximate*, with a good probability, the filtering function used in Toyocrypt by a low degree poynomial and to apply the XL algorithm [9] for solving the algebraic system. The method presented in [8] is well adapted for the Toyocrypt example but does not seem to be a general method.

In this paper we present a general and efficient method to find the initial state of the register: we simply compute a Gröbner basis of algebraic system $\mathcal{S}(f, N)$. The difficulty is then to evaluate the complexity of this computation.

As many other algorithms, Gröbner bases algorithms behave much better in practice than in the worst case. Algebraic systems coming from non linear filter generators are

typical examples: the computation of Gröbner bases over $\mathbb{F}_2$, has an asymptotic worst-case time bound which is exponential, while its running time is bounded by a low-degree polynomial in the particular case of stream generator. Hence, even if very efficient algorithms for computing Gröbner bases are now available (F_5 [12]), the drawback of Gröbner bases algorithms is that the complexity is *difficult to evaluate theoretically.*

We have seen than the previous trivial bound $\mathcal{O}(L^{\omega deg(f)})$ depends strongly on the degree of the algebraic normal form of f. But we can also represent the boolean function f by one or several *rational form* $f = \frac{N_f}{D_f}$ where N_f, and D_f are polynomials of degree d less than $deg(f)$. More precisely, we prove that $d \leq \lfloor \frac{k+1}{2} \rfloor$ where k is the number of variables of the filtering function. From this, if we have $N \approx L^d$ output bits, then the Gröbner basis computation reduces to simple linear algebra, so we find a complexity of $\mathcal{O}(L^{\omega d})$.

We have to make a careful distinction between the Gröbner computation of $\mathcal{S}(f, N)$ and the computation of the representation $f = \frac{N_f}{D_f}$ whose primary purpose is to evaluate precisely d (and thus N) for a particular f. This might be confusing since the computation of N_f and D_f is also the result of *another* Gröbner basis computation. However this second Gröbner basis computation (technically a basis of the ideal of relations of f) is very easy and has to be done only once for a given f. The first computation Gröbner is the difficult part and must be done for every given sequence of output bits.

Of course we have to make several computer simulations to *validate* the previous estimation. First we have collected a list of stream ciphers satisfying Golič's design criteria and filtering functions found in literature [15, 13, 5, 7, 6, 17]. For all the examples we found that the degree d is much less than the theoretical bound (and always less than 4 for all the examples). Thus we establish that for a reasonable number of given output bits (say several kbytes) we can recover efficiently the secret key for real size stream ciphers (80 bits). Even more surprisingly, we show experimentally that for some of the examples we can recover the initial state in polynomial time with only $L + \epsilon$ output bits.

This paper is organized as follows. Section 2 describes nonlinear filter generators and their properties. Section 3 introduces Gröbner bases and basic properties. Section 4 presents the rational representation of boolean functions and focuses on computational complexity bound of the attack. Section 5 gives a detailed list of running-times for many stream ciphers which confirm the theoretical approach. Section 7 compares our attack with correlation attacks. Section 8 proposes another strategy: for a fixed stream cipher compute a symbolic Gröbner basis in a precomputation phase; then the initial state can be found very efficiently by substituting actual values of the running-key in the generic Gröbner basis.

2 Nonlinear Filter Generator Description

We describe briefly a nonlinear filter generator. It is composed from a LFSR and a filtering function. We refer to [1, 2] for further details.

2.1 Linear Feedback Shift Registers (LFSR)

Let q be an integer, we define LFSR on the finite field $\mathbb{F}_q$. A linear feedback shift register produces a sequence, $\mathbf{u} = u_0, u_1, \ldots$, satisfying the linear recurrence relation, $u_n = \sum_{j=1}^{L} c_j u_{n-j}$, $n \geq L$ where L is the length of the LFSR, $c_j \in \mathbb{F}_q$ for $j = 1, \ldots, L$, and $u_i \in \mathbb{F}_q, i \geq 0$.

The L stages, $U_n = (u_n, u_{n+1}, \ldots, u_{n+L-1})$, is called a *state* of the shift register and we note $\mathbf{U} = (U_n)_{n=0}^{\infty}$ the state sequence. We define the *feedback polynomial* to be $P_F(X) = 1 - c_1 X - c_2 X^2 - \cdots - c_{L-1} x^{L-1} - c_L X^L$. The first L output symbols, $u_0, u_1, \ldots, u_{L-1}$, are initially loaded into the LFSR, these symbols are called the *initial state*. This is also the secret key of the LFSR.

The linear recurrence between the stages implies a relation between consecutive state: $U_{n+1} = A\, U_n$ where A is the companion matrix ofP_F.

The sequences $\mathbf{U} = U_0, U_1, \ldots$ preoduced by linear feedback registers have many interesting properties such as a very long periodicity. If the feedback polynomial P_f is primitive the period is $q^L - 1$.

The main drawback with LFSR sequences is that if we are given a sequence of L consecutive output symbols, then, due to the linearity, we would calculate the output symbol at an arbitrary time instance. So, one cannot use a maximum-length LFSR directly as a key-stream generator.

2.2 Non linear boolean function

Non linear boolean functions' purpose in key-stream generators is to hide the linearity introduced by the LFSRs. A boolean function is a function $f : \mathbb{F}_2^k \to \mathbb{F}_2$.

From an algebraic point of view, a boolean function f can be described by its *algebraic normal form* (ANF). Let $a_0, a_1, \ldots, a_k, a_{1,2}, \ldots, a_{1,2,\ldots,k}$ be some elements in $\mathbb{F}_2$ such that:

$$f(x_1, \ldots, x_k) = a_0 + \sum_{m=1}^{k} \sum_{1 \leq i_1, \cdots, i_m \leq k} a_{i_1, \cdots, i_m} x_{i_1} \ldots x_{i_m}$$

where addition and multiplication are in $\mathbb{F}_2$. The degree of the ANF of f is called the *degree* of f (and denoted by $deg(f)$).

As we want, on output, random-looking sequences, we need to restrict the choice of the boolean function. A k variables boolean function f is *balanced* if $\mathbb{P}(f(\mathbf{x}) = 0) = \mathbb{P}(f(\mathbf{x}) = 1) = \frac{1}{2}$, when $\mathbf{x}$ is chosen uniformly in $\mathbb{F}_2^k$.

2.3 The Nonlinear Filter Generators

The nonlinear filter generator is the combination of the LFSR with a boolean function. Again, if the boolean function, the feedback polynomial or the connection between the function and the LFSR are not chosen properly then the whole system may increase the risk of being attacked.

As the length L (say $L \approx 128$) of the LFSR and the number k (say $k \approx 10$) of variables of the boolean function f are usually different, we need to introduce the connexions between the two elements.

Let $\gamma = (\gamma_i)_{i=1}^{k}$ be an increasing sequence of nonnegative integers so that $\gamma_1 = 0$ and $\gamma_k \leq L-1$. Then the output sequence $\mathbf{z} = (z_n)_{n=0}^{\infty}$ of the nonlinear filter generator is defined by $z_n = f(u_{n+\gamma_1}, \ldots, u_{n+\gamma_k}), \quad n \geq 0$. The sequence γ is called the *tapping sequence* of the LFSR.

To avoid different attacks, Golic[14] proposes design criteria for nonlinear filter generators.

3 Algebraic Approach of Nonlinear Filter Generator

3.1 Algebraic model

We may suppose that f is a boolean function on L variables. With this notation, the output sequence can be written with the LFSR state expression $\mathbf{U} = (U_n)_{n=0}^{\infty} : \; z_n = f(U_n) = f(A^n U_0), \quad n \geq 0$

An algebraic attack of the nonlinear filter generator, if we know N bits of the key-stream, is to solve the system :

$$z_0 = f(U_0), z_1 = f(A.U_0), \cdots, z_{N-1} = f(A^{N-1}.U_0) \tag{1}$$

Gröbner basis provides an efficient tool for solving systems of polynomial equations. We associate to (1) the ideal J_N spanned by $J_N =< z_0 - f(X), z_1 - f(A.X), \ldots, z_{N-1} - f(A^{N-1}.X) >$ where X the vector $(x_0, \ldots, x_{L-1})$. The Gröbner basis computation gives a simpler list of generators of the ideal J_N.

The Gröbner basis of (1) give the solutions of this system in the algebraic closure $\overline{\mathbb{F}_2}$ of $\mathbb{F}_2$. Since we only want solutions in $\mathbb{F}_2$, we add the field equations $x_i^2 + x_i$ for $i = 1, \ldots, L$. Hence, we have to compute the Gröbner basis of

$$I_N =< x_0^2 + x_0, \ldots, x_{L-1}^2 + x_{L-1}, z_0 - f(x_0, \ldots, x_{L-1}), \ldots, z_{N-1} - f(A^{N-1}\,(x_0, \ldots, x_{L-1})) >$$

An equivalent method is to work in the ring $\mathcal{R} = \mathbb{F}_2[x_0, \ldots, x_{L-1}] / < x_0^2 + x_0, \ldots, x_{L-1}^2 + x_{L-1} >$, and to consider the ideal I_N in $\mathcal{R}$ (in $\mathcal{R}$, all polynomial have degree one in in each variable):

$$I_N =< z_0 - f(x_0, \ldots, x_{L-1}), \ldots, z_{N-1} - f(A^{N-1}\,(x_0, \ldots, x_{L-1})) > \tag{2}$$

In the rest of the paper $\mathcal{S}(f, N)$ denote this algebraic system. We can notice that if we take $N > L$, the algebraic system $\mathcal{S}(f, N)$ (2) is *over-defined*: it has more equations than variables.

3.2 Definitions of Gröbner bases

First we introduce some definitions on Gröbner bases. For more details, we refer to [4] or [10]. Let us consider a field $\mathbb{K}$.

Definition 1 *A* monomial ordering *on* $\mathbb{K}[x_1, \ldots, x_L]$ *is a relation* $\succ$ *on the set of monomials* x^α, $\alpha = (\alpha_1, \ldots, \alpha_L) \in \mathbb{Z}^L$, *satisfying :*

(i) $\succ$ *is a total ordering.*

(ii) *If* $x^\alpha \succ x^\beta$ *and* $\gamma \in \mathbb{Z}^L$ *then* $x^{\alpha+\gamma} \succ x^{\beta+\gamma}$.

(iii) $\succ$ *is a well-ordering, i.e. every nonempty subset of the monomial set has a smallest element under* $\succ$.

Example 1 *The DRL order (Degree Reverse Lexicographic order):*
Let x^α *and* x^β *two monomials. We say* $x^\alpha \succ x^\beta$ *if* $|\alpha| = \sum_{i=1}^{L} \alpha_i > |\beta| = \sum_{i=1}^{L} \beta_i$ *or* $|\alpha| = |\beta|$ *and* $\exists i \in \{1, \ldots, L\}$ *s.t.* $\alpha_i < \beta_i$ *and* $\forall i < j \leq L$, $\alpha_j = \beta_j$.
In practice, a DRL Gröbner basis leads to faster computation.

The elimination order $[x_0, \ldots, x_{L-1}]$, $[z_0, \ldots, z_{N-1}]$*: (we will use this order in section 8) For any monomial* m *we can write* $m = m_x\, m_z$ *where* m_x *(resp.* m_z*) depends only on* x_i *(resp* z_j*). We say that* $m = m_x\, m_z \gg m' = m'_x\, m'_z$ *if and only if* $m_x \succ m'_x$ *or* ($m_x = m'_x$ *and* $m_z \succ m'_z$*).*

Let $f = \sum_\alpha a_\alpha x^\alpha$ be a nonzero polynomial in $\mathbb{K}[x_1, \ldots, x_L]$, the leading monomial is $LM(f) = max_\succ\{x^\alpha\}$ and the leading coefficient is $LC(f) = a_\alpha$ so that $LM(f) = x^\alpha$.

Definition 2 *Let us fix a monomial ordering* $\succ$. *A finite subset* $G = \{g_1, \ldots, g_s\}$ *of an ideal* I *is said to be a* Gröbner basis *if* $\forall f \in I$, $\exists g \in G$ *such that* $LM(g)$ *divides* $LM(f)$.

Proposition 1 *(Specialization) Let* G *be a Gröbner basis for the elimination order* $[x_0, \ldots, x_{L-1}]$, $[z_0, \ldots, z_{N-1}]$.

If we substitute in G *the variables* $(z_i)_{i=0}^{N-1}$ *with explicit values, we still have a DRL order Gröbner basis of the ideal.*

Proposition 2 *For any monomial ordering, the (reduced) Gröbner basis of* I_N *defined by (2) for a specific sequence* $(z_i)_{i=0}^{N}$ *is:*

- $< x_0 + a_0, \ldots, x_{L-1} + a_{L-1} >$ *if* $(a_0, \ldots, a_{L-1})$ *is the only solution.*
- $< 1 >$ *if there is no solution.*

Another very useful property of Gröbner bases is that it is possible to find *all* the algebraic relations among several polynomials $f_1, \ldots f_m$ (see [10] page 338 for a precise definition of the ideal of relations). We will use this proposition to compute a "rational form" of a boolean function.

Proposition 3 *([10] page 340) Fix a monomial order in $\mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$ where any monomial involving one of the $x_1, \ldots, x_n$ is greater than all monomials in $k[y_1, \ldots, y_m]$ (an elimination ordering) and let G be the Gröbner basis for this ordering. Then $G \cap k[y_1, \ldots, y_m]$ describe all the relations among $f_1, \ldots, f_m$.*

The previous definition 2 does not depend on any algorithm. The first hirstorical algorithm for computing Gröbner bases was the Buchberger algorithm [10, 4]. More efficient algorithms are now available such as the algorithm F_4 [11] which introduces linear algebra in the computation. More recently, the algorithm F_5 [12] which avoids reduction to zero for generic systems. With a good implementation of this algorithm, we can solve real size stream cipher (see section 5).

3.3 Behavior of the System according to the Number of Equations

The following proposition proves that as long as we do not have the solution, every new equation gives useful informations.

Proposition 4 *Let us fix an output sequence $(z_i)_{i=0}^{\infty}$ from a nonlinear filter generator. Let I_N be defined as (2).*
$\exists N_0 \in \mathbb{N}$ so that for all $N \geq N_0$, $I_N = I_{N_0}$ and for all $0 \leq N < N_0$, $I_N \subsetneq I_{N+1}$.

Proof 1 *Since the LFSR is periodic with a period T, $z_{T-1} - f(A^{T-1}\mathbf{U}_0) = z_0 - f(\mathbf{U}_0)$ with $\mathbf{U}_0 = (x_0, \ldots, x_{L-1})$. So $I_T = I_{T-1}$.*
Let us consider $N_0 = min\{N \text{ s.t. } I_N = I_{N+1}\}$. For all $0 \leq N < N_0$, $I_N \subsetneq I_{N+1}$.
Since $I_{N_0} = I_{N_0+1}$, $z_{N_0} - f(A^{N_0}.\mathbf{U}_0) \in I_{N_0}$. So $\exists P$ a polynomial such that:

$$z_{N_0} - f(A^{N_0}\mathbf{U}_0) = P(z_0 - f(\mathbf{U}_0), \ldots, z_{N_0-1} - f(A^{N_0-1}\mathbf{U}_0), x_0^2 + x_0, \ldots, x_{L-1}^2 + x_{L-1})$$

With the change of variable $\mathbf{U}_0 \mapsto A\mathbf{U}_0 = (x_1, \ldots, x_{L-1}, x_0 + \sum_{i=1}^{L-1} \alpha_i x_i)$,

$$\begin{aligned} z_{N_0} - f(A^{N_0+1}\mathbf{U}_0) &= P(z_0 - f(A\mathbf{U}_0), \ldots, z_{N_0-1} - f(A^{N_0}\mathbf{U}_0), x_1^2 + x_1, \ldots, x_{L-1}^2 + x_{L-1}, \\ &\qquad (x_0 + \sum_{i=1}^{L-1} \alpha_i x_i)^2 + x_0 + \sum_{i=1}^{L-1} \alpha_i x_i) \\ z_{N_0+1} - f(A^{N_0+1}\mathbf{U}_0) &= \tilde{P}(z_1 - f(A\mathbf{U}_0), \ldots, z_{N_0} - f(A^{N_0}\mathbf{U}_0), \quad x_0^2 + x_0, \ldots, x_{L-1}^2 + x_{L-1}) \end{aligned}$$

So $z_{N_0+1} - f(A^{N_0+1}.(x_0, \ldots, x_{L-1})) \in I_{N_0}$, i.e. $I_{N_0+2} = I_{N_0+1} = I_{N_0}$.
By iteration, we deduce that $\forall N \geq N_0$, $I_N = I_{N_0}$.

4 Complexity of Gröbner Bases Computation

4.1 General Result

Recent algorithms for computing Gröbner bases (F_4, F_5) incrementally construct matrices in degree 2, 3, ... D:

$$A_D = \begin{array}{c} \\ m_1 \times f_{i_1} \\ m_2 \times f_{i_2} \\ m_3 \times f_{i_3} \\ \cdots \end{array} \begin{array}{c} \overset{momoms\ degree\ \leq\ D\ in\ x_1,\ldots,x_L}{} \\ \left(\begin{array}{c} \cdots \\ \cdots \\ \cdots \\ \cdots \end{array} \right) \end{array}$$

where $m_1, m_2, \ldots$ are monomials such that the total degree of $m_j f_{i_j}$ is less than D. The next step in the algorithm is to compute a row echelon of A_D using linear algebra techniques. It must be emphasized that the rows of A_D is a *small subset* of all the possible rows $\{mf_i$ s.t. $1 \leq i \leq N$ and m any monomials.t. $deg(m) \leq D - deg(f_i)\}$.

Consequently, to estimate the complexity of the computation of a Gröbner basis, we need to know the maximal degree of the polynomials occurring in the computation. Let d_{max} be this maximal degree then the whole complexity of the algorithm is equivalent to compute row echelon form of $A_{d_{max}}$ so that the complexity is $(\sum_{i=0}^{d_{max}} \binom{L}{i})^{\omega}$ where $\omega \leq 3$ is the complexity of linear algebra.

As we work in the ring $\mathcal{R}$, all the monomial have a degree 1 in each variables, hence the total degree of a monomial is less than L. It is therefore clear that the complexity of the Gröbner basis computation can be done in (single) exponential time. Of course this is only a rough upper bound and must be compared to the complexity of exhaustive search $\mathcal{O}(L\,2^L)$.

It is very difficult to evaluate more accurately the complexity of Gröbner basis computation and especially in the case of over determined systems of polynomial equations (that is to say when $N \gg L$ where N is the number of equations). We extract from [3] (see also [9]) this useful result:

Result 1 *Let $\mathcal{S}$ be an algebraic system of $N = L\log^2(L)$* random *equations in L variables, then a Gröbner basis can be computed in sub-exponential time ($\mathcal{O}(e^{\frac{L\log(\log(L))}{\log(L)}})$).*

In other word, for a nonlinear filter generator with LFSR length L, if the number N of known z_i is higher than $L\log^2(L)$ then we can expect to find the initial state of the generator in sub-exponential time in L. In view of the fact that we have *no proof* that the algebraic equations obtained from a non linear filter generator are *random* this result must also be validated by real computer simulations. Moreover, sub-exponential time is very close to exponential so this result does not tell us how the algebraic attack simplifies the resolution *in practice*. On the other hand, this result clearly indicates that the more equations you have the more able you are to compute the Gröbner basis and thus to recover the secret key.

4.2 Algorithm F_5 and Low Degree Relations

We have noticed by computer simulation that the Gröbner basis computation of the ideal (2) behaves as if the function f had a lower degree $d \leq deg(f)$. To understand this, we can examine in more detail the algorithm F_5.

The algorithm F_5[12] is an incremental algorithm on the elements of the ideal $I_N =< z_0 - f(x_0,\ldots,x_{L-1}),\ldots,z_{N-1} - f(A^{N-1}(x_0,\ldots,x_{L-1})) >$: the algorithm compute first

the DRL Gröbner basis of $< z_0 - f(x_0, \ldots, x_{L-1}) >$, and then a Gröbner basis of $< z_0 - f(x_0, \ldots, x_{L-1}), z_1 - f(A\,(x_0, \ldots, x_{L-1})) >$, and so on ...

So, in order to evaluate the complexity, we need to estimate first the degree of elements in the ideal of relation (see proposition 3) $< z_0 - f(x_0, \ldots, x_{L-1}) >$.

Proposition 5 *Let a be in $\mathbb{F}_2$. The reduced Gröbner basis of $I_a =< f(x_1, \ldots, x_k) + a >$ in $\mathcal{R}$ for the DRL order contains an independent linear basis of all the relation $g \in \mathcal{R}$ with minimal degree d so that:*

$$\forall (x_1, \ldots, x_k) \in \mathbb{F}_2^k, f(x_1, \ldots, x_k) = a \Rightarrow g(x_1, \ldots, x_k) = 0 \tag{3}$$

We note N_d the dimension of the vector space of relations g of degree d.

Proof 2 *Let g be a minimal degree relation satisfying (3).*

So $V(I_a) = \{\mathbf{u} \in \mathbb{F}_2^k, f(\mathbf{u}) = a\} \subset V(g) = \{\mathbf{v} \in \overline{\mathbb{F}_2}^k, g(\mathbf{v}) = 0\}$ where $\overline{\mathbb{F}_2}$ is the algebraic closure of $\mathbb{F}_2$. According to the Nullstellensatz theorem, exists $m \in \mathbb{N}^*$ so that $g^m \in I_a$. Or g^m is reduced to g by $\{x_1^2 + x_1, \ldots, x_k^2 + x_k\}$, so g belongs to I_a.

Let $(f_1, \ldots, f_m)$ be the reduced Gröbner basis of I_a for the DRL order. First, for all $i \in \{1, \ldots, m\}$, f_i satisfies (3). $g \in I_a$. So $\exists i_0 \in \{1, \ldots, m\}$ s.t. $LM(f_{i_0})$ divides $LM(g)$. As we work with the DRL order, we have degree(f_{i_0})$\leq$degree(g) and as g a minimal degree relation satisfying (3), degree(f_{i_0})=degree(g).

In the same way, we construct $(i_0, \ldots, i_t)$ s.t. $LM(g - f_{i_0} - \cdots - f_{i_{t-1}}) \prec LM(g - f_{i_0} - \cdots - f_{i_t})$ and degree(f_{i_j})=degree(g). As $g \in I$, this sequence is finite. and $\exists h \in \mathbb{N}$ so that $g = f_{i_0} + \cdots + f_{i_h}$. So the Gröbner basis contains a linear basis of all the relations g with minimal degree satisfying (3). As it is a reduced basis, the extract basis is linearly independent.

This proposition proves that the DRL Gröbner basis contains all the minimal degree relations satisfied by the elements $(x_1, \ldots, x_k) \in \mathbb{F}_2^k$ solutions of the equation $f(x_1, \ldots, x_k) = a$. So we can estimate a bound of complexity in some cases.

Remark 1 *Another meaningful interpretation is that proposition 3 gives a "rational" representation of the boolean function f: $f = \frac{N_f}{D_f}$ where N_f, and D_f are polynomials of degree d less than $deg(f)$. For instance the boolean function used in LILI128 [15] is an algebraic function of degree 6 in 10 variables but the Gröbner basis of the ideal of relations $< F - f >$ contains 14 relations of degree 4. For instance:*

$$(f + x5 + x4\,x8 + x4\,x7 + x2\,x6 + x2\,x9 + x6 + x7 + x10)\,x1\,x3 = 0$$

another relation can be written as a fraction:

$$f = \frac{x_2\,x_3\,x_4\,x_5 + x_1\,x_9\,x_3\,x_4 + x_1\,x_2\,x_4\,x_5 + x_1\,x_3\,x_4\,x_5 + x_7\,x_3\,x_2\,x_1 + x_6\,x_3\,x_2\,x_4 + \cdots}{x_2\,x_3\,x_4 + x_1\,x_3\,x_4 + x_2\,x_1\,x_3 + x_1\,x_2\,x_4 + x_1\,x_2 + x_3\,x_2 + x_1\,x_4 + x_1\,x_3}$$

4.3 A bound of Complexity

Let $M(L,d)$ be the number of monomials of degree smaller than d, $M(L,d) = \sum_{i=0}^{d} \binom{L}{i}$. We can easily evaluate the complexity of computing a Gröbner basis of $N \geq M(L,d)$ equations of degree d since the computation is only linear algebra on all the monomials:

Theorem 1 *If $N \geq \frac{M(L,d)}{N_d}$, then the Gröbner basis can be computed in $\mathcal{O}((M(L,d))^{\omega}) = \mathcal{O}(L^{d\omega})$.*

Remark 2 *In the previous theorem we can use for d the value given by the computation of the ideal of relations of the boolean function given by proposition 5. For instance, for LILI128 [15], we can use $d = 4$ instead of 6 and $N_4 = 14$.*

4.4 An Upper bound for d

Theorem 2 *Let f be any boolean function with k inputs (k variables).*

There exits a constant $a \in \mathbb{F}_2$ so that the minimal degree d of the DRL Gröbner basis's elements of $I_a =< f(x_1, \ldots, x_k) + a >$ is lower than $\lfloor \frac{k+1}{2} \rfloor$ where $\lfloor y \rfloor$ is the whole number portion of y.

Proof 3 *To proof this theorem, we have to find a nonzero polynomial $P \in \mathcal{R}$ so that $P \in I_a$. Let us consider $\mathcal{R}$ as a vector space. $(1, x_1, \ldots, x_k, x_1x_2, \ldots, x_1 \cdots x_k)$ is a basis of $\mathcal{R}$. So dim($\mathcal{R}$)=2^k.*

- *Assuming that k is even (i.e. $k = 2l$), we consider the family*

$$\mathcal{F} = \{m, mf \in \mathcal{R} |\ m \text{ monomial and degree}(m) \leq \lfloor \frac{k+1}{2} \rfloor = l\}.$$

$$\#\mathcal{F} = 2\sum_{i=0}^{l} \binom{k}{i} = \sum_{i=0}^{l} \binom{k}{i} + \sum_{i=k}^{l} \binom{l}{i} = 2^k + \binom{k}{l}$$

  So the family $\mathcal{F}$ is linearly dependent. As $\{m|\ m$monomial and degree$(m) \leq \lfloor \frac{k+1}{2} \rfloor\}$ is free, $\exists P, Q \in E$ with $P \neq 0$ and degree(P),degree(Q)$\leq \lfloor \frac{k+1}{2} \rfloor$ so that $Pf + Q = 0$.

- *Assuming that k is odd ($k = 2l+1$), $f(x_1, \ldots, x_k) = f_1(x_1, \ldots, x_{k-1})x_k + f_2(x_1, \ldots, x_{k-1})$. So $(x_k + 1)f(x_1, \ldots, x_k) = (x_k + 1)f_2(x_1, \ldots, x_{k-1})$. As $k - 1$ is even, $\exists P', Q' \in E$ with $P' \neq 0$ and degree(P'),degree(Q')$\leq \lfloor \frac{(k-1)+1}{2} \rfloor = l$ so that $P'f_2 + Q' = 0$. For $P = (x_k + 1)P'$ and $Q = (x_k + 1)Q'$, we have degree(P)$\leq l + 1 = \lfloor \frac{k+1}{2} \rfloor$ because k is odd. In the same way, degree(Q)$\leq \lfloor \frac{k+1}{2} \rfloor$.*

We have shown that $\exists (P_j, Q_j),\ j \in J$ with degree(P_j),degree(Q_j)$\leq \lfloor \frac{k+1}{2} \rfloor$ so that $P_j \neq 0$ and $P_jf + Q_j = 0$. If $\forall j \in J$, $Q_j = 0$, then for $a = 1$, $P_j \in \mathcal{I}_a$ and $P_j \neq 0$. Else $\exists J_1 \subset J$ so that $\forall j \in J_1$, $Q_j \neq 0$ and for $a = O$, $Q_j \in I_a$.

Remark 3 *We can find boolean functions that satisfied this theorem for a but not for $\overline{a} = a+1$.*

Let us consider the boolean function $f(x_1, x_2) = 1 + x_1 + x_2 + x_1x_2$. The DRL Gröbner bases of $< f(x_1, x_2) + a >$ in $\mathcal{R}$ is $< x_1, x_2 >$ for $a = 1$ and $< f(x_1, x_2) >$ for $a = 0$.

Corollary 1 *If we have $M(L, \lfloor \frac{k+1}{2} \rfloor)$ output elements, then the complexity of the Gröbner basis computation is $\mathcal{O}((\sum_{i=0}^{\lfloor \frac{k+1}{2} \rfloor} \binom{i}{L})^\omega) = \mathcal{O}(L^{\lfloor \frac{k+1}{2} \rfloor \omega})$.*

Proof 4 *From the previous proof, there is at least one equation with degree $\lfloor \frac{k+1}{2} \rfloor$. Then the proof is similar to the proof of theorem 1.*

Remark 4 *This result is not optimal. We do not use the fact that f is balanced in the demonstration, and so we can expect better result in practice (see Annex A). For instance, the boolean function $(x_1, \ldots, x_1) \mapsto x_2\, x_3\, x_4\, x_5 + x_2\, x_3 + x_1$ verifies $d = 2 < 3$ and $N_d = 2$.*

Since LILI128 [15] is a boolean function in 10 variables $k = 10$ we expect to find degree 5 relations. In fact we found 14 relations of degree 4.

5 Experimental Results

For the computer simulations , we use the algorithm F_5([12]) modified to include the Frobenius map $h^2 = h$. The algorithm is implemented in C in the program FGb (there are also specific linear algebra implementation for the field $\mathbb{F}_2$). Furthermore, FGb integrate the generation of the algebraic system $\mathcal{S}(f, N)$. Without care, this part could be the most consuming part of the computation (it was the case in our first implementation in the general computer algebra system Maple).

We describe a simple method to generate efficiently the equations:

Proposition 6 *Let $g(x_0, \ldots, x_{L-1}) = g_1(x_0, \ldots, x_{L-2}) + x_{L-1}g_2(x_0, \ldots, x_{L-2})$. be the m^{th} equation generated.*

Next the $(m+1)^{th}$ equation is given by :

$$z_m - z_{m-1} + g_1(x_1, \ldots, x_{L-1})) + \sum_{j=0}^{L-1} \delta_{c_{L-j}} x_j g_2(x_1, \ldots, x_{L-1}))$$

with $c_0 - c_1X - c_2X^2 - \cdots - c_{L-1}x^{L-1} - c_LX^L = P_F(X)$ the feedback polynomial and δ_b the Kronecker's symbol.

A complexity bound of the generation of a equations is $(w(P_F)+1)\binom{L-1}{L+deg(f)-2} + \binom{deg(f)}{L-1}$ with d the degree of f (and g).

We remark that the complexity of the generation of the equations depends on the weight of P_F (the number of nonzero coefficient in the feedback polynomial P_F), whereas, the

computation of Gröbner basis depends *only* on the degree of the boolean function f (see section 4.3).

We present some simulation results for several LFSR of various size $128 \geq L \geq 40$. The running-times are given for a HP workstation with an alpha EV68 processor at 1000 Mhz. All considered filtered functions are balanced and can be found in the literature: all the boolean function examples are taken from [5, 7, 6] for the CanFil examples and from [17] for the LBGZ examples. We can find their algebraic normal form and their minimum degree d in Annex A.

Example	L	N	$w(P_F)$	Complexity	Solve
LBGZ 1	40	1071	17	$L^{3\omega}$	18 s
LBGZ 1	80	8541	47	$L^{3\omega}$	1 h 32 m
LBGZ 1	89	11758	53	$L^{3\omega}$	4 h 28 m
LBGZ 2	40	3568	17	$L^{3\omega}$	19.3 s
LBGZ 2	80	28468	39	$L^{3\omega}$	1 h 44 s
LBGZ 2	89	39190	53	$L^{3\omega}$	4 h 32 m
CanFil 1	80	6342	39	$L^{2\omega}$	1.1 s
CanFil 1	128	12384	63	$L^{2\omega}$	10.2 s
CanFil 2	80	6342	39	$L^{2\omega}$	1.1 s
CanFil 2	128	12384	63	$L^{2\omega}$	10.3 s
CanFil 3	80	3774	39	$L^{2\omega}$	1.3 s
CanFil 3	128	12384	63	$L^{2\omega}$	12.4 s
CanFil 4	80	6342	39	$L^{2\omega}$	1 s
CanFil 4	128	12384	63	$L^{2\omega}$	9.5 s
Example	L	N	w(P)	Complexity	Solve
CanFil 5	80	1622	35	$L^{2\omega}$	0.1 s
CanFil 5	128	4129	63	$L^{2\omega}$	9.1 s
CanFil 6	80	1621	39	$L^{2\omega}$	0.8 s
CanFil 6	128	4129	63	$L^{2\omega}$	8.9 s
CanFil 7	80	1081	39	$L^{2\omega}$	0.97 s
CanFil 7	128	2753	63	$L^{2\omega}$	10 s
CanFil 8	40	1071	17	$L^{3\omega}$	18.1 s
CanFil 8	80	17081	51	$L^{3\omega}$	1 h 45 m
CanFil 8	89	23515	49	$L^{3\omega}$	4 h 35 m
CanFil 9	40	10702	17	$L^{3\omega}$	21.2 s
CanFil 9	70	57227	37	$L^{3\omega}$	33 m 21 s
CanFil 10	40	1071	17	$L^{3\omega}$	16.5 s
CanFil 10	89	11757	53	$L^{3\omega}$	4 h 21 m

Table 1: **Nonlinear filter generator on DS25 1000 Mhz**

All results presented in the above table confirm the validity of the previous complexity estimation of theorem 1.

6 Attack when the number of bits of the running-key is very small

What is the minimum number of bits of the running-key required to find a unique solution to the system $\mathcal{S}(f, N)$? A very coarse upper bound is:

Proposition 7 *Let be a nonlinear filter generator with a maximum-length LFSR (length =L) and a nonzero balanced boolean function f.*

Any output sequence $(z_i)_{i=0}^{2^L-2}$ has at most one *initial state which gives this sequence as the output of the nonlinear filter generator.*

Thus, the system has only a solution if we have $N \geq 2^L - 1$ elements. This proposition does not tell us the minimial number N. Again, we are doing computer simulations.

The figure 1 represents, for a ratio of the number of given outputs N by the length of the secret key L, the frequency of examples where we find the solution.

According to this graph, it seems that we only need $N = L + \epsilon$ to find the initial state of the nonlinear filter generator. So we need a number of outputs proportional to the length of the initial state to find a unique solution. Note that even if N is smaller than L we can compute the Gröbner basis of $\mathcal{S}(f, N)$ and found all the possible solutions (this is similar to the list decoding concept in Error Correcting Codes). However, when N is small the computation of the Gröbner is much more difficult and it is not even clear if it can be done in polynomial time.

Very surprisingly, we found that some examples in the previous table 1 can be attacked very efficiently even when N is small. See table 2 below. This is an open issue to understand and to be able to predict such a behavior from the boolean function and the tapping sequence.

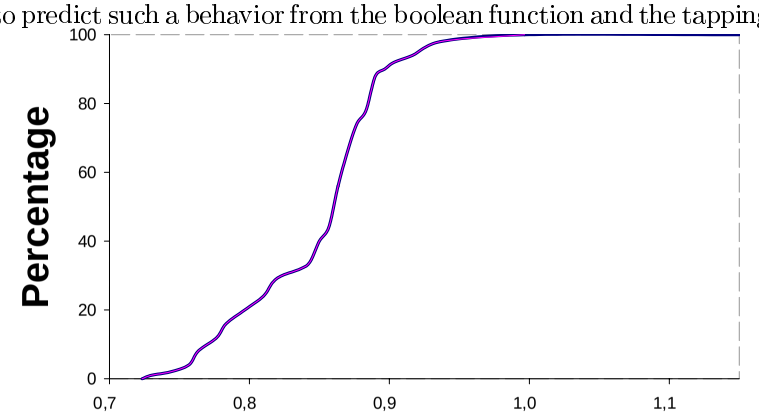

Figure 1. Experimental analysis of the minimal number of output bits

Example	length L of LFSR	number N of outputs	Gröbner
CanFil 5	40	44	22.3 s
CanFil 5	50	54	46.0 s
CanFil 5	60	66	89.0 s
CanFil 5	70	77	221.0 s

Table 2. Gröbner attack when $N \approx L$ on HP DS25 1000 Mhz

7 Fast Correlation Attacks Comparison

The essential principle of attacks by correlation is to compare the research of the initial state of a register to a problem of correction of errors on a code $\mathcal{C}$.

These attacks imply change in the design criteria of nonlinear filter register. To prevent from fast correlation, the boolean function f must be *m-resilient*,i.e. if for m variables fixed in $\mathbb{F}_2$, the $k-m$ variables boolean function $\tilde{f}$ deduced from f is still balanced.

These particular boolean functions have an important propriety demonstrated by Siegenthaler [18]: a m-resilient function f over $\mathbb{F}_2^k$ satisfies the relation $deg(f) \leq k-m-1$.

So the choice of a m-resilient function decreases the degree of f. Then, it is easier to solve the system with an algebraic attack which depends on the degree. In other way, the functions f which are strongly secure for algebraic attack have a high degree and then they are correlate. So it is easy to solve them with correlation attacks. We can say that the two attacks are complementary.

In [6], the boolean function is approximated by a linear function $x \mapsto \alpha.x$, $\alpha \in \mathbb{F}_2^L$. This approximation give a linear relation for an output. Fast correlation attacks derive from $\mathcal{C}$ a new code $\mathcal{C}'$ having a lower dimension $\nu < L$, for which a fast decoding method is feasible. Then they find the ν first elements of the initial state.Such a code $\mathcal{C}'$ is obtained by computing all linear combination of δ relations which vanish on the last $(L-\nu)$ positions. Parameter δ does usually not exceed 4 or 5.

The fast correlation algorithm have two parts:

- A precomputation part, in which we find all the δ relations which vanish on the last $(L-\nu)$ positions if they are combined.
- A decoding part, in which we find the ν first elements of the initial state.

The minimum number of outputs required by the attack is given by:

$$(\delta!\nu)^{\frac{1}{\delta}} 2^{\frac{L-\nu}{\delta}} < N_{min} \leq (2ln(2)\delta!\nu)^{\frac{1}{\delta}} 2^{\frac{L-\nu}{\delta}}$$

So fast correlation attacks find ν elements of the initial state where Gröbner basis computation find the whole state. Furthermore, correlation attack have a probabilistic rate of success whereas Gröbner basis computation always find the result.

The most important point is that number of equation needed by correlation attacks is *exponential* whereas Gröbner basis attacks need only polynomial number of outputs. On the

other hand a drawback of the Gröbner attack is that this number depends strongly on the choice of the boolean functions f. So in general, Gröbner basis computation need a lower number of outputs as is made clear by the table 3.

We present the number of outputs needed for LFSR of length 40 with $\delta = 2$ and $\nu = 20$ according to examples proposed in [6]. We can remark that for the example 9, Gröbner basis method needs more outputs than Correlation attacks. It comes from the fact that for a small L, the number L^d is higher than 2^L. But for real size L, Gröbner basis method would need less outputs then Correlation attacks.

Example	1	2	3	4	5	6	7	8	9	10
Gröbner basis	821	821	821	412	412 or 44	412	274	1071	10701	1071
Correlation	7625	7625	7625	7625	7625	7625	7625	7625	7625	7625

Table 3. Number of required output bits

8 Gröbner attack with Precomputation Phase

For now, we have computed a DRL Gröbner basis for an explicit output sequence $(z_i)_{i=0}^{N-1}$. Thus, for another explicit output sequence, we need to compute *again* the Gröbner basis whereas only the constants in equations $\mathcal{S}(f, N)$ have been modified.

Another solution is to add new variables $(z_i)_{i=0}^{N-1}$ and to compute a "symbolic Gröbner basis". Then for all explicit sequences we need only to substitute the values of z_i in the Gröbner basis.

More precisely, we have to compute a Gröbner basis for the elimination ordering $\succ$ introduced in section 3.2. The polynomial ring is now $\mathcal{R}'_N = \mathbb{F}_2[x_0, \ldots, x_{L-1}, z_0, \ldots, z_{N-1}]/ < x_0^2 + x_0, \ldots, x_{L-1}^2 + x_{L-1}, z_0^2 + z_0, \ldots, z_{N-1}^2 + z_{N-1} >$

From proposition 1, we know that if we substitute the variables $(z_i)_{i=0}^{N-1}$ with explicit values, we still have a Gröbner basis of the ideal I_N (but not necessarily a minimal Gröbner basis).

So to evaluate the complexity we have to bound the complexity of the precomputation phase and the complexity of the substitution stage.

Theorem 3 *If we take N equations with N high enough to have only one solution for all possible outputs sequences, we find, in the computed Gröbner basis, equations with a degree one for the variables x_j, $0 \leq j \leq L-1$. These linear equations can be written*

$$CX = B$$

where C and B are matrices with polynomial coefficients. C is of rank L for a given output $(z_i)_{i=0}^{N-1}$.

Proof 5 *After the substitution in the Gröbner basis, we still have a Gröbner basis. As we have only one solution for the system, the new basis contains equations of degree 1 for all*

the x_j. Hence the Gröbner basis contains linear equations, $CX = B$ where C and B are polynomial matrices in the variables $(z_i)_{i=0}^{N-1}$.

Moreover, if for all outputs, we only have only one solution, the matrix C have rank L for a given output $(z_i)_{i=0}^{N-1}$.

Let d' be the minimal degree of the relations $PF+Q$ in the ideal $< F - f(x_0, \ldots, x_{L-1}) >$ (there relations appear during the computation of the Gröbner basis for an elimination ordering).

Theorem 4 *If $N \geq \frac{M(L,d')}{N_{d'}}$, the precomputation phase can be done in*

$$(N_{d'} + L + 1)N_{d'}^2 N^3$$

operations. The cost of substitution is

$$L(L+1)N$$

where $M(L, d')$ is the number of monomials in $\mathbb{F}_2[x_0, \ldots, x_{L-1}]$ of degree less than d'.

Proof 6 *Similarly to section 4.3, the Gröbner basis computation reduces to linear algebra on a matrix with $NN_{d'}$ lines and $M(L, d') + (L+1)N$ columns. So a complexity bound is $(NN_{d'})^2(NN_{d'} + (L+1)N) = (N_{d'} + L + 1)N_{d'}^2 N^3$.*

Furthermore, we obtain a polynomial matrix C with only linear equations as for B. With such relations, the resolution is only the substitution of $(z_i)_{i=0}^{N-1}$ in C and B which have complexity at most $L(L+1)N$ and then an inversion of a $L \times L$ matrix in $\mathbb{F}_2$.

9 Conclusion

We have a presented a cryptanalysis of non linear filter generators based on fast algorithms for computing Gröbner bases. We proof theoretically and experimentally that non linear boolean functions of degree D behave, from an algebraic of point, as a fonction of lower degree $d < D$ (and in practice $d \leq 4$ for all the functions we have tested). Then we have established that with only L^d bits of the running key we can recover efficiently the secret key. For some examples, we were able to recover the secret key with only $L + \epsilon$ bits of the running-key. However, it is an open issue to predict the complexity of the Gröbner computation when the number of output bits is $L + \epsilon$ and it is not clear if it can be done in polynomial time.

Acknowledgements

We gratefully acknowledge several useful discussions with A. Canteaut. We would like to thank A. Canteaut, S. Leveiller and C. Fontaine to have provide us a list of boolean functions. We are indebted to the LIP6 for its partial support of this work (Alpha DS25) and to the DGA (Celar).

References

[1] P. van Oorschot A. Menezes and S. Vantome. *Handbook of Applied Cryptography*. http://www.cacr.math.uwaterloo.ca/hac/, 1996.

[2] G. Ars. Une application des bases de gröbner en cryptographie. *DEA de Rennes I*, 2001.

[3] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. Complexity of gröbner bases computation of generic systems. in preparation, 2003.

[4] T. Becker and V. Weispfenning. *Gröbner Bases : A Computational Approch to Commutative Algebra*. Graduate Texts in Mathematics. Springer-Verlag, 1993.

[5] A. Canteaut and E Filiol. Ciphertext only reconstruction of stream ciphers based on combination generators. *Fast Software Encryption, LNCS 1978*, pages 165–180, 2000.

[6] A. Canteaut and E Filiol. On the influence of the filtering function on the performance of fast correlation attacks on filter generators. Personal Communication, 2002.

[7] A. Canteaut and M Trabbia. Improved fast correlation attacks using parity-check equations of weight 4 and 5. *Eurocrypt, LNCS 1807*, pages 573–588, 2000.

[8] N. Courtois. Higher order correlation attacks, xl algorithm, and cryptanalysis of toyocrypt. *ICISC, LNCS 2587*, 2002.

[9] Nicolas Courtois, Adi Shamir, Jacques Patarin, and A. Klimov. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Eurocrypt'2000*, volume 1807 of *Lecutures Notes in Computer Science*, pages 392–407. Springer Verlag, 2000.

[10] D. O'Shea D. Cox, J. Little. *Ideals, Varieties, and Algorithms : An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer-Verlag, 1996.

[11] J.C. Faugère. A new efficient algorithm for computing gröbner bases f4. *Journal of Pure and Applied Algebra*, 139:61–88, 1999.

[12] J.C. Faugère. A new efficient algorithm for computing gröbner bases without reduction to zero f5. *ISSAC*, 2002. ACM Press.

[13] C. Fontaine. *Contribution à la recherche de fonctions booléennes hautement non linéaires, et au marquage d'images en vue de la protection des droits d'auteur*. PhD thesis, Université de Paris VI, 1998.

[14] J. Dj. Golić. On the security of nonlinear filter generators. *Fast Software Encryption*, pages 173–188, 1996.

[15] J. Dj. Golič L. Simpson, E. Dawson and W. Millan. Lili keystream generator. *Selected Areas in Cryptography*, pages 248–261, 2000.

[16] W. Meier and O. Staffelbach. Fast correlation attacks on certain stream ciphers. *Journal of Cryptology*, pages 159–176, 1989.

[17] P. Guillot S. Leveiller, J. Boutros and G. Zémor. Cryptanalysis of nonlinear filter generators with {0,1}-metric viterbi decoding. *ICIS 2001, LNCS 2288*, 2001.

[18] T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, pages 776–780, 1984.

A Annex : list of boolean functions

- LBGZ 1 [17], the boolean function f is $1 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_1x_7 + x_2x_3 + x_7x_2 + x_1x_2x_3 + x_1x_2x_6 + y_1x_2x_7$
- LBGZ 2, the boolean function f is $x_1 + x_4 + x_5 + x_6 + x_7 + x_1x_2 + x_1x_7 + x_6x_2 + x_6x_3 + x_3x_8 + x_1x_2x_4 + x_1x_2x_6 + x_1x_2x_7 + x_8x_1x_2 + x_1x_2x_3 + x_6x_1x_3 + x_1x_2x_3x_4 + x_1x_2x_3x_5 + x_8x_3x_2x_1$
- Examples from articles [5, 7, 6].
  - CanFil 1, the boolean function f is $x_1\,x_2\,x_3 + x_1\,x_4 + x_2\,x_5 + x_3$
  - CanFil 2, the boolean function f is $x_1x_2x_3 + x_1x_2x_4 + x_1x_2x_5 + x_1x_4 + x_2x_5 + x_3 + x_4 + x_5$
  - CanFil 3, the boolean function f is $x_2x_3x_4x_5 + x_1x_2x_3 + x_2x_4 + x_3x_5 + x_4 + x_5$
  - CanFil 4, the boolean function f is $x_1x_2x_3 + x_1x_4x_5 + x_2x_3 + x_1$
  - CanFil 5, the boolean function f is $x_2\,x_3\,x_4\,x_5 + x_2\,x_3 + x_1$
  - CanFil 6, the boolean function f is $x_1x_2x_3x_5 + x_2x_3 + x_4$
  - CanFil 7, the boolean function f is $x_1x_2x_3 + x_2x_3x_4 + x_2x_3x_5 + x_1 + x_2 + x_3$
  - CanFil 8, the boolean function f is $x_1\,x_2\,x_3 + x_2\,x_3\,x_6 + x_1\,x_2 + x_3\,x_4 + x_5\,x_6 + x_4 + x_5$
  - CanFil 9, the boolean function f (2-resilient function with maximal nonlinearity) is $x_1 + x_2 + x_5 + x_6 + x_2x_4x_5x_7 + x_2x_5x_6x_7 + x_3x_4x_6x_7 + x_1x_2x_4x_7 + x_1x_3x_4x_7 + x_1x_3x_6x_7 + x_1x_4x_5x_7 + x_1x_2x_5x_7 + x_1x_2x_6x_7 + x_1x_4x_6x_7 + x_3x_4x_5x_7 + x_2x_4x_6x_7 + x_3x_5x_6x_7 + x_6x_7 + x_4x_6 + x_4x_7 + x_5x_7 + x_2x_5 + x_3x_4 + x_3x_5 + x_3x_4x_5 + x_3x_4x_7 + x_3x_6x_7 + x_5x_6x_7 + x_2x_6x_7 + x_1x_4x_6 + x_1x_5x_7 + x_2x_4x_5 + x_1x_4 + x_2x_7 + x_2x_3x_7 + x_1x_3x_5x_7 + x_1x_2x_7 + x_1x_4x_5 + x_1x_2x_3x_7$
- CanFil 10, the boolean function f is $x_1x_2x_3 + x_2x_3x_4 + x_2x_3x_5 + x_1 + x_2 + x_3 + x_6x_7$

- Fontaine 1 [13], the boolean function f is $x_1+x_2+x_3+x_4+x_5+x_6+x_7+x_3x_4x_5x_6+x_2x_3x_4x_5+x_2x_3x_4x_7+x_2x_3x_5x_6+x_2x_3x_6x_7+x_2x_4x_5x_6+x_2x_4x_5x_7+x_2x_5x_6x_7+x_3x_4x_6x_7+x_1x_2x_5x_7+x_1x_3x_4x_5+x_1x_3x_4x_6+x_1x_3x_4x_7+x_1x_3x_5x_6+x_1x_3x_6x_7+x_1x_4x_5x_7+x_1x_5x_6x_7+x_1x_2x_4x_6+x_1x_2x_4x_7+x_1x_2x_5x_6+x_1x_2x_6x_7+x_1x_4x_6x_7+x_3x_4x_5x_7+x_2x_4x_6x_7+x_3x_5x_6x_7+x_6x_7+x_5x_6+x_2x_3x_4x_5x_6+x_4x_5+x_4x_7+x_5x_7+x_2x_5+x_3x_4+x_3x_5+x_4x_5x_6+x_3x_4x_6+x_3x_5x_6+x_3x_5x_7+x_3x_6x_7+x_5x_6x_7+x_2x_4x_7+x_4x_6x_7+x_1x_5x_6+x_1x_4x_6+x_1x_5x_7+x_1x_2x_5+x_1x_2x_6+x_1x_3x_4+x_1x_3x_5+x_2x_3x_6+x_2x_4x_5+x_1x_2+x_1x_3+x_1x_4+x_1x_7+x_3x_6+x_2x_3+x_2x_4+x_3x_7+x_1x_3x_4x_5x_6+x_1x_2x_3x_4x_5+x_1x_2x_3x_4x_6+x_1x_2x_3x_4x_7+x_1x_2x_3x_5x_6+x_1x_2x_3x_5x_7+x_1x_2x_3x_6x_7+x_1x_2x_4x_5x_6+x_1x_2x_4x_5x_7+x_1x_2x_5x_6x_7+x_1x_3x_4x_6x_7+x_1x_4x_5x_6x_7+x_1x_2x_3x_4x_5x_6+x_2x_3x_4x_5x_7+x_2x_3x_4x_6x_7+x_8+x_9$
- LILI 128 [15] The boolean function f is $x_1\,x_4\,x_5+x_1\,x_7+x_2\,x_8+x_3\,x_9+x_3\,x_{10}+x_{10}\,x_2+x_7+x_8+x_9+x_6+x_2\,x_3\,x_5+x_1\,x_4\,x_7+x_2\,x_4\,x_7+x_1\,x_2\,x_8+x_1\,x_3\,x_8+x_2\,x_3\,x_8+x_2\,x_4\,x_8+x_9\,x_1\,x_2+x_7\,x_3\,x_1+x_2\,x_4\,x_5+x_6\,x_2\,x_1+x_6\,x_4\,x_1+x_7\,x_1\,x_2+x_1\,x_2\,x_3\,x_7+x_1\,x_5+x_1\,x_2\,x_3\,x_8+x_1\,x_2\,x_3\,x_{10}+x_7\,x_5\,x_4\,x_2+x_9\,x_1\,x_2\,x_4+x_4\,x_9\,x_3\,x_2+x_5\,x_1\,x_2\,x_4+x_2\,x_3\,x_5\,x_1+x_6\,x_1\,x_2\,x_4\,x_5+x_5\,x_1\,x_7\,x_4\,x_2+x_7\,x_4\,x_3\,x_2+x_5\,x_4+x_7\,x_3\,x_2\,x_4\,x_1+x_8\,x_1\,x_4\,x_3+x_6\,x_2\,x_3\,x_5\,x_4\,x_1+x_7\,x_3\,x_2\,x_5\,x_4\,x_1+x_6\,x_1\,x_4\,x_3+x_6\,x_4\,x_5\,x_3\,x_2+x_6\,x_4\,x_5\,x_2+x_7\,x_5\,x_4\,x_3\,x_2+x_8\,x_3\,x_2\,x_4\,x_1+x_7\,x_4\,x_1\,x_2$
  with degree 6 and $\gamma=[1,2,4,8,13,21,31,45,66,81]$

Example				For a=0 or a=1	
Name	$deg(f)$	k	$\lfloor\frac{k+1}{2}\rfloor$	d	N_d
LBGZ 1	3	7	4	3	10
LBGZ 2	4	8	4	3	3
CanFil 1	3	5	3	2	1
CanFil 2	3	5	3	2	1
CanFil 3	4	5	3	2	1
CanFil 4	3	5	3	2	2
CanFil 5	4	5	3	2	2
CanFil 6	4	5	3	2	2
CanFil 7	3	5	3	2	3
CanFil 8	3	6	3	3	10
CanFil 9	4	7	4	3	1
CanFil 10	3	7	3	3	10
Fontaine 1	6	9	5	4	35
LILI 128	6	10	5	4	14

With :

- k the number of variable of f
- d the degree of the low-degree relations in $I_a =< f(x_1,\ldots,x_k)+a >$

- N_d the number of linearly independent low-degree relations.

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Futurs : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

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