Algebraic and Correlation Attacks against Linearly Filtered Non Linear Feedback Shift Registers

Côme Berbain[1], Henri Gilbert[1], and Antoine Joux[2]

[1] Orange Labs
38-40 rue du Général Leclerc, 92794 Issy-les-Moulineaux, France
[2] DGA and Université de Versailles
45 avenue des Etats-Unis, 78035 Versailles Cedex, France
come.berbain@orange-ftgroup.com
henri.gilbert@orange-ftgroup.com
antoine.joux@prism.uvsq.fr

Abstract. The filter generator is a well known and extensively studied stream cipher construction. It consists of a Linear Feedback Shift Register (LFSR) filtered by a non linear Boolean function. In this paper we focus on the dual construction, namely a linearly filtered Non linear Feedback Shift Register (NFSR). We show that the existing algebraic and correlation attacks against the filter generator can be transposed to mount algebraic or correlation attacks against this dual construction. We investigate such attacks and extend them to the case where a linearly filtered NFSR is combined linearly with one or more non linearly filtered LFSRs. We apply our algebraic attack to a modified version of Grain-128, resulting in an attack requiring 2^{105} computations and 2^{39} keystream bits. Even though this attack does not apply to the original Grain-128, it shows that the use of a NFSR is not sufficient to avoid all algebraic attacks.

1 Introduction

Stream ciphers represent, together with block ciphers, one of the two main classes of symmetric encryption algorithms. They produce a pseudo-random keystream sequence of numbers over a small alphabet, typically the binary alphabet $\{0, 1\}$. To encrypt a plaintext sequence, each plaintext symbol is combined with the corresponding symbol of the keystream sequence using a group operation, usually the exclusive or operation over $\{0, 1\}$.

A classic way to build a random number generator is to use a Linear Feedback Shift Register (LFSR) and to apply a non-linear Boolean function f to the current LFSR state to produce the keystream. This construction is known as the filter generator. It has been extensively studied over the past years resulting in a large number of criteria for the design of such ciphers. For example, the correlation and fast correlation attacks [23, 24, 9, 19] against this scheme can be avoided if the function f has a high order of correlation immunity or satisfies certain criteria [15]. Algebraic attacks [12, 2, 6, 11] led to the notion of algebraic immunity of Boolean function.

Recently new stream ciphers using Non linear Feedback Shift Registers (NFSRs) were proposed as an alternative to LFSR-based stream ciphers. Finalist candidates to

Appeared in : SAC 2008, LCNS , pp. 2–,

the eSTREAM project like Trivium [8] or Grain [17] are using one or several NFSRs combined or not with LFSRs.

In this paper, we analyze the dual of the classical filter generator construction, i.e. a Non linear Feedback Shift Register with a linear output function. We show that it is easy to formally express any internal state variable as a linear combination of the initial state variables and of keystream bits, and this (more surprisingly) allows mounting algebraic or correlation attacks against such a scheme. We extend our analysis to the case where a linearly filtered NFSR is linearly combined with one or several non linearly filtered LFSRs. This allows us to mount an attack against a modified version of Grain-128. Even though this attack does not apply to the original Grain-128, it shows that the use of a NFSR is not sufficient to avoid all algebraic attacks. In particular it contradicts the common idea that the increase in the degree due to the NFSR allows to avoid algebraic attacks.

The paper is organized as follows: in Section 2, we introduce linearly filtered NFSRs and we explain why they might seem to naturally resist algebraic attacks. In Section 3, we introduce a simple formal technique applicable to any linearly filtered NFSR. In Section 4 and 5, we show how to mount algebraic and correlation attacks against these schemes; in Section 6, we extend our attacks to linear combinations of a linearly filtered NFSR and non-linearly filtered LFSRs and we present our attack against a modified version of Grain-128.

2 Linearly Filtered NFSRs

The filter generator, i.e. a LFSR filtered by a nonlinear Boolean function, has been widely studied, and some ciphers are based on this construction like WG [14] or Sfinks[7], two of the candidates to the eSTREAM competition.

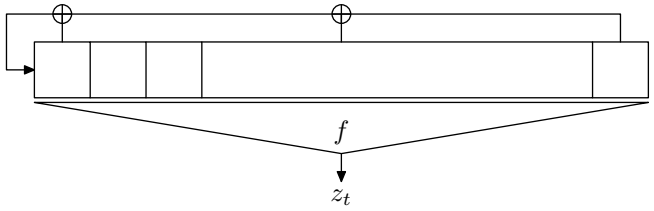

Fig. 1. Filter Generator

A large number of attack techniques applicable to filter generators have been proposed like correlation and fast correlation attacks [13, 18–20] or algebraic and fast algebraic attacks [12, 11, 1, 2].

For now on, we consider the dual construction, i.e. we swap the update and output functions, i.e. the linear and the non linear functions. The resulting system is a linearly filtered Non Linear Feedback Shift Register (NFSR).

Appeared in : SAC 2008, LCNS , pp. 2–,

More formally we consider an n-bit NFSR and denote its initial state by $(x_0, \ldots, x_{n-1})$. This NFSR is updated using a Boolean function g:

$$x_{t+n} = g(x_t, \ldots, x_{t+n-1})$$

At each iteration, the function g is applied to the current internal state of the NFSR and a new value x_{t+n} is produced. We denote by d_g the total degree of g, i.e. the number of variables in the ANF (algebraic normal form) representation of g.

The considered output function is linear, i.e. each keystream bit z_t is a linear combination of the internal state of the NFSR at time t.

$$z_t = \bigoplus_{k=0}^{n-1} \alpha_k x_{t+k}$$

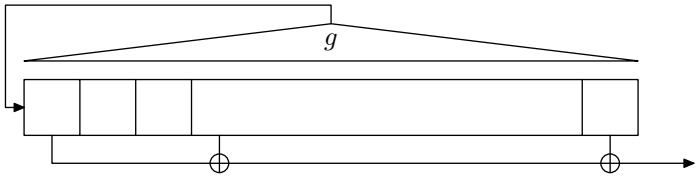

Fig. 2. Linearly filtered NFSR

All the results of this paper can be applied to any linearly filtered NFSR. However in order to render our presentation easier to follow, we will illustrate our results with a first simple example inspired from Grain (a second example inspired from Grain-128 will be introduced in Section 6). In our first example, we took the NFSR from version 1 of Grain stream cipher [17] and slightly modified the output function. In the original Grain, the output of the cipher is composed of a linear combination of the NFSR internal state and of a Boolean function of the LFSR internal state and of a single bit of the NFSR internal state. We removed the LFSR and the associated Boolean function and kept the linear filter of the NFSR.

This NFSR is 80-bit long and it is governed by the recurrence:

$$x_{t+80} = g(x_t, x_{t+1}, \ldots, x_{t+79}),$$

where the expression of nonlinear feedback function g is given by

$$\begin{aligned} g(x_t, x_{t+1}, \ldots, x_{t+79}) = {} & x_{t+62} \oplus x_{t+60} \oplus x_{t+52} \oplus x_{t+45} \oplus x_{t+37} \oplus x_{t+33} \oplus x_{t+28} \\ & \oplus x_{t+21} \oplus x_{t+14} \oplus x_{t+9} \oplus x_t \oplus x_{t+63}x_{t+60} \oplus x_{t+37}x_{t+33} \\ & \oplus x_{t+15}x_{t+9} \oplus x_{t+60}x_{t+52}x_{t+45} \oplus x_{t+33}x_{t+28}x_{t+21} \\ & \oplus x_{t+63}x_{t+45}x_{t+28}x_{t+9} \oplus x_{t+60}x_{t+52}x_{t+37}x_{t+33} \\ & \oplus x_{t+63}x_{t+60}x_{t+21}x_{t+15} \oplus x_{t+63}x_{t+60}x_{t+52}x_{t+45}x_{t+37} \\ & \oplus x_{t+33}x_{t+28}x_{t+21}x_{t+15}x_{t+9} \\ & \oplus x_{t+52}x_{t+45}x_{t+37}x_{t+33}x_{t+28}x_{t+21}. \end{aligned}$$

Appeared in : SAC 2008, LCNS , pp. 2–,

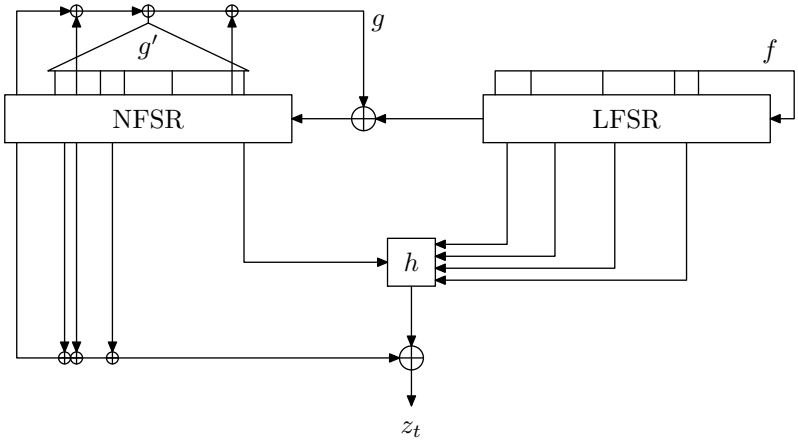

Fig. 3. Grainv1

The cipher output bit z_t is given by the following linear function of the current NFSR internal state:

$$z_t = x_{t+1} \oplus x_{t+2} \oplus x_{t+4} \oplus x_{t+10} \oplus x_{t+31} \oplus x_{t+43} \oplus x_{t+56} \oplus x_{t+63}$$

One of the motivations for NFSR based stream ciphers is that they are generally believed to be naturally immune against algebraic attacks. In fact due to the structure of the NFSR, each internal state variable can be written as a function of degree d_g of the n previous internal state variables. Consequently the degree of the algebraic expression of any internal state variable in the initial variables $(x_0, \ldots, x_{n-1})$ is growing. In the case of a linearly filtered NFSR, the algebraic expression of any keystream bit is a linear combination of the internal state variable. Consequently, the degree of the algebraic expression of each keystream bit is also growing.

In our example, the first 17 keystream bits can be written as linear combinations of the initial state variables. The next 17 keystream bits can be written as polynomials of degree 6. The next 17 polynomials are of degree 10, and the degree keeps growing until it reaches the number of variables. The size of the "blocks" of equations of constant degree is determined by the difference between the position of the feedback (80 in our example) and the position of the tap of highest index in the expression of the update function (63 in our example).

Algebraic attacks as independently discovered by Courtois and Meier [12] and by Ars and Faugère [3] try to reduce the degree of the polynomials corresponding to keystream bits by the use of annihilators, and then linearize the obtained system in order to solve it. These attacks require that a large quantity of keystream bits be available and that each keystream bit can be expressed as a polynomial of fixed degree. It is commonly believed that since the degree is growing with the number of keystream bits, this makes algebraic attacks based on such equations inefficient against these systems.

Appeared in : SAC 2008, LCNS , pp. 2–,

3 A Preliminary Observation

We now introduce a very simple technique that allows us to formally express any internal state variable of the NFSR as a linear combination of the initial state variables and of keystream bits. We consider the sequence of internal state variables $(x_i)_{i\geq 0}$. The initial state of the NFSR is composed of the n first variables $(x_0, \ldots, x_{n-1})$. We can relate the variables x_i thanks to the non linear update function of the NFSR g. This leads to increasing degrees as stated earlier. We propose instead to use the linear filtering function to derive linear relations between these variables. We recall the expression of z_t for any t:

$$z_t = \bigoplus_{k=0}^{n-1} \alpha_k x_{t+k}$$

We prove the correctness of this technique by induction. We denote by i the highest value $0 \leq k \leq n-1$ such that α_k is not equal to zero. Let us consider the first bit of keystream which is dependent on x_n. We can write

$$z_{n-i} = x_n \oplus \bigoplus_{k=0}^{i-1} \alpha_k x_{k+n-i}$$

By exchanging the terms z_{n-i} and x_n, we can express x_n as a linear combination of a keystream bit and of a subset of the initial state variables $(x_0, \ldots, x_{n-1})$.

Let us now assume that for all $j \leq t$, all bits x_j can be expressed as a linear combination of the initial state variables and of keystream bits. Let us consider the keystream bit z_{t+1-i}. It results from the definition of i that this is the first keystream bit which depends of x_{t+1}. We can write

$$z_{t+1-i} = x_{t+1} \bigoplus_{k=0}^{i-1} \alpha_k x_{k+t+1-i}$$

By exchanging the terms z_{t+1-i} and x_{t+1}, we can express x_{t+1} as a linear combination of a keystream bit and of variables x_j with $j < t+1$. By applying the induction assumption, we can replace all these variables x_j by their respective linear combination and we finally express x_{t+1} as a linear combination of keystream bits and of the initial state variables $(x_0, \ldots, x_{n-1})$.

The complexity of building such linear expressions only depends on the number of variables we want to express since the computation can be done in a efficient way by following the induction process we just described. If we want to be able to express N variables, our technique requires $n \cdot N$ computations and $(n+1) \cdot N$ bits of memory.

In our example, the former technique gives:

$$\begin{aligned}
x_{80} &= z_{17} \oplus x_{76} \oplus x_{60} \oplus x_{48} \oplus x_{27} \oplus x_{21} \oplus x_{19} \oplus x_{18} \\
x_{81} &= z_{18} \oplus x_{77} \oplus x_{61} \oplus x_{49} \oplus x_{28} \oplus x_{22} \oplus x_{20} \oplus x_{19} \\
x_{82} &= z_{19} \oplus x_{78} \oplus x_{62} \oplus x_{50} \oplus x_{29} \oplus x_{23} \oplus x_{21} \oplus x_{20} \\
x_{83} &= z_{20} \oplus x_{79} \oplus x_{63} \oplus x_{51} \oplus x_{30} \oplus x_{24} \oplus x_{22} \oplus x_{21} \\
x_{84} &= z_{21} \oplus x_{80} \oplus x_{64} \oplus x_{52} \oplus x_{31} \oplus x_{25} \oplus x_{23} \oplus x_{22}
\end{aligned}$$

Appeared in : SAC 2008, LCNS , pp. 2–,

The variable x_{84} depends on x_{80}. By a simple substitution, we get:

$$x_{84} = z_{21} \oplus z_{17} \oplus x_{76} \oplus x_{64} \oplus x_{60} \oplus x_{52} \oplus x_{48} \oplus x_{31} \oplus x_{27} \oplus x_{25} \oplus x_{23} \oplus x_{22} \oplus x_{21} \oplus x_{19} \oplus x_{18}$$

4 Algebraic Attacks

The above observation allows to express each of the variables x_i as a linear combination of the initial state variables and of keystream bits. We denote by L_t the linear expression associated with variable x_t.

Before mounting an algebraic attack against the linearly filtered NFSR, we need to establish a basic property on algebraic immunity of function $g+x_n$, where g is a Boolean function of n variables $(x_0, \ldots, x_{n-1})$ and x_n is an extra Boolean variable.

Theorem 1. *Let g be a Boolean function of n inputs and degree d_g and let h be an annihilator of g, i.e. we have $hg = 0$ or $h(g+1) = 0$. Then $h(x_n+1)$ is an annihilator of $g + x_n$ (resp. $(g + x_n + 1)$) and we have*

$$AI(g + x_n) \leq AI(g) + 1$$

Let us consider the case where $hg = 0$. We have

$$\begin{aligned} h(x_n + 1)(g + x_n) &= hg(x_n + 1) + h(x_n + 1)x_n \\ &= 0 \cdot (x_n + 1) + h \cdot 0 = 0 \end{aligned}$$

The case where h is an annihilator of $(g + 1)$ is similar. This shows that $h(x_n + 1)$ is an annihilator of $g + x_n$ (resp. $(g + x_n + 1)$).

We can now mount an algebraic attack against the linearly filtered NFSR.

Theorem 2. *Let g be a Boolean function. If the filter generator, i.e. a n-bit LFSR filtered by g is vulnerable to an algebraic attack of complexity T that is using M keystream bits and an annihilator of g of degree d, then we can mount an algebraic attack against a linearly filtered NFSR of n bits updated with g by using an annihilator of $g + x_n$ or $g + x_n + 1$ of degree at most $d+1$ with $M' = M + \binom{n}{d+1}$ keystream bits and of complexity upper bounded by $(M')^\omega + n \cdot M'$.*

In order to mount our attack, we first use the technique introduced at the previous section to express each variable x_t as a linear combination of the initial state variables and of keystream bits. We then use the update function of the NFSR g. We have

$$x_{t+n} = g(x_t, \ldots, x_{t+n-1}).$$

By replacing each variable x_t by its linear expression L_t, we get

$$L_{t+n} = g(L_t, \ldots, L_{t+n-1})$$

which is an algebraic equation of degree d_g.

Since we get an algebraic equation for each keystream bit, we can use an annihilator h of g of degree d to mount an algebraic attack using the annihilator of degree at most $d + 1$

$$h(L_t, \ldots, L_{t+n-1}) \cdot (L_{t+n} + 1).$$

Appeared in : SAC 2008, LCNS , pp. 2–,

This attack will use at most M' bits of keystream with

$$M' = \sum_{k=0}^{d+1} \binom{n}{k},$$

and will have time complexity at most

$$n \cdot M' + M'^{\omega}$$

where ω is the exponent of the linear solving algorithm (2.8 for the Strassen algorithm).

However looking directly for low degree annihilators of $g + x_{t+n}$ can allow to derive equations on the initial state of the NFSR of degree lower than $d + 1$. Moreover since for all polynomials of n inputs there exists an annihilator of g of degree at most $\lceil \frac{n}{2} \rceil$, there exists an annihilator of $g + x_n$ degree at most $\lceil \frac{n+1}{2} \rceil$. When n is even, $\lceil \frac{n+1}{2} \rceil$ and $\lceil \frac{n}{2} \rceil + 1$ are equal but when n is odd, $\lceil \frac{n+1}{2} \rceil$ is strictly lower than $\lceil \frac{n}{2} \rceil + 1$. This shows that the bound on the algebraic immunity of $g + x_n$ given by Theorem 1. is not an equality and that it is sometimes possible to find annihilators of $g + x_n$ that have a lower degree than the one achieved by deriving an annihilator of $g + x_n$ from an annihilator of g as presented earlier.

For our example, since g is of degree 6, it is straightforward to derive algebraic equations of degree 6. Moreover we can remark that the degree of $(x_{t+28} \oplus 1)(x_{t+60} \oplus 1)g(x_t, \ldots, x_{t+79})$ is only 4. We consequently derive equations of degree 4 in 80 variables, which allow us to recover the initial state of the NFSR by linearization with a complexity of 2^{49} operations using 2^{21} keystream bits and memory.

5 Correlation Attacks

Our preliminary observation of Section 3, which allows us to mount algebraic attacks against linearly filtered NFSRs, can also be used to build correlation attacks against these schemes.

Theorem 3. *Let g be the Boolean function. If the filter generator, i.e. a n-bit LFSR filtered by g is vulnerable to a correlation attack of complexity T that is using M keystream bits, then a linearly filtered NFSR of n bits updated with g is also vulnerable to a correlation attack of complexity $T + n \cdot M$ that is using M keystream bits.*

In order to mount a correlation attack against the linearly filtered NFSR, we first look for a linear approximation of the update function g. Let us denote by $\mathcal{L}_g$ such a linear approximation and by ϵ its bias. With probability $\frac{1}{2} + \epsilon$ we have

$$x_{t+n} = \mathcal{L}_g(x_t, \ldots, x_{t+n-1}),$$

and by replacing each variable by its linear expression as for algebraic attacks, we get with the same probability

$$L_{t+n} = \mathcal{L}_g(L_t, \ldots, L_{t+n-1}).$$

Appeared in : SAC 2008, LCNS , pp. 2–,

We can mount a correlation attack in order to recover the initial state of the NFSR. Classical techniques of correlation attacks can be applied in order to build parity check equations on a small number of variables, like relations filtering or collision search [20]. In order to improve the efficiency of the correlation attack, a Fast Walsh Transform computation can be used as in [10]. One can also (almost equivalently) notice that the problem of recovering the initial state of the NFSR from the above equations can be viewed as an instance of the Learning Parity with Noise Problem LPN and consequently be solved by the techniques described in [21, 22], where the Fast Walsh Transform is also used in an essential way.

In our example, the best linear approximation of the update function g is

$$\mathcal{L}_g = x_{t+62} \oplus x_{t+60} \oplus x_{t+52} \oplus x_{t+45} \oplus x_{t+37} \oplus x_{t+28} \oplus x_{t+21} \oplus x_{t+14} \oplus x_t.$$

It matches the function g with probability $\frac{594}{1024}$. We can mount two correlation attacks against our scheme: in the first one we filter the linear relations in order to retain only those relations involving the $m < n$ variables x_0 to x_{m-1}, while in the second attack we derive new linear approximation equations (of lower bias) involving $m < n$ unknown variables x_0 to x_{m-1} by combining the available approximate equations pairwise, and retaining only those pairs of relations for which the $n - m$ last coefficients collide. Then in both cases we use a Fast Walsh Transform computation in order to compute the correlation and to determine the correct value of the m bits. The first technique above allows us to recover 40 bits with 2^{52} operations and 2^{42} keystream bits, while the second technique allows us to recover 30 bits with 2^{35} operations and 2^{33} keystream bits.

6 Linearly Filtered NFSR Combined with Non Linearly Filtered LFSRs

While having high non linearity and resistance to algebraic equations, NFSRs have the drawback that it is more difficult, contrary to the case of LFSRs, to prove useful statistical properties like period length or linear complexity. A possible solution to this problem is to combine a NFSR with a LFSR: the LFSR brings its good statistical properties while the NFSR is a highly non-linear component. This is the approach of the stream cipher Grain [17].

This led us to consider linear combinations of a linearly filtered NFSR and one or several non linearly filtered LFSRs. Let us consider a m-bit LFSR of initial state $(y_0, \ldots, y_{m-1})$ filtered by a Boolean function h of degree d_h linearly combined with a linearly filtered NFSR of n bits updated by a Boolean function g of degree d_g.

6.1 Algebraic Attacks

The preliminary observation we made on a single linearly filtered NFSR can be easily adapted to this case. Each keystream bit can be written as

$$z_t = \bigoplus_{k=0}^{n-1} \alpha_k x_{t+k} \oplus h(y_t, \ldots, y_{t+m-1})$$

Appeared in : SAC 2008, LCNS , pp. 2–,

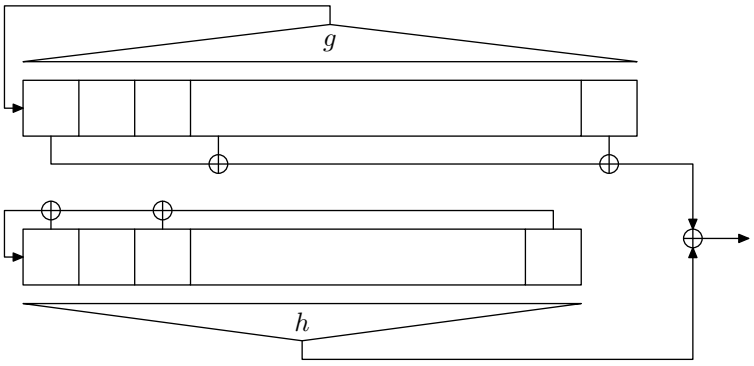

Fig. 4. Combination of a linearly filtered NFSR and a non linearly filtered LFSR

This allows us to write each variable x_t as the sum of a polynomial of degree d_h in the LFSR initial state variables $(y_0, \ldots, y_{m-1})$ and a linear combination of the NFSR initial state variables $(x_0, \ldots, x_{n-1})$ and of keystream bits. The polynomial of degree d_h is a sum of several instances of the function h, and is thus involving LFSR variables y_i. Using the LFSR feedback polynomials, we can express all the instances of h as polynomials of degree d_h in the LFSR initial state variables $(y_0, \ldots, y_{m-1})$.

The number of instances of h in the expression of x_t is equal to the number of keystream bit involved in the expression of x_t and consequently is determined by the difference between the taps of the non linear update function. However this number is growing with t and the complexity of finding the algebraic expression of N variable x_t as a polynomial in the initial state variables of the LFSR and NFSR can be bounded by $n \cdot N^2$.

The extension of our preliminary observation to the considered scheme allows us to derive equations to mount an algebraic attack in the same way as earlier. We replace the expression of each bit x_t in the update function g and we get equations of degree $d_g \cdot d_h$ in $n + m$ variables. Here again classical techniques of algebraic cryptanalysis may allow to reduce the degree of this system of equations. For example finding an annihilator of $g + x_{t+n+1} + 1$ of degree $d < d_g$ allows us to reduce the degree of the equations to $d \cdot d_h$. The total complexity of the attack is $n \cdot M^2 + M^\omega$ where

$$M = \sum_{k=0}^{d} \binom{n+m}{k},$$

and d the final degree of the set of equations.

It is possible to extend algebraic attacks to the case where p non linearly filtered LFSRs are linearly combined with a linearly filtered NFSR. In that case, it is possible to mount an algebraic attack against such a scheme by writing equations of degree at most $d_g \cdot \max_i d_{h_i}$ in $n + \sum_i m_i$ variables, where the $i - th$ of the p LFSRs has size m_i and a filtering function of degree d_{h_i}.

Appeared in : SAC 2008, LCNS , pp. 2–,

6.2 Application to a modified version of Grain-128

In 2006 the eSTREAM project invited the authors of the hardware candidates with a 80-bit key length to present a 128-bit version of their cipher. Grain-128 was introduced by Hell, Johansson, Maximov, and Meier [16] as a response to this invitation. It is built on the same principle as Grain, but uses a 128-bit key and 128-bit IVs: it uses a 128-bit NFSR updated by a function g, a 128-bit LFSR, and an output function. We denote the NFSR internal state at clock t by $X_t = (x_t, \ldots, x_{t+127})$ and the LFSR internal state at clock t by $Y_t = (y_t, \ldots, y_{t+127})$.

In order to achieve a very efficient design for hardware purposes, the authors have chosen a small degree for the function g. The update of the NFSR internal state is governed by the relation

$$\begin{aligned} x_{t+128} = y_t \oplus x_t \oplus x_{t+26} \oplus x_{t+56} \oplus x_{t+92} \oplus x_{t+96} \oplus x_{t+3}x_{t+67} \oplus x_{t+11}x_{t+13} \\ \oplus x_{t+17}x_{t+18} \oplus x_{t+27}x_{t+59} \oplus x_{t+40}x_{t+48} \oplus x_{t+61}x_{t+65} \oplus x_{t+68}x_{t+84}. \end{aligned}$$

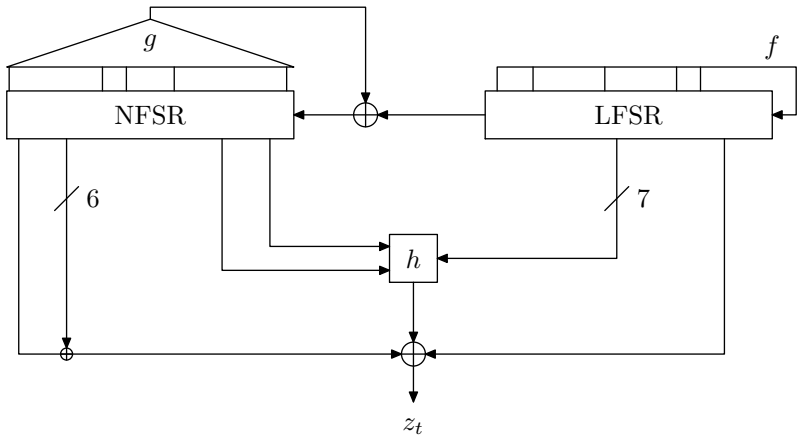

Fig. 5. The original Grain-128

In order to avoid attacks, two bits of the NFSR internal state instead of one in Grainv1 are input to the non-linear output function h:

$$h(X_t, Y_t) = x_{t+12}y_{t+8} \oplus y_{t+13}y_{t+20} \oplus x_{t+95}y_{t+42} \oplus y_{t+60}y_{t+79} \oplus x_{t+12}x_{t+95}y_{t+95}$$

Each keystream bit can be written as a the XOR of a linear combination of the NFSR internal state, a bit of the LFSR internal state, and the output of function h:

$$z_t = L(X_t) \oplus y_{t+93} \oplus h(X_t, Y_t)$$

with

$$L(X_t) = x_{t+2} \oplus x_{t+15} \oplus x_{t+36} \oplus x_{t+45} \oplus x_{t+64} \oplus x_{t+73} \oplus x_{t+89}$$

Appeared in : SAC 2008, LCNS , pp. 2–,

In the paper describing Grain-128, the authors discuss the resistance of the algorithm to algebraic attacks: "In Grain-128, an NFSR is used to introduce nonlinearity together with the function h(). Solving equations for the initial 256 bit state is not possible due to the nonlinear update of the NFSR. The algebraic degree of the output bit expressed in initial state bits will be large in general and also varying in time. This will defeat any algebraic attack on the cipher."

We now introduce a modified version of Grain-128. In this version, we replace the two bits of the NFSR internal state that were input into the non-linear output function h by two bits of the LFSR internal state. As stated by the authors of Grain-128, the non-linearity of the algorithm comes from the NFSR and from this function h. We now present an algebraic attack against this modified version, which shows that with the modified version of the function h it is possible to write equations of constant degree even tough the non-linear update of the NFSR is supposed to make the degree vary in time, as claimed by the authors of Grain-128.

For the modified version, the new function h depends only of the LFSR internal state. In order to keep the properties of function h in particular the number of variables and the degree, we replace x_{t+12} and x_{t+95} by y_{t+12} and y_{t+94}. This choice of the two new taps is not significant for our attack.

$$h(Y_t) = y_{t+8}y_{t+12} \oplus y_{t+13}y_{t+20} \oplus y_{t+42}y_{t+94} \oplus y_{t+60}y_{t+79} \oplus y_{t+12}y_{t+94}y_{t+95}$$

In order to keep the keystream dependent on the two bits x_{t+12} and x_{t+95}, we add them to the linear part of the output. Consequently each keystream bit can now be written as:

$$z_t = L(X_t) \oplus x_{t+12} \oplus x_{t+95} \oplus y_{t+93} \oplus h(Y_t)$$

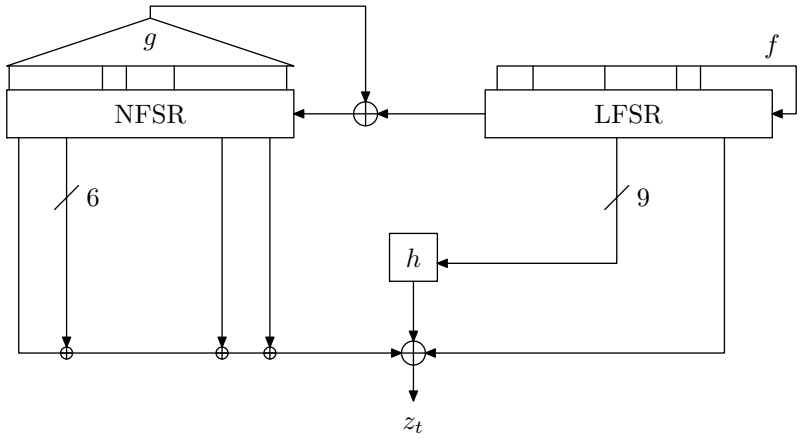

Fig. 6. Modified version of Grain-128

This modified version is almost identical to the case illustrated in Figure 4. The only difference is the influence of the LFSR output on the update of the NFSR. Using the

Appeared in : SAC 2008, LCNS , pp. 2–,

technique described for algebraic attacks against combination of a NFSR and a LFSR, we can express each variable bit of the NFSR internal state x_t as the sum of a polynomial of degree d_h in the LFSR initial state variables $(y_0, \ldots, y_{127})$, of a linear combination of the NFSR initial state variables $(x_0, \ldots, x_{127})$ and of keystream bits. The polynomial of degree d_h is a sum of several instances of the function h and of linear combinations of the LFSR initial state variables stemming from the term y_{t+93} for different values of t. As already explained this allows us, by replacing the expression of each bit x_t and of the bit y_t inside of the update function of the NFSR, to write equations of degree $d_g \cdot d_h = 6$ in the 256 variables of the LFSR and NFSR initial states. By linearizing these equations, we can recover the 256 variables in time $128 \cdot M^2 + M^{2.73}$, where

$$M = \sum_{k=0}^{6} \binom{256}{k}.$$

This result in an attack of complexity 2^{105} using 2^{39} keystream bits.

The described attack is not applicable to the original Grain-128 due to the non-linearity in the NFSR state variables. However it shows that the argument used by the authors to justify the immunity against algebraic attack was incomplete.

6.3 Correlation Attacks

Unlike the simple linearly filtered NFSR case, it seems difficult to mount correlation attack against a combination of a linearly filtered NFSR and non linearly filtered LFSRs by using our preliminary observation. This is due to the increasing number of instances of the h function. If the functions g and h are replaced by linear approximations, the resulting bias is decreasing as the number of the instances of h grows. Consequently unless the biases for g and h are very strong and the number of required keystream bits is very low, it seems difficult to mount correlation attacks against these schemes by using our preliminary observation.

It is however possible to mount a correlation attack in the special case where the linear filtering function of the NFSR has one single non-zero coefficient. In that case we have

$$z_t = x_t \oplus h(y_t, \ldots, y_{t+m-1}).$$

By using a linear approximation $\mathcal{L}_g$ of bias ϵ_g and weight w and a linear approximation $\mathcal{L}_h$ of bias ϵ_h, we can derive approximate relations

$$\begin{aligned} x_{t+n} &\simeq \mathcal{L}_g(x_t, \ldots, x_{t+n-1}) \\ &\simeq \bigoplus_{j=0}^{w} x_{t+i_j} \\ z_{t+n} \oplus h(y_{t+n}, \ldots, y_{t+n+m-1}) &\simeq \bigoplus_{j=0}^{w} z_{t+i_j} \oplus h(y_{t+i_j}, \ldots, y_{t+i_j+m-1}) \end{aligned}$$

Appeared in : SAC 2008, LCNS , pp. 2–,

By replacing the non-linear outputs of h with its linear approximation $\mathcal{L}_h$, we get approximate relations

$$z_{t+n} \oplus \mathcal{L}_h(y_{t+n}, \ldots, y_{t+n+m-1}) \simeq \bigoplus_{j=0}^{w} z_{t+i_j} \oplus \mathcal{L}_h(y_{t+i_j}, \ldots, y_{t+i_j+m-1}),$$

which can be re-expressed as approximate relations involving the initial state bits of the LFSR. By using the Piling up Lemma, we can compute the equivalent bias of approximate relations. We get:

$$\epsilon = \epsilon_g (2\epsilon_h)^{w+1}$$

Consequently if the bias ϵ_g and ϵ_h are large enough and if the weight w of the linear approximation of g is small enough, a correlation attack is possible against the construction of Figure 4 in this special case.

7 Conclusion

In this paper, we have shown that the dual case of the filter generator, i.e. a linearly filtered NFSR, is vulnerable to the same kind of attacks than the filter generator. We described how to mount algebraic and correlation attacks against this scheme. These attacks were illustrated on an example NFSR taken from the Grain stream cipher. We then extended these attacks to combinations of a linearly filtered NFSR and one or several non linearly filtered LFSRs. We illustrated the latter extension by an algebraic attack on a modified version of Grain-128, which can be broken in 2^{105} computations with 2^{39} keystream bits. This attack is not applicable to the original Grain-128 but is shows that the use of a NFSR is not sufficient to avoid all algebraic attacks. As far as we know, none of these attacks is directly applicable to stream cipher candidates submitted to the eSTREAM competition or recognized stream ciphers like SNOW 2.0 or MUGI.

The techniques presented in this paper can be easily extended to ciphers in which $t > 1$ bits of the current state are non-linearly updated at each step while t or more linear combinations of the state bits are output as keystream bits. It is an open question whether those attack techniques can be also extended to ciphers in which $t > 1$ state bits are non-linearly updated, while only $t' < t$ linear combinations of the state bits are output.

References

1. F. Armknecht and M. Krause. Algebraic Attacks on Combiners with Memory. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 162–175. Springer, 2003.
2. G. Ars and J. Faugère. An Algebraic Cryptanalysis of Nonlinear Filter Generators using Gröbner Basis. Technical report, INRIA, 2003.
3. G. Ars and J.-C. Faugère. An algebraic cryptanalysis of nonlinear filter generators using groebner basis. INRIA, 2003.
4. C. Berbain. *Analyse et conception d'algorithmes de chiffrement flot*. PhD thesis, Université Paris.Diderot (Paris 7), 2007.

Appeared in : SAC 2008, LCNS , pp. 2–,

5. C. Berbain, H. Gilbert, and A. Maximov. Cryptanalysis of grain. In M. J. B. Robshaw, editor, *Fast Software Encrytion - FSE 2006*, volume 4047 of *Lecture Notes in Computer Science*, pages 15–29. Springer-Verlag, 2006.
6. A. Braeken and J. Lano. On the (Im)Possibility of Practical and Secure Nonlinear Filters and Combiners. In *Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 159–174. Springer, 2005.
7. A. Braeken, J. Lano, N. Mentens, B. Preneel, and I. Verbauwhede. SFINKS: A Synchonous Stream Cipher for Restricted Hardware Environments. eSTREAM, ECRYPT Stream Cipher Project, 2005.
8. C. De Cannière and B. Preneel. Trivium: Specifications. eSTREAM, ECRYPT Stream Cipher Project, 2005.
9. A. Canteaut and M. Trabbia. Improved Fast Correlation Attacks Using Parity-Check Equations of Weight 4 and 5. In B. Preneel, editor, *Advances in Cryptology—EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 573–588. Springer-Verlag, 2000.
10. P. Chose, A. Joux, and M. Mitton. Fast Correlation Attacks: An Algorithmic Point of View. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 209–221. Springer, 2002.
11. N. Courtois. Fast Algebraic Attacks on Stream Ciphers with Linear Feedback. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 176–194. Springer, 2003.
12. N. Courtois and W. Meier. Algebraic Attacks on Stream Ciphers with Linear Feedback. In Eli Biham, editor, *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 345–359. Springer, 2003.
13. H. Englund and T. Johansson. A New Simple Technique to Attack Filter Generators and Related Ciphers. In *Selected Areas in Cryptography*, pages 39–53, 2004.
14. G. Gong and Y. Nawaz. The WG Stream Cipher. eSTREAM, ECRYPT Stream Cipher Project, 2005.
15. A. Gouget and H. Sibert. Revisiting correlation-immunity in filter generators. In *Selected Areas in Cryptography - SAC 2007*, volume 4876 of *Lecture Notes in Computer Science*, pages 378–395. Springer-Verlag, 2007.
16. M. Hell, T. Johansson, A. Maximov, and W. Meier. A Stream Cipher Proposal: Grain-128. eSTREAM, ECRYPT Stream Cipher Project, 2006.
17. M. Hell, T. Johansson, and W. Meier. Grain - A Stream Cipher for Constrained Environments. eSTREAM, ECRYPT Stream Cipher Project, 2005.
18. T. Johansson and F. Jönsson. Fast Correlation Attacks Based on Turbo Code Techniques. In *Advances in Cryptology—CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 181–197. Springer-Verlag, 1999.
19. T. Johansson and F. Jönsson. Improved Fast Correlation Attacks on Stream Ciphers via Convolutional Codes. In *Advances in Cryptology—EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 347–362. Springer-Verlag, 1999.
20. S. Leveiller, G. Zémor, P. Guillot, and J. Boutros. A New Cryptanalytic Attack for PN-generators Filtered by a Boolean Function. In *Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 232–249. Springer, 2002.
21. E. Levieil and P.A. Fouque. An Improved LPN Algorithm. In Roberto De Prisco and Moti Yung, editors, *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 2006.
22. Vadim Lyubashevsky. The Parity Problem in the Presence of Noise, Decoding Random Linear Codes, and the Subset Sum Problem. In Chandra Chekuri, Klaus Jansen, José D. P. Rolim, and Luca Trevisan, editors, *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation*

Appeared in : SAC 2008, LCNS , pp. 2–,

Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings, volume 3624 of *Lecture Notes in Computer Science*, pages 378–389. Springer, 2005.

23. W. Meier and O. Staffelbach. Fast Correlation Attacks on Stream Ciphers. In C.G. Günter, editor, *Advances in Cryptology—EUROCRYPT'88*, volume 330 of *Lecture Notes in Computer Science*, pages 301–316. Springer-Verlag, 1988.
24. T. Siegenthaler. Correlation-immunity of Non-linear Combining Functions for Cryptographic Applications. *IEEE Transactions on Information Theory*, 30:776–780, 1984.

Appeared in : SAC 2008, LCNS , pp. 2–,