Published in *Progress in Cryptology - Indocrypt 2005* LNCS 3797

Two algebraic attacks against the F-FCSRs using the IV mode

Thierry P. Berger
LACO, Facult des Sciences de Limoges
23 avenue Albert Thomas, F-87060 Limoges Cedex, FRANCE
email: thierry.berger@unilim.fr

Marine Minier
INSA Lyon, CITI
21 Avenue Jean Capelle, F-69621 Villeurbanne Cedex, FRANCE
email: marine.minier@insa-lyon.fr

Abstract. This article presents some new results concerning two algebraic attacks against the F-FCSR constructions proposed in [2]. We focus on the parameters of the stream ciphers proposed that permit to mount algebraic attacks when using the IV mode. The complexity obtained for the first attack described here is 2^{45} binary instructions using 2^{15} known IV values for the construction F-FCSR-SF1. All the proposed attacks are full key recovery attacks. We do not contest that the FCSRs are a good and new idea, we just say that the chosen parameters do not ensure the security level claimed.
Keywords: stream cipher, cryptanalysis, algebraic attack.

Introduction

In [8] and [7], a new class of attacks called "algebraic attacks" was introduced. Those cryptanalyses use the fact that the relation between the initial state constructed in the general case from the key and the internal state at time t is linear. Then an attacker could construct a huge system of equations from the observed output words using the previous remark and he does not have any more but to solve the obtained system.

So finding non linear transition functions for stream ciphers becomes urgent. Some propositions called T-functions were made by A. Klimov and A. Shamir in [12–14]. An other possible choice proposed in [2] and in [3] is to use an FCSR: a binary automaton with carries. All the results concerning the complexity, the provided period comes from the 2-adic theory. In [2], the authors proposed four constructions (F-FCSR-SF1, F-FCSR-SF8 F-FCSR-DF1 and F-FCSR-DF8) based on a same simple construction called F-FCSR. Two others constructions called F-FCSR-8 and F-FCSR-H based on the same principles were proposed in the call for stream cipher primitives of the European Network of Excellence ECRYPT (see [3] and [17]).

In this paper, we propose two algebraic attacks with known IV values against the F-FCSRs based upon a bad choice of the parameters when using

the IV mode for the constructions proposed in [2] (not for the one described in [3]): even if the transition function is not linear, the degree between the key bits and the first output bit is very low. The first attack proposed is a traditional one but the second one uses some particular properties of the structure of the FCSRs. In fact, we could, if we made an exhaustive search on some particular key bits, control and lower the degree between the key bits and the first output bit. These two attacks are full key recovery attacks.

We want to point out one more time that those attacks do not threaten the FCSRs themselves but just shows that the parameters of [2] were not carefully chosen. We do not contest the security level provided by the FCSR (especially against the algebraic attacks), we only claim that the security margin induced by the total construction proposed in [2] is not sufficient. Notice also that two attacks with chosen IVs against the constructions proposed in [2] will be presented at SAC'05 by E. Jaulmes and F. Muller (see [10]). They have also studied the version presented in [3] and published their analyses on the ECRYPT web-site (see [9]).

This paper is organized as follows: after a short recall about the FCSRs themselves and about the constructions proposed in [2] and in [3], Section 2 describes the particular properties used to mount the proposed attacks whereas Section 3 describes the two proposed algebraic attacks.

1 Background on the F-FCSRs

The Feedback with Carry Shift Registers were introduced first by Klapper and Goresky in [11]. In [2], T. Berger and F. Arnault proposed to use them as the transition function of a filtered stream cipher. We first recall how an FCSR automaton works. For more details on the F-FCSRs, the reader could refer to [1, 2].

1.1 The FCSR automaton

Let q be a negative integer such as $|q|$ is prime and p be a number such as $0 \leq p < |q|$. Then you could write p as $p = \sum_{i=0}^{n-1} p_i 2^i$ and $d = \frac{1-q}{2} = \sum_{i=0}^{n-1} d_i 2^i$. The FCSR automaton with feedback prime q and an initial value p produces the 2-adic expansion of p/q that could be seen as an infinite sequence of bits a_i such as (see [15]):

$$p = q \cdot \sum_{i=0}^{\infty} a_i 2^i$$

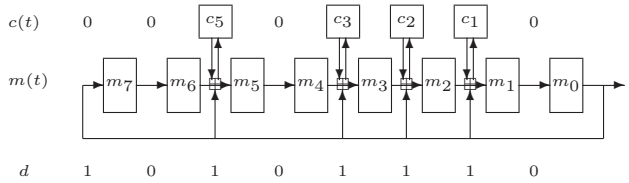
Let us consider the sequence of integers $p(t)$ defined by: $p(0) = p$, $p(t+1) = (p(t) - qa_i)/2$. It is easy to verify that $0 \leq p(t) < -q$ and $p(t)/q = \sum_{j=t}^{\infty} a_j 2^j$.

The sequences (a_i) and $(p(t))$ could be generated from an FCSR automaton defined using two registers (sets of cells): a main register M and a carry register C.

The main register M contains n binary cells, each bit is denoted by $m_i(t)$ $(0 \leq i \leq n-1)$. We call the integer $m(t) = \sum_{i=0}^{n-1} m_i(t)2^i$ the content of M.

The carry register contains ℓ cells where $\ell + 1$ is the number of nonzero d_i digits, i.e. the Hamming weight of d. More precisely, the carry register contains one cell for each nonzero d_i with $0 \leq i \leq n-2$. We denote $c_i(t)$ the binary digit contained in this cell. We put $c_i(t) = 0$ when $d_i = 0$ or when $i = n-1$. We call the integer $c(t) = \sum_{i=0}^{n-2} c_i(t)2^i$ the content of C. The Hamming weight of the binary expansion of $c(t)$ is at most ℓ. Note that, if $d_i = 0$, then $c_i(t) = 0$ for all t. We denote by $c_{j_1}(t), \ldots, c_{j_\ell}(t)$ the active carries cells, i.e. the ℓ cells corresponding with $d_i = 1$. We have $c(t) = \sum_{i=1}^{\ell} c_{j_i}(t)2^{j_i}$.

A simple example with $q = -347$, $d = 174 = 0\text{xAE}$, $k = 8$ and $\ell = 4$ is described on the figure just bellow.



The symbol $\boxplus$ denotes the addition with carry.

The transition function of the registers could be written

$$m(t+1) = (m(t))_{<<1} \oplus c(t) \oplus m_0(t)d \tag{1}$$

$$c(t+1) = (m(t))_{<<1} \otimes c(t) \oplus c(t) \otimes m_0(t)d \oplus m_0(t)d \otimes (m(t))_{<<1} \tag{2}$$

where $\oplus$ denotes bitwise XOR, $\otimes$ denotes bitwise AND, and $<< 1$ is a simple shift to the left.

Note that $m_0(t)$ is the least significant bit of $m(t)$ and represents the feedback bit. The integers $m(t)$, $c(t)$ and d are integers of bit-size n (or less).

So if $m(0) = p$, at time t, the following relations are always satisfied:

$$p(t) = m(t) + 2c(t).$$

The transition function could also be described at the cell level:

$$m_i(t+1) = m_{i+1}(t) \oplus d_i c_i(t) \oplus d_i m_0(t) \tag{3}$$

$$c_i(t+1) = d_i \left(m_{i+1}(t)c_i(t) \oplus c_i(t)m_0(t) \oplus m_0(t)m_{i+1}(t)\right) \tag{4}$$

The period T of the FCSR automaton is maximal if $|q|$ is prime and the order of 2 modulo q is exactly $|q| - 1$. In that case, T is equal to $|q| - 1$, so we have: $2^n < T < 2^{n+1} - 1$. The number of the possible states of the FCSR automaton is $2^{n+\ell}$.

In [2], the authors proposed to use the following parameters $n = 128$ and $\ell = 68$ before to apply on the chosen FCSR a filtering function. They choose the prime number q equal to:

$$q_1 = -493877400643443608888382048200783943827$$

In [3], the authors proposed another primitive called F-FCSR-H and designed for hardware utilization with a register length equal to $n = 160$ bits. The corresponding connection integer is:

$$q_2 = -199352459131827501532804161134421503646014008796 3$$

that corresponds with $n = 160$ and $\ell = 82$.

1.2 The proposed constructions

The four constructions proposed in [2] filter some of the bits of the main register with $q = q_1$ in the following way:

- **F-FCSR-SF1:** The filtering function is known and consists of a linear function $f = (f_0, \cdots, f_{n-1})$ on $GF(2)^n$. If $s(t)$ denotes the output bit at time t, we have: $s(t) = \bigoplus_{i=0}^{n-1} f_i \cdot m_i(t)$.
- **F-FCSR-SF8:** The filtering function is also known but the aim here is to output one byte, so the filtering function consists in 8 sub-filters $F_0, \cdots, F_7$ on 16 bits linearly independent and publicly known. The output byte $S(t)$ is then the XOR at sixteen bits level between the eight sub-filters and the main register M folded at byte level.
- **F-FCSR-DF1:** This is the same construction as SF1 but, this time, the filter is unknown and derived from the key.
- **F-FCSR-DF8:** This is the same construction as SF8 but, this time, the filter is unknown and the eight sub-filters are derived from the key.

In [3], the authors proposed a first construction called **F-FCSR-8** corresponding with the F-FCSR-DF8 construction. The second construction submitted called **F-FCSR-H** corresponds with the case where $n = 160$, $l = 82$ with the q value equal to q_2 and F-FCSR-SF8 (the 8 sub-filters are constructed using the d value) is applied with a key setup and an IV injection defined as follows: $M = K + 2^{80} \cdot IV$. The carry register C is initialized to 0 and 128 iterations are discarded at each IV change (for the details of the used filter see [3]).

In the previous proposed constructions, the initialization of the FCSR using the key K of length $l_K = n$ is $m(0) = K$ and $c(0) = 0$: $p(0) = m(0) + 2 \cdot c(0) = K$.

1.3 Description of the IV mode

An IV mode is also proposed in [2] where the IV value is directly injected in the cells of the carry register at bit level whereas the key is injected in the main register and in the filter if required according the F-FCSR version used. After this initialization, the FCSR is clocked 6 times and the 6-th output bit or byte becomes the first output bit or byte according the version we use.

This article focus on some algebraic attacks using this IV mode: the number of clocks is not sufficient to prevent the stream-cipher from this kind of attacks: the degree of the first output is too small.

2 Particular algebraic properties of the FCSR automaton

We focus on this section on several algebraic properties of the FCSR that will be used to mount the cryptanalyses presented in section 3.

2.1 Some results on the degree and the number of monomials of algebraic equations

We consider here that at time $t \geq 0$, the main register $M = m(t)$ is composed of $m_i(t)$ with $i \in [0..n-1]$ and the carry register $C = c(t)$ of $c_{j_i}(t)$ with $i \in [0..\ell - 1]$. These values $m_i(t)$ and $c_{j_i}(t)$ could be seen as polynomials in the first indeterminates $(m_0(0), \cdots, m_{n-1}(0), c_{j_1}(0), \cdots, c_{j_\ell}(0))$. In order to perform algebraic attacks on the FCSR, we are going to study in this section the degree and the number of monomials occurring in these polynomials.

We denote by $deg(m(t))$ and $deg(c(t))$ the maximum of the degree of each $m_i(t)$, resp. $c_{j_i}(t)$ in terms of the monomials constructed from the unknowns $(m_0(0), \cdots, m_{n-1}(0), c_{j_1}(0), \cdots, c_{j_\ell}(0))$.

Lemma 1. *The following relations on the degree are satisfied:*

$$deg(m(t+1)) \leq Max(deg(m(t)), deg(c(t)))$$
$$deg(c(t+1)) \leq Max(deg(m(t)) + deg(c(t)), 2 \cdot deg(m(t)))$$

Proof: It is a direct consequence of the equations (1) and (2). □

Proposition 1. *We have* $deg(m(t)) \leq Fib(t)$ *and* $deg(c(t)) \leq Fib(t+1)$, $\forall t \geq 1$ *where* $Fib(t+1)$ *is the* $(t+1)$*-th term of the Fibonacci sequence such as* $Fib(0) = 1$ *and* $Fib(1) = 1$.

Proof: This proof could be made by induction:
At $t = 0$, we have $deg(m(0)) = 1 = Fib(0)$ and $deg(c(0)) = 1 = Fib(1)$.
Now, suppose that the relations $deg(m(t)) \leq Fib(t)$ and $deg(c(t)) \leq Fib(t+1)$ hold for t. Using Lemma 1, we deduce

$$deg(m(t+1)) \leq Max(Fib(n), Fib(n+1)) = Fib(n+1)$$
$$deg(c(t+1)) \leq Fib(n) + Fib(n+1) = Fib(n+2)$$

□

This bound is just an upper bound that could only be reached if $d_0 = 1$. In the FCSRs we study, due to the fact that $|q|$ is prime for a security aim (see [1, 2]), d is always even and $d_0 = 0$. So, the degree of $m(t)$ for the FCSR defined using q_1 is under this bound.

We are also interested in the number of distinct monomials that can occur in the $m(t)$ and in the $c(t)$ polynomials.

Proposition 2. *The polynomials $m_i(t)$ and $c_i(t)$ only depend on the indeterminates $(m_0(0), \cdots, m_{t-1}(0), c_0(0), \cdots, c_{t-1}(0)$ and $(m_{i+1}(0), \cdots, m_{i+t}(0), c_i(0), \cdots, c_{i+t-1}(0))$.*

Proof: This result could be obtained by induction using equations (3) and (4). □

The most important consequence of this result is the fact that, even if the degree of an algebraic equation is s, this equation does not contain all the monomials of degree less or equal to s, but only a small part of them.

It seems difficult to determine exactly the number of monomials given by Proposition 2 but we have computed the degree of algebraic equations and the number of distinct monomials occurring in the main register (i.e. in the polynomials $m_i(t)$ for the value of $q = q_1$ given in [2] and $0 \leq t \leq 6$). For example, if $t = 6$, the degree of $m(t)$ is 10 and the maximal number of monomials of $m(6)$ is 274891. (Note that, due to the complexity of computing the algebraic equations, we were not able to obtain these values for $t \geq 7$.)

2.2 Algebraic equations with known carries

We have seen in Section 1.3 that in the IV mode described in [2], the initial contents of carries are known, i.e. $c_{j_1}(0), \cdots, c_{j_\ell}(0)$ become fixed values. From Proposition 2, we deduce the following corollary:

Corollary 1. *If the initial contents of carries $c_{j_1}(0)$, ..., $c_{j_\ell}(0)$ are known, the polynomials $m_i(t)$ only depend $(m_0(0), \ldots, m_{t-1}(0), m_{i+1}(0), \ldots, m_{i+t}(0))$ for $t \geq 1$. The maximal degree of $m(t)$ satisfies the upper bound $deg(m(t)) \leq 2t$.*

As previously, this upper bound is not reached as soon as $\ell < n$. We have computed the degree and the number of distinct monomials in $m(t)$ for the value of $q = q_1$ given in [2] and $0 \leq t \leq 7$ (see Table 1) and have compared them with the usual upper bound given by the sum of the binomial coefficients ($\sum_{i=0}^{d} C_{128}^i$ where d is the degree given in Table 1).

Table 1. $m_i(0)$ unknown, $c_i(0)$ known.

nb of iterations	0	1	2	3	4	5	6	7
nb of monomials	128	129	256	758	2490	8830	32836	125420
Degree in $m_i(0)$	1	1	2	3	4	6	8	10
Binomial coefficient	129	129	8257	349633	11017633	$\approx 2^{32}$	$\approx 2^{40}$	$\approx 2^{48}$

In the second attack presented here, we use a stronger property based on the fact that the knowledge of some feedback bits limits the increase of the degree and the number of monomials. This knowledge is equivalent to those of $m_0(0)$, $m_1(0)$, ...,$m_{t-1}(0)$. We suppose now that not only the initial values of

the carries are known but also the t values $m_0(0)$, $m_1(0)$, ...,$m_{t-1}(0)$ of the main register.

Proposition 3. *Suppose that* $c_{j_1}(0)$, ..., $c_{j_\ell}(0)$, $m_0(0)$, ..., $m_{t-1}(0)$ *are known. For* $1 \leq s \leq t$, *the monomials occurring in* $m_i(s)$ *are those obtained from the set of indeterminates* $\{m_{i+1}(0), \cdots, m_{i+t}(0)\}$ *and of degree strictly less than* s. *The degree of* $m(s)$ *satisfies the relation* $deg(m(s)) < s$.

Moreover, if I_s *denotes the set of all possible monomials of* $m_i(s)$, *then the size of* I_s *can be computed by the following recurring relations:* $\#\{I_1\} = \#\{I_2\} = n - t + 1$, *and* $\#\{I_{s+1}\} = 2.\#\{I_s\} - 2^{s-1}$, $\forall s \ / \ 2 \leq s < t$.

Proof: The first part of the proposition is a direct consequence of the Proposition 2 considering that the variables $c_{j_1}(0), \cdots, c_{j_\ell}(0)$ and $m_0(0),\cdots,m_{t-1}(0)$ are known.

Using the equations (3) and (4), it is easy to verify that $deg(m(1)) = deg(m(2)) = 1$, and then that the number of possible monomials is $n - t + 1$ (including the constant term 1).

From equations (3) and (4) and from the knowledge of $c_{j_1}(0), \ldots, c_{j_\ell}(0)$ and $m_0(j)$ for $0 \leq j < t$, we deduce that $deg(m(s+1) = deg(c(t)) \leq deg(m(t)) + 1$. It implies that $deg(m(s)) \leq s - 1$ for $1 < s < t$. We also deduce from the same equations that

$$I_1 = I_2 = \{1, m_t(0), m_{t+1}(0), \ldots, m_{n-1}(0)\}.$$

The monomials of I_s are exactly those of the form $m_{i_1}(0)m_{i_2}(0)\cdots m_{i_r}(0)$, with $t \leq i_1 < i_2 < \ldots < i_r < n$, $r < s$ and $i_r - i_1 < s$.

Clearly I_s is a subset of I_{s+1}. Moreover, the new monomials of I_{s+1} are obtained in the following way: each monomial $m_{i_1}(0)m_{i_2}(0)\ldots m_{i_r}(0)$ corresponds to a new one $m_{i_1}(0)m_{i_2}(0)\ldots m_{i_r}(0)m_{i_1+s}(0)$. It is possible if and only if $i_1 < n - s$. There are 2^{s-1} monomials in I_d such that $i_1 \geq n - s$. This gives the recurring relation $\#\{I_{s+1}\} = 2.\#\{I_s\} - 2^{s-1}$. □

The so obtained bound $b = \#\{I_t\}$ is a good approximation on the number of monomials in the algebraic equations after t iterations.

Table 2 gives the results obtained with q_1, $t = 6$, $c_{j_1}(0) = \ldots = c_{j_{68}}(0) = 1$ and $m_0(0) = \ldots = m_5(0) = 1$. (Notice that all the values of the second row of this table reach the bound $\#\{I_s\}$.)

Table 2. $m_i(0)$ known for $i := 0$ to 5, $c_i(0)$ known.

nb of iterations s	0	1	2	3	4	5	6
nb of monomials in $m(s)$	123	123	123	244	484	960	1904
$deg(m(s))$	1	1	1	2	3	4	5

3 Algebraic attacks with known IV values

The two attacks presented in this section are attacks with known IV values.

3.1 General principle of an algebraic attack

Algebraic attacks were introduced by N. Courtois and W. Meier in [8] and in [7] and exploit the fact that the dependence between the key bits and the internal states at time t is linear when using an LFSR. Suppose, for example, that the key K is directly injected in the first initial state of size n at $t = 0$: $Init_0 = (K_0, \cdots, K_{n-1})$ where $(K_0, \cdots, K_{n-1})$ is the representation of K at bit level. Suppose also that the output bit (or word) $s(t)$ could be written at time t: $s(t) = f(L^t(K_0, \cdots, K_{n-1}))$ where f is a boolean function from $GF(2)^m$ into $GF(2)^k$ and L is the linear transition function. Then, you could built a system of equations of degree $deg(f)$ for different t values where the unknown variables are the key bits. It is possible to solve the obtained system using a relinearization technique (see [5]) or a dedicated algorithm using the Gröbner basis (see [4]).

There are many improvements of such techniques: you could find some low degree multiples of f to lower the general degree of the built system (see [16]), try to find a relation using several output words (see [7]) and so on.

If an FCSR is used as a transition function, the problem becomes more difficult due to the fact that this transition function is no more linear. However, if the number of iterations is sufficiently small, the degree of the corresponding system linking the output words and the key bits stays reasonable. Moreover, all the filtering function proposed in [2] and in [3] are linear and do not increase the degree of the system. More formally, the obtained system could be written as: $s(t) = f(T^t(K_0, \cdots, K_{n-1}))$ where f is a linear function from $GF(2)^n$ into $GF(2)^k$ ($k = 1$ or 8 for the constructions studied here) and where T is the FCSR transition function: the degree of the t-th equation depends on the degree of T^t. The first equation at time $t = 0$ is linear, the following one is quadratic and the degree increases at each clock according the relation demonstrated in Section 2.

3.2 A first simple attack

The principle of this attack is very simple: the first output after a change of a known IV gives an algebraic equation which can be computed, since there are only 6 iterations before the first output.

So, suppose that, as described in [2], the initial value of the main register is $m(0) = (m_0(0), \cdots, m_{127}(0)) = (K_0, \cdots, K_{127})$ where each K_i, $\forall i \in [0..127]$ denotes a key-bit of the key K and that the initial value of the carry register denoted by $c(0) = (c_0(0), \cdots, c_{67}(0))$ is known and is equal to $IV = (IV_0, \cdots, IV_{67})$. In [2], the first output $s(t) = s(6)$ (that could be a bit or a byte) is computed after six clocks, the previous outputs being discarded. So, we could construct the following simple algebraic attack against all the constructions proposed in [2] when using the IV mode:

- For a subset of N known IV values, compute the first output bit (or byte) $s(6)$ and generate the corresponding system with N' equations.
- Linearize the obtained system and use a Gaussian elimination to solve it.
- When you find a solution, test the obtained key for a known IV by generating few key-stream bits.

So, the complexity of this attack is about $(N')^3$ basic binary instructions. Let us now determinate the required number of known IV values for the four constructions described in Section 1.2.

First, we have seen in Section 2 that the number of possible monomials after 6 clocks, given in Table 1 is $32836 \simeq 2^{15}$. So, in the case of **F-FCSR-SF1**, this number corresponds exactly to the number of unknowns due to the fact that the filter is linear and completely known. So, the complexity of the previous attack is about 2^{45} basic binary operations for a number of known IV values equal to $N = N' = 2^{15}$.

For the **F-FCSR-SF8** construction, the number of monomials depending on the key bits is always the same 2^{15} but less known IV values are required: each output byte $S(6)$ gives 8 equations. So, the complexity is the same than previously but the number of known IV values required is equal to $N = 2^{12}$ whereas the number of equations is always the same $N' = 2^{15}$.

In the case where the **F-FCSR-DF1** construction is used, the filter is dynamic and constructed from the key. So we could consider it as 128 unknown coefficients denoted by f_i, $\forall i \in [0..127]$:

$$s(6) = \bigoplus_{i=0}^{127} f_i \cdot m_i(6).$$

So, taking into account the f_i values as unknowns, the number of monomials is multiplied by a factor 128 and then we need about $N' \simeq 128 \cdot 2^{15} = 2^{22}$ equations generated from $N = 2^{22}$ known IV values for a complexity equal to $2^{22 \cdot 3} = 2^{66}$ basic binary instructions.

In the last case (**F-FCSR-DF8**), where the filter is unknown and derived from the key and where the output is one byte, you could write the output bits in the following way:

$$S_j(6) = \bigoplus_{i=0}^{15} f_{8j+i} \cdot m_{8j+i}(6)$$

for $j = 0, \cdots, 7$.

So if you only consider one output bit (the first one for example), the number of unknowns added is only 16 instead of 128. So, the number of monomials is multiplied by a factor 16 and you need 2^{19} known IV values to generate 2^{19} equations for a complexity equal to $2^{3 \cdot 19} = 2^{57}$ basic binary instructions.

All the previous results are summed up in Table 3.

Table 3. First attack

Algorithm	Attack	Complexity	Data
F-FCSR-SF1 IV mode	algebraic	2^{45}	2^{15}
F-FCSR-DF1 IV mode	algebraic	2^{66}	2^{22}
F-FCSR-SF8 IV mode	algebraic	2^{45}	2^{12}
F-FCSR-DF8 IV mode	algebraic	2^{57}	2^{19}

3.3 Improving the previous attack

We have seen in Section 2.2 that we could lower the degree and so the number of monomials of $m(t)$ by knowing t feedback bits. So, we could improve the previous attack by sharing it in two parts, first we perform an exhaustive search on $t = 6$ feedback bits (i.e. the bits $m_0(0), \cdots, m_5(0)$) by generating, for each value, a system of equations and after by solving this simpler system.

The algorithm is then the following one:

- for each possible value of $m_0(0), \cdots, m_5(0)$ do
- for N known IV values, compute the first corresponding output word $s(6)$ (a bit or a byte).(In case you use F-FCSR-DF8, take only into account the first output bit $S_0(t)$.)
- generate the system of N' equations
- solve the corresponding system by linearization
- when you find a solution, test the obtained key by generating few keystream bits.

We detail here the complexity of a such attack for F-FCSR-DF1 (the details of the other cases are left to the reader). The number of monomials is $1904 \approx 2^{10.89}$ (c.f. Table 2). So, taking into account the 128 unknown bits of the filter, $N = 128 \cdot 2^{10.89}$, $N' = 128 \cdot 2^{10.89}$ and the total complexity of the previous attack is $2^{3 \cdot 17.89} \cdot 2^6 \approx 2^{60}$ operations considering a resolution with a simple linearization.

The corresponding complexity for F-FCSR-DF8 is 2^{51} operations, for F-FCSR-SF1, that corresponds with 2^{39} operations and for F-FCSR-SF8 with 2^{39} operations.

Table 4. Second attack

Algorithm	Attack	Complexity	Data
F-FCSR-SF1 IV mode	exhaust. + alg	2^{39}	2^{11}
F-FCSR-DF1 IV mode	exhaust. + alg	2^{60}	2^{18}
F-FCSR-SF8 IV mode	exhaust. + alg	2^{39}	2^{8}
F-FCSR-DF8 IV mode	exhaust. + alg	2^{51}	2^{15}

We have implemented the first attack described (see section 3.2) on a small example to prove its relevance with $q = -112979$ and $d = 56490$. So, the main register M contains 16 binary cells $m(t)$ and the carry register $c(t)$ 8 cells. We consider here that the number of initial clocks is 3. We solve the obtained system using the implementation of the Buchberger algorithm provided by

magma 2.9. To simplify the resolution we consider here that the unknown filtering variables are directly the key bits. Solve the obtained system takes about fifteen minutes on a Pentium 4 and gives 328 possible solutions including the good one: K =0xdde5.

More, when we compute the number of exact monomials for the previous example, we obtain the following results:

Table 5. Experimental number of monomials for $n = 16$

nb of iterations	0	1	2	3	4	5	6	7
nb of monomials	16	17	32	86	250	766	2372	7148
Degree in $m_i(0)$	1	1	2	3	4	6	8	10

3.4 Could we apply this attack on F-FCSR-8 and on F-FCSR-H ?

Those attacks especially the second one could not be applied on the two versions proposed for the ECRYPT call for stream cipher primitives [17] (see [3]) due to a greater number of clocks before the first output generation.

To prevent the FCSR constructions using an IV mode from the algebraic attacks proposed in this paper, the required number of clocks t before generating an output must verify the following inequality $2^n > 2^t \cdot (\#\{I_t\})^{2.37}$ where 2.37 is the coefficient of the resolution of a linear system given in [6] and where $\#\{I_t\}$ is the cardinal of the set defined in Proposition 3. For example, if $n = 128$, the minimal number of initial clocks must be at least equal to 34. If $n = 160$ for a key length equal to 80 bits, this lower bound is equal to 20.

The new parameters chosen for F-FCSR-8 (F-FCSR-DF8 using 64 initial clocks instead of 6, the number of monomials given by Proposition 3 is then close to 2^{64}) and F-FCSR-H (the construction presented in Section 1.2 with 160 initial clocks and $n = 160$ for a key length equal to 80 bits, then the number of possible monomials is 2^{80}) described in the ECRYPT submission (see [3] for more details) verifies the previous conditions and prevent the new proposed FCSR constructions from the two attacks described in this paper that become more expansive than the exhaustive key search.

In the estimation of the number of monomials made here, we do not take into account the time required to compute all those algebraic equations (we do not evaluate the corresponding complexity) but experimentally, it seems that this complexity becomes greater than the resolution of the system itself for more than 10 iterations.

Conclusion

We present in this paper two algebraic attacks against the F-FCSR constructions proposed in [2] based on some bad choices in the stream-cipher parame-

ters but also on some particular algebraic properties of the FCSRs described here.

We do not contest the security level provided by the FCSR, we only claim that the security margin induced by the total construction proposed in [2] is not sufficient. The proposed parameters (size of the FCSR, number of clocks before the first output,...) must be enlarged. This is what have been done in the constructions submitted to the ECRYPT call for stream ciphers [3]. However, some other attacks could be applied on those two new versions as noticed by E. Jaulmes and F. Muller in [9].

References

1. F. Arnault and T.P. Berger. Design and properties of a new pseudo-random generator based on a filtered FCSR automaton. In *IEEE, Transactions on Computers*, 2005. To appear.
2. F. Arnault and T.P. Berger. F-FCSR: design of a new class of stream ciphers. In *Fast Software Encryption - FSE 2005*, volume 3557 of *Lecture Notes in Computer Science*, pages 83–97. Springer-Verlag, 2005.
3. F. Arnault, T.P. Berger, and C. Lauradoux. The FCSR: primitive specification and supporting documentation. ECRYPT - Network of Excellence in Cryptology, Call for stream Cipher Primitives 2005. `http://www.ecrypt.eu.org/stream/`.
4. G. Ars and J.-C. Faugère. An algebraic cryptanalysis of nonlinear filter generators using Gröbner bases. Research Report INRIA Lorraine, number 4739, 2003.
5. D. Coppersmith and S. Winograd. On the asymptotic complexity of matrix multiplication. *SIAM Journal on Computing*, 11(3):472–492, August 1982.
6. D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic programming. *Journal of Symbolic Computation*, 9(3):251–280, 1990.
7. N. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 177–194. Springer-Verlag, 2003.
8. N. Courtois and W. Meier. Algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 345–359. Springer-Verlag, 2003.
9. E. Jaulmes and F. Muller. Cryptanalysis of ecrypt candidates F-FCSR-8 and F-FCSR-H. ECRYPT Stream Cipher Project Report 2005/046, 2005. `http://www.ecrypt.eu.org/stream`.
10. E. Jaulmes and F. Muller. Cryptanalysis of the F-FSCR stream cipher family. In *Selected Areas in Cryptography - SAC 2005*, *Lecture Notes in Computer Science*. Springer-Verlag, 2005. To appear.
11. A. Klapper and M. Goresky. 2-adic shift registers. In *Fast Software Encryption - FSE'93*, volume 809 of *Lecture Notes in Computer Science*, pages 174–178. Springer-Verlag, 1993.
12. A. Klimov and A. Shamir. A new class of invertible mappings. In *CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 470–483. Springer-Verlag, 2002.
13. A. Klimov and A. Shamir. Cryptographic applications of T-functions. In *Selected Areas in Cryptography - SAC 2003*, volume 3006 of *Lecture Notes in Computer Science*, pages 248–261. Springer-Verlag, 2004.
14. A. Klimov and A. Shamir. New applications of T-functions in block ciphers and hash functions. In *Fast Software Encryption - FSE'05*, Lecture Notes in Computer Science, pages 19–32. Springer-Verlag, 2005. to appear.
15. N. Koblitz. p-adic numbers, p-adic analysis and zeta-functions. Springer-Verlag, 1997.

16. W. Meier, E. Pasalic, and C. Carlet. Algebraic attacks and decomposition of Boolean functions. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 474–491. Springer-Verlag, 2004.
17. Network of Excellence in Cryptology ECRYPT. Call for stream cipher primitives. `http://www.ecrypt.eu.org/stream/`.