A New Approach for FCSRs*

François Arnault[1], Thierry Berger[1], Cédric Lauradoux[2],
Marine Minier[3], and Benjamin Pousse[1]

[1] XLIM (UMR CNRS 6172), Université de Limoges
123 avenue Albert Thomas, F-87060 Limoges Cedex - France
`first name.name@xlim.fr`
[2] Information Security Group
UCL / INGI / GSI
2, Place Saint Barbe
B-1348 Louvain-la-Neuve - Belgium
`cedric.lauradoux@uclouvain.be`
[3] Lyon University - CITI Laboratory - INSA de Lyon
6, avenue des arts, 69621 Villeurbanne Cedex - France
`marine.minier@insa-lyon.fr`

Abstract. The Feedback with Carry Shift Registers (FCSRs) have been proposed as an alternative to Linear Feedback Shift Registers (LFSRs) for the design of stream ciphers. FCSRs have good statistical properties and they provide a built-in non-linearity. However, two attacks have shown that the current representations of FCSRs can introduce weaknesses in the cipher. We propose a new "ring" representation of FCSRs based upon matrix definition which generalizes the Galois and Fibonacci representations. Our approach preserves the statistical properties and circumvents the weaknesses of the Fibonacci and Galois representations. Moreover, the ring representation leads to automata with a quicker diffusion characteristic and better implementation results. As an application, we describe a new version of F-FCSR stream ciphers.

Keywords: Stream cipher, FCSRs, ℓ-sequence, ring FCSRs.

1 Introduction

The FCSRs have been proposed by Klapper and Goresky [1,2,3] as an alternative to LFSRs for the design of stream ciphers. FCSRs share many of the good properties of LFSRs: sequences with known period and good statistical properties. But unlike LFSRs, they provide an intrinsic resistance to algebraic and correlation attacks because of their quadratic feedback function. However, two recent results [4,5] have shown weaknesses in stream ciphers using either the Fibonacci or Galois FCSR. Hell and Johansson [5] have exploited the bias in the carries behaviour of a Galois FCSR to mount a very powerful attack against

* This work was partially supported by the french National Agency of Research: ANR-06-SETI-013.

M.J. Jacobson Jr., V. Rijmen, and R. Safavi-Naini (Eds.): SAC 2009, LNCS 5867, pp. 433–448, 2009.

the F-FCSR stream cipher [6,7]. Fisher *et al.* [4] have considered an equivalent of the F-FCSR stream cipher based upon a Fibonacci FCSR to study the linear behavior of the induced system.

We present a new approach for FCSRs, which we call the ring representation or ring FCSR. This representation is based on the adjacency matrix of the automaton graph. A ring FCSR can be viewed as a generalization of the Fibonacci and Galois representations. Similar structure has been widely studied for the LFSR case as in [8,9,10], and is a building block of the stream cipher Pomaranch where LFSRs are used [11]. However, we present here for the first time this structure in the FCSR case.

A Fibonacci FCSR, has a single feedback function which depends on multiple inputs. A Galois FCSR has multiple feedbacks which all share one common input. A ring FCSR can be viewed as a trade-off between the two extreme cases. It has several feedback functions with different inputs. An example of ring FCSR is shown in Fig. 1.

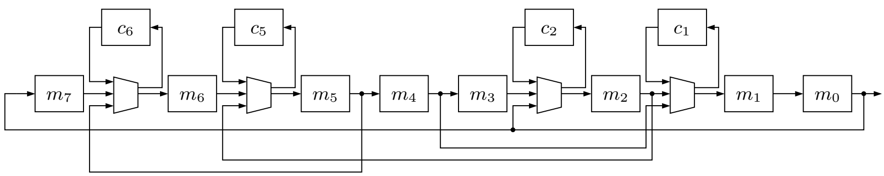

Fig. 1. An example of a ring FCSR ($q = -347$)

Ring FCSRs have many advantages, while preserving all the good and traditional properties of Galois/Fibonnacci FCSRs (known period, large entropy,...). The main one is that the attack of Hell and Johansson [5] does not work with Ring FCSR. Also, they have better diffusion properties. Moreover, ring representation allows fine tune in the implementation.

Section 2 gives an overview on FCSRs theory and classical representations. Section 3 presents ring FCSRs. We discuss implementation in Section 4 and a new version of F-FCSR is proposed in Section 5.

2 Theoretical Background

First, we will recall some basic properties of 2-adic integers. For a more theoretical approach the reader can refer to [1,2,12,13,14].

2.1 2-adic Numbers and Period

A 2-adic integer is formally a power series $s = \sum_{i=0}^{\infty} s_i 2^i$, $s_i \in \{0, 1\}$. This series always converges if we consider the 2-adic topology. The set of 2-adic integers is denoted by $\mathbb{Z}_2$. Addition and multiplication in $\mathbb{Z}_2$ can be performed by reporting the carries to the higher order terms, i.e. $2^n + 2^n = 2^{n+1}$ for all $n \in \mathbb{N}$. If there exists an integer N such that $s_n = 0$ for all $n \geq N$, then s is a positive integer. Every odd integer q has an inverse in $\mathbb{Z}_2$.

The following property gives a complete characterization of eventually periodic binary sequences in terms of 2-adic integers (see [13] for the proof).

Property 1. *Let $S = (s_n)_{n\in\mathbb{N}}$ be a binary sequence and let $s = \sum_{i=0}^{\infty} s_i 2^i$ be the corresponding 2-adic integer. The sequence S is eventually periodic if and only if there exist two numbers p and q in $\mathbb{Z}$, q odd, such that $s = p/q$.*

Moreover, S is strictly periodic if and only if $pq \leq 0$ and $|p| \leq |q|$. In this case, we have the relation $s_n = (p \cdot 2^{-n} \mod q) \mod 2$.

The period of S is the order of 2 modulo q, i.e., the smallest integer T such that $2^T \equiv 1 \pmod{q}$. The period satisfies $T \leq |q| - 1$. If q is prime, then T divides $|q| - 1$. If $T = |q| - 1$, the sequence S is called an ℓ-sequence. As detailed in [1,2,13,15], ℓ-sequences have many proved properties that could be compared to the ones of m-sequences: known period, good statistical properties, fast generation, etc. In summary, FCSRs have almost the same properties as LFSRs but they have a nonlinear structure.

2.2 Galois FCSRs

A Galois FCSR (as shown in Fig. 2) consists of an n-bit main register $M = (m_0, \ldots, m_{n-1})$ with some fixed feedback positions $d_0, \ldots, d_{n-1}$. All the feedbacks are controlled by the cell m_0, and $n-1$ binary carry cells $C = (c_0, \ldots, c_{n-2})$. At time t, an automaton in state (M, C) is updated in the following way:

1. Compute the sums $x_i = m_{i+1} + c_i d_i + m_0 d_i$ for all i such that $0 \leq i < n$ with $m_n = 0$ and $c_{n-1} = 0$ and where m_0 represents the feedback bit;
2. Update the state as follows: $m_i \leftarrow x_i \mod 2$ for all $i \in [0..n-1]$ and $c_i \leftarrow x_i$ div 2 for $0 \leq i < n$ for all $i \in [0..n-2]$.

The reader can refer to [13] for a complete description of Galois FCSRs and some properties. We recall however a very important one, found in [16].

Property 2. *Let $q = 1 - 2\sum_{i=0}^{n-1} d_i 2^i$ and $r_i = \sum_{t=0}^{\infty} m_i(t) 2^t$ (for $0 \leq i < n$); r_i is the 2-adic integer corresponding to the sequence observed in the i-th cell of the main register M. Then, for all $0 \leq i < n$, there exists $p_i \in \mathbb{Z}$ such that $r_i = p_i/q$.*

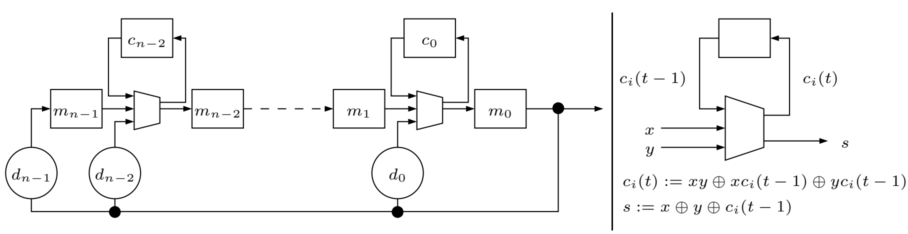

Fig. 2. A Galois FCSR and 2-bit adder with carry

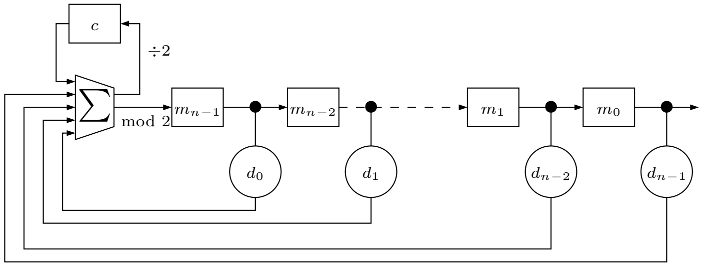

Fig. 3. A Fibonacci FCSR

In a Galois FCSR, a single cell controls all the feedbacks. As a consequence, there exist some correlations between the carries values and the feedback value. This fact is the basis of the attack presented in [5].

2.3 Fibonacci FCSRs

A Fibonacci FCSR (represented in Fig. 3) is composed of a main register $M = (m_0, \ldots, m_{n-1})$ with n binary cells. The binary feedback taps $(d_0, \ldots, d_{n-1})$ are associated to an additional carry register c of $w_H(d)$ binary cells, where $w_H(d)$ is the Hamming weight of $d = (1 + |q|)/2$.

An automaton in state (M, c) is updated in this way:

1. compute the sum $x = c + \sum_{i=0}^{n-1} m_i d_{n-1-i}$;
2. then, update the state: $M \leftarrow (m_1, \ldots, m_{n-1}, x \mod 2)$, $c \leftarrow x \text{ div } 2$.

As shown in [13], Property 2 also holds for Fibonacci FCSRs : the sequence observed in a cell m_i is a 2-adic integer.

The cell m_{n-1} is the only one with a non-linear behaviour in a Fibonacci FCSR. As shown in [4], an attack can be carried out if a linear filter is used with a Fibonacci FCSR.

3 A New Approach for FCSRs

Galois and Fibonacci FCSRs are two different automata with similar properties, as seen in the previous section. In a Galois FCSR, the first cell m_0 modifies $w_H(d)$ cells of the main register. In a Fibonacci FCSR, the cell m_{n-1} is modified by $w_H(d)$ cells of the main register. Ring representation of FCSRs is a trade-off between these extreme cases.

Definition 1. *A ring FCSR is an automaton composed of a main shift register of n binary cells $m = (m_0, \ldots, m_{n-1})$, and a carry register of n integer cells $c = (c_0, \ldots, c_{n-1})$. It is updated using the following relations:*

$$\begin{cases} m(t+1) = Tm(t) + c(t) \mod 2 \\ c(t+1) = Tm(t) + c(t) \quad \text{div}\, 2 \end{cases} \quad (1)$$

$$T_R = \begin{pmatrix} 0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,1\,0\,0\,0 \\ 1\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,1\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,1\,0\,1 \\ 1\,0\,0\,0\,0\,0\,0\,0 \end{pmatrix}$$

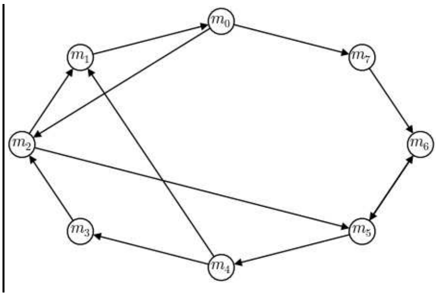

Fig. 4. Matrix and graph representation of FCSR presented in Fig.1

where T is a $n \times n$ matrix with coefficients 0 or 1 in $\mathbb{Z}$, called transition matrix, of this form:

$$\begin{pmatrix} * & 1 & & & & \\ & * & 1 & & (*) & \\ & & * & 1 & & \\ & & & \ddots & \ddots & \\ & & (*) & & * & 1 \\ 1 & & & & & * \end{pmatrix}$$

Note that $\div 2$ is the traditional expression: X $\operatorname{div} 2 = \frac{X - (X \mod 2)}{2}$.

Ring FCSRs differ from Fibonacci and Galois FCSRs in the fact that any cell can be used as a feedback for any other cell. A more convenient way to draw ring FCSRs is presented in Figure 4, which represents the same FCSR as the one in Figure 1.

3.1 Remarks on the Transition Matrix

According to Definition 1, we have the following property, where $t_{i,j}$ is the element at the i-th row and j-th column:

$$T = (t_{i,j})_{0 \le i,j < n} \text{ with } t_{i,j} = \begin{cases} 1 & \text{if cell } m_j \text{ is used to update cell } m_i, \\ 0 & \text{otherwise.} \end{cases}$$

As the main register of a ring FCSR is by definition a shift register, the over-diagonal of the transition matrix T is full of ones, i.e. for all $0 \le i < n$ we have $t_{i,i+1 \bmod n} = 1$. For example, the FCSR presented in Fig.1 has the following transition matrix T_R (and $q = -347$):

This notation agrees with the one proposed in [13]. In particular, Galois and Fibonacci FCSRs have respectively transition matrices T_G and T_F of the form:

$$T_G = \begin{pmatrix} d_0 & 1 & & & & \\ d_1 & 0 & 1 & & (0) & \\ d_2 & & 0 & 1 & & \\ \vdots & & & \ddots & \ddots & \\ d_{n-2} & & (0) & & 0 & 1 \\ 1 & & & & & 0 \end{pmatrix} \qquad T_F = \begin{pmatrix} 0 & 1 & & & & \\ & 0 & 1 & & (0) & \\ & & 0 & 1 & & \\ & (0) & & \ddots & \ddots & \\ & & & & 0 & 1 \\ 1 & d_{n-2} & \dots & d_2 & d_1 & d_0 \end{pmatrix}$$

3.2 Characterizing the Cells Content

Definition 1 introduces the transition matrix of a ring FCSR. We explain now how the value q can be computed from the transition matrix T.

Let $m_i(t)$ denote the content of the i-th cell of the main register at time t and $M_i(t)$ the series observed in this cell, from time t:

$$M_i(t) = \sum_{k \in \mathbb{N}} m_i(t+k)2^k.$$

From Equation 1, we derive the following relation

$$M(t+1) = TM(t) + c(t) \tag{2}$$

where $M(t) = (M_0(t), \cdots, M_{n-1}(t))$, and $c(t) = (c_0(t), \cdots, c_{n-1}(t))$ is the content of the carry register at time t.

The series $M_i(t)$ and the vector $M(t)$ play a fundamental role in our approach. We have the following important generalisation of Property 2.

Theorem 1. *The series $M_i(t)$ observed in the cells of the main register are 2-adic expansion of p_i/q with $p_i \in \mathbb{Z}$ and with $q = \det(I - 2T)$.*

Proof. According to the definition of $M_i(t)$ and to Definition 1, we have $M(t) = 2M(t+1) + m(t)$ where $m(t)$ is a binary vector of size n. Using Equation 2, we get:

$$(I - 2T) \cdot M(t) - 2 \cdot c(t) - m(t) = 0.$$

Considering the adjugate of $I - 2T$, we obtain:

$$\det(I - 2T) \cdot M(t) = Adj(I - 2T)(m(t) + 2 \cdot c(t)).$$

In this relation, the right member is a vector of integers $(p_0(t), \ldots, p_{n-1}(t))$. Dividing by $\det(I - 2T)$, we obtain $M_i(t) = p_i(t)/\det(I - 2T)$.

Lemma 1. *With the notation of Theorem 1, if $q = \det(I - 2T)$ is prime, and if the order of 2 in $\mathbb{Z}/q\mathbb{Z}$ is maximal, then each M_i is an ℓ-sequence.*

4 Implementation Properties

We detail in this section the new implementation characteristics of ring FCSRs. All this section applies also to LFSRs by replacing addition with carry with addition modulo 2.

Path/fan-out – The Galois FCSR is considered in many works [13,17,18] as the best representation for hardware implementation. It has a better critical path, i.e, a shorter longest path, than a Fibonacci FCSR. A drawback of the Galois representation is that the fan-out of the feedback cell m_0 is $w_H(d)$ with $d = (1 + |q|)/2$. At the opposite, the Fibonacci representation has a fan-out of 2. A ring FCSR allows the designer to tune both the critical path and the fan-out through the choice of the transition matrix:

Table 1. Comparison of the different representations

	Fibonacci	Galois	Ring
Path	$\lceil \log_2(w_H(d)) \rceil$	1	$\max(\lceil \log_2(w_H(a_i)) \rceil)$
Fan-out	2	$w_H(d)$	$\max(w_H(b_i))$
Cost ($\#_{adders}$)	$w_H(d) - 1$	$w_H(d) - 1$	$w_H(T) - n$

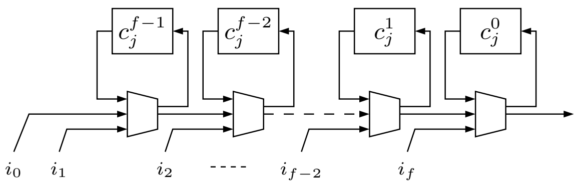

Fig. 5. A naive adder

– the critical path is given by the row a_i with the largest number of 1s;
– the fan-out is given by the column b_i with the largest number of 1s.

We compare in Table 1 the critical path, the fan-out and the cost of the different representations of an FCSR. We have expressed the critical path as the number of adders crossed. The choice of the adder has also an impact on the path of a ring FCSR. A naive adder (Fig. 5) composed of a serialisation of generic adder leads to a path of $max(w_H(a_i)) - 1$ adders. However, it is possible to exploit the commutativity to perform additions in parallel. This reduces the critical path to $max(\lceil \log_2(w_H(a_i)) \rceil)$ adders.

For each given q, it should be possible to find a transition matrix corresponding to a critical path with only one adder and a fan-out equal to 2. This is the case of the ring FCSR given in Fig. 1.

Cost – Ring FCSR have implementations which require fewer gates than Fibonacci/Galois equivalent ones. This possibility was first observed in [10] for LFSRs. However, the solution proposed in [10] is specific to LFSRs and cannot be applied systematically to FCSRs. The number $\#_{adders}$ of 2-bit adders required in the different representations of an n-bit FCSR is shown in Table 1. Ring representation is the only one that allows to find an implementation with less than $(w_H(d) - 1)$ 2-bit adders. For $q = -347$, a Galois or Fibonacci representation leads to $\#_{adders} = 4$. A ring representation with the following transition matrix T_R:

$$T_R = \begin{pmatrix} 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&1 \\ 0&1&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 1&0&1&0&0&0&0&0 \end{pmatrix}$$

leads to an implementation with $\#_{adders} = 3$, a fan-out of 2 and a critical path of 1 adder.

Side-channel attacks – It seems possible to work out an equivalent of the side-channel attack of Joux and Delaunay [18] on Galois FCSR using the results of Hell and Johansson [5]. Such an attack would exploit the power consumption to recover the feedback m_0 (because of the excessive fan-out of the feedback cell) and therefore how the carry cells are modified. As the ring FCSR has a reduced fan-out and uncorrelated carries, it is a better alternative to prevent side-channel attacks.

5 F-FCSR Based on Ring Representation

In this section, we propose a generic algorithm to construct F-FCSR stream ciphers based upon a ring FCSR with a linear filter. We give two particular examples which are F-FCSR-H v3 and F-FCSR-16 v3. Any designer using the proposed algorithm could generate its own stream cipher according to the following parameters:

– key length k and IV length v that will provide the corresponding size $n := k + v$ of the T matrix (usually $k = v$);
– the number u of bits output at each clock taken between 1 and $n/16$ to ensure a hard inversibility of the filter. Moreover for later design we require u to be a divisor of n;
– the number of willing feedbacks ℓ usually taken between $n/2-5$ and $n/2+5$ to ensure a sufficient non linear structure and a sufficiently weighted filter.

The algorithm is composed of 3 particular steps: the choice of the matrix T, the choice of the linear filter and the key/IV setup.

5.1 The Choice of the Matrix *T*

According to the remarks in Section 3, we pick a $n \times n$ random matrix T with the following requirements:

– the matrix must be composed of 0 and 1 and with a general weight equal to $n + \ell$;
– the over-diagonal must be full of 1 and $t_{n-1,0} = 1$ (to preserve the ring structure of the automaton);
– the number of ones for a given row or a given column must be at most two. This last condition allows a better diffusion by maximizing the number of cells reached by the feedbacks. It also provides uncorrelated carries and a fan-out bounded by 2.

For each picked matrix with the previous requirements, test if:

1. $\log_2(q) \geq n$; $\det(T) \neq 0$;
2. $q = \det(I - 2T)$ is prime; the order of 2 modulo q is $|q| - 1$.

The first condition ensures a non-degenerated matrix. The second ensures good statistical properties and a long period.

This matrix completely defines the ring FCSR. The diffusion speed (which is faster than in Galois/Fibonacci FCSRs) is related to the diameter d of the transition graph. This diameter is the maximal distance between two cells of the main register. In other words, d is the distance after which any cell of the main register have been influenced by any other cell through the feedbacks. It corresponds to the minimal number of clocks required to have all the cells of the main register influenced by any other cell. d should be small for better diffusion.

5.2 The Filter

As in the previous versions of F-FCSR [7], we use a linear filter to extract the keystream in order to break the 2-adic structure of the automaton. This also prevents linearization attacks over the set of 2-adic numbers. The filter includes the cells of the main register which receive a feedback to prevent correlation attacks. The periodic structure of the filter in the previous versions of F-FCSR has been exploited in [5] to speed up the linear part of the attack. We prefer now a non periodic structure:

– let $F = \{m_{f_0}, \cdots, m_{f_{\ell-1}}\}$ be the set of all the cells m_i that receive a feedback and indexed in this way: the row f_i of the matrix T has more than one 1 for $0 \leq i < \ell$, and $f_i < f_{i+1}$.
– The u bits of output are: $\forall\, 0 \leq i < u,\ z_i = \bigoplus_{j \equiv i \mod u} m_{f_j}$.

5.3 Key and IV Setup

As shown in [5], if at a given time, the FCSR is in a synchronized state (i.e. a state from which after a finite number of steps the automaton will return, i.e. a state belonging to the main cycle), adjacent states of the main cycle could be directly deduced using only multiplications over $\mathbb{Z}/q\mathbb{Z}$. Moreover, as shown in [16], a Galois FCSR is synchronized in at most $n + 4$ clocks, but in reality, few clocks are sufficient. So, to completely avoid the weakness of the key and IV setup used in [5], we prefer to maintain a non synchronized state during the key and IV setup. The new key and IV setup creates a transformation that is really hard to invert, in order to prevent a direct key recovery attack.

However, using a ring FCSR leads to a new problem: we can not ensure the entropy of the automaton. In the case of F-FCSR with Galois or Fibonacci structure, zeroing the content of the carry register prevents collisions (i.e. one point of the states graph with two preimages) and warrants a constant entropy. This particular property comes from the particular structure of the adjoint matrix $(I - 2T)^*$, which has successive powers of two in its first row in case of a Galois FSCR (in a Fibonacci FCSR, a similar property holds for the last row). In the ring case, no obvious structure exists in $(I - 2T)^*$. Note that in this case the collisions search becomes an instance of the subset sum problem, with a complexity equals to $2^{n/2}$ (if the carries are zeroes) or $2^{3n/2}$ (in the general case).

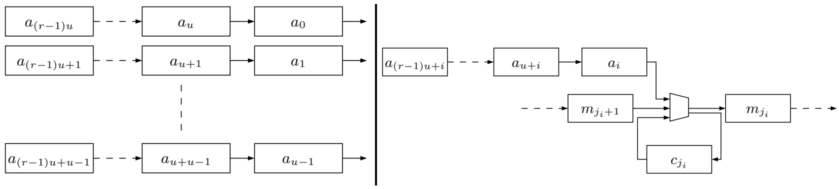

Fig. 6. Disposition of the cells $a_0, \ldots, a_{n-1}$ in u shift registers and connection of a shift register in position j_i

Thus, the new key and IV setup aims to stay on non-synchronized states as long as possible and to limit the entropy loss. We connect at u different places shift registers of length $r = n/u$ (this corresponds to adding n binary cells $a_0, \ldots, a_{n-1}$ at different places as shown in Fig. 6).

The u positions denoted by $J := \{j_0 < \cdots < j_{u-1}\}$ where the u shift registers are connected have been chosen such that, for all $0 \leq i < u$, no adder exists between cells m_{j_i+1} and m_{j_i} (i.e. $w_H(R_{j_i}) = 1$ where R_{j_i} is the j_i^{th} row of the matrix T). Each shift register is connected using a 2-bit adder with carry (as shown in Fig. 6). The content of cell m_{j_i} after transition depends on the values of m_{j_i+1}, a_i and of the carry cell c_{j_i}.

With these u shift registers inserted in the ring FCSR, the key and IV setup works as follows:

- Initialize $(a_0, \ldots, a_{n-1})$ with $(K \| 0^{n-k-v} \| IV)$, $M \leftarrow 0$, $C \leftarrow 0$.
- The FCSR is clocked r times. At each clock, the FCSR is filtered using F to produce a u bits vector $z_0, \ldots, z_{u-1}$ used to fill back $a_{(r-1)u}, \ldots, a_{(r-1)u+u-1}$: $a_{(r-1)u+i} \leftarrow z_i$ for $0 \leq i < u$.
- The FCSR is clocked $\max(r, d+4)$ times discarding the output.

The first step of the key and IV setup allows an initial diffusion of the key through the simple shift registers. The next r clocks helps a complete diffusion of the IV and of the key in the FCSR. The diffusion is complete at the end of the key and IV setup. If an attacker is able to recover the state just at the end of the key and IV setup, he won't be able to use this information to recover the key because of the occurence of non-synchronized states that are hard to inverse: for a given m_{k+1} bit value of the main register, the values c_k and m_k producing m_{k+1} are not unique and this leads to a combinatorial explosion when an attacker wants to recover a previous state.

As previously mentioned, this construction does not provide a bijection and behaves more like a random function. From this point, two attacks are essentially possible: direct collisions search and time memory data trade-off attack for collisions search built upon entropy loss. As mentioned before, direct collisions search has a cost of $2^{(n/2)}$ if the attacker is able to clear the carry bits. With the use of a ring FCSR that does not allow a direct control of the carry bits through the feedback bit, the probability to force to 0 the carry bits is about $2^{-\ell}$. Thus such an attack is more expensive than a key exhaustive search. In the other cases, the corresponding complexity is $2^{(3n/2)}$ preventing collisions search.

TMDTO attacks are possible if a sufficient quantity of entropy is lost. As studied in [19], considering that the key and IV setup are random function, the induced entropy loss is about 1 bit, so considering an initial entropy equal to n bits, the entropy after the key and IV setup is close to $n-1$ bits. Is it possible to exploit this entropy loss for a collisions search in a TMDTO attack? A well-known study case is the attack proposed in [20] by J. Hong and W.H. Kim against the stream cipher MICKEY. Even if this attack seems to work, A. Rock has shown in [19] that the query complexity in the initial states space could not be significantly reduced and that the attacks based on the problem of entropy loss are less efficient than expected especially regarding the query complexity. So, we conjecture, that our key and IV setup behaves as a random function, and that the induced entropy loss is not sufficient to mount a complete TMDTO attack for collisions search taking into account the query complexity.

5.4 F-FCSR-H v3 and F-FCSR-16 v3

The details of the two constructions, especially the corresponding T matrices, are given respectively in Appendix A and B. The respective parameters are the following ones:

- For F-FCSR-H v3: $k = 80$, $v = 80$, $\ell = 82$, $n = 160$, $u = 8$, $d = 24$;
- For F-FCSR-16 v3: $k = 128$, $v = 128$, $\ell = 130$, $n = 256$, $u = 16$, $d = 28$.

These two automata have been chosen with an additional property: $(|q| - 1)/2$ prime. This condition ensures maximal period for the output stream. However this condition is hard to fill. So we don't require this condition in the general case.

5.5 Resistance against Known Attacks

We do not discuss here resistance against traditional attacks such as correlation / fast correlation attacks, guess and determine attacks, algebraic attacks, etc. Some details about this can be found in [7]. Resistance against TMDTO attacks was considered in Section 5.3. We focus now on the two recent attacks [5] and [4] against FCSR and F-FCSR.

The attack presented in [5] against F-FCSR, which is based on a Galois FCSR, relies on the existence of correlations between the carries and the feedback values. More precisely, the control of the m_0 bit leads to the control of the feedback values. If the feedback can be forced to 0 during t consecutive clocks, the behavior of the stream cipher becomes linear, and its synthesis is possible by solving a really simple system. This linear behavior happens with a probability about 2^{-t} for a Galois FCSR. If instead a ring FCSR is used, this probability decreases to $2^{-t \cdot k}$ where k is the number of cells of the main register controlling a feedback. Thus, for k values corresponding to most ring FCSR, the linear behavior probability becomes so small that the cost of the corresponding attack becomes higher than an exhaustive search. Also the attack from [5] relies on situations where the carries remain constant during t consecutive clocks. We made an experiment with F-FCSR-H v3 to search for states for which carries does not change during transition. Looking over 2^{38} states, we found only 41 different states for which carries remains constant after one transition. We found none for which carries remains constant after two transitions.

In [4], the authors propose a linearization attack against a linearly filtered Fibonacci FCSR. This attack does not affect any version of F-FCSR. In a Fibonacci FCSR, the carries only influence one bit of the main register at each clock. Thus, if one could imagine to build a F-FCSR using a Fibonacci FCSR, such a generator would be subject to an attack where the control of the carries leads to the control of a part of the main register. Thus, we recommend to NOT use a Fibonacci FCSR in a linearly filtered stream cipher.

6 Conclusion and Future Work

In this paper, we have presented a new approach for FCSRs that unifies the two classical representations. We can obtain, with the ring representation, better diffusion characteristics and faster implementations. The recent attacks designed against F-FCSR are prevented, when using a ring FCSR, as shown in Section 5.

References

1. Klapper, A., Goresky, M.: 2-adic shift registers. In: Anderson, R. (ed.) FSE 1993. LNCS, vol. 809, pp. 174–178. Springer, Heidelberg (1994)
2. Klapper, A., Goresky, M.: Feedback shift registers, 2-adic span and combiners with memory. Journal of Cryptology 10(2), 111–147 (1997)
3. Klapper, A.: A survey of feedback with carry shift registers. In: Helleseth, T., Sarwate, D., Song, H.-Y., Yang, K. (eds.) SETA 2004. LNCS, vol. 3486, pp. 56–71. Springer, Heidelberg (2005)
4. Fischer, S., Meier, W., Stegemann, D.: Equivalent Representations of the F-FCSR Keystream Generator. In: ECRYPT Network of Excellence - SASC Workshop, pp. 87–94 (2008), `http://www.ecrypt.eu.org/stvl/sasc2008/`
5. Hell, M., Johansson, T.: Breaking the F-FCSR-H stream cipher in real time. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 557–569. Springer, Heidelberg (2008)
6. Arnault, F., Berger, T.P.: F-FCSR: Design of a new class of stream ciphers. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 83–97. Springer, Heidelberg (2005)
7. Arnault, F., Berger, T.P., Lauradoux, C.: Update on F-FCSR Stream Cipher. ECRYPT - Network of Excellence in Cryptology (Call for stream Cipher Primitives - Phase 2 2006) (2006), `http://www.ecrypt.eu.org/stream/`
8. Roggeman, Y.: Varying feedback shift registers. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 670–679. Springer, Heidelberg (1990)
9. Jansen, C.J., Helleseth, T., Kholosha, A.: Cascade jump controlled sequence generator and pomaranch stream cipher (version 2). eSTREAM, ECRYPT Stream Cipher Project, Report 2006/006 (2006), `http://www.ecrypt.eu.org/stream`
10. Mrugalski, G., Rajski, J., Tyszer, J.: Ring generators - new devices for embedded test applications. IEEE Trans. on CAD of Integrated Circuits and Systems 23(9), 1306–1320 (2004)

11. Jansen, C.J., Helleseth, T., Kholosha, A.: Pomaranch version 3. eSTREAM, ECRYPT Stream Cipher Project (2006), http://www.ecrypt.eu.org/stream
12. Koblitz, N.: p-adic numbers, p-adic analysis and Zeta-Functions. Springer, Heidelberg (1997)
13. Goresky, M., Klapper, A.: Fibonacci and Galois representations of feedback-with-carry shift registers. IEEE Transactions on Information Theory 48(11), 2826–2836 (2002)
14. Arnault, F., Berger, T.P.: Design and Properties of a New Pseudorandom Generator Based on a Filtered FCSR Automaton. IEEE Transaction on Computers 54(11), 1374–1383 (2005)
15. Lauradoux, C., Röck, A.: Parallel generation of ℓ-sequences. In: Golomb, S.W., Parker, M.G., Pott, A., Winterhof, A. (eds.) SETA 2008. LNCS, vol. 5203, pp. 299–312. Springer, Heidelberg (2008)
16. Arnault, F., Berger, T.P., Minier, M.: Some Results on FCSR Automata With Applications to the Security of FCSR-Based Pseudorandom Generators. IEEE Transactions on Information Theory 54(2), 836–840 (2008)
17. Goldberg, I., Wagner, D.: Architectural considerations for cryptanalytic hardware. Technical report, Secrets of Encryption Research, Wiretap Politics & Chip Design (1996)
18. Joux, A., Delaunay, P.: Galois LFSR, embedded devices and side channel weaknesses. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 436–451. Springer, Heidelberg (2006)
19. Röck, A.: Stream ciphers using a random update function: Study of the entropy of the inner state. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 258–275. Springer, Heidelberg (2008)
20. Hong, J., Kim, W.H.: TMD-Tradeoff and State Entropy Loss Considerations of Streamcipher MICKEY. In: Maitra, S., Veni Madhavan, C.E., Venkatesan, R. (eds.) INDOCRYPT 2005. LNCS, vol. 3797, pp. 169–182. Springer, Heidelberg (2005)

A Description of the Transition Matrix for F-FCSR-H v3

Input parameters: $k = 80$ (key length), $v = 80$ (IV length), $\ell = 82$ (number of feedbacks), $n = 160$ (size of T), $u = 8$ (number of output bits), $d = 24$ (diameter of the graph).

We give here the description of the transition matrix $T = (t_{i,j})_{0 \leq i,j < 160}$ (see Fig. 7 for graphic representations):

- For all $0 \leq i < 160$, $t_{i,i+1 \mod 160} = 1$;
- For all $(i, j) \in S$, $t_{i,j} = 1$, where $S = \{$ (1, 121); (2, 133); (4, 44); (5, 82); (9, 38); (11, 40); (12, 54); (14, 105); (15, 42); (16, 63); (18, 80); (19, 136); (20, 2); (21, 35); (23, 28); (25, 137); (28, 131); (31, 102); (36, 41); (39, 138); (40, 31); (42, 126); (44, 127); (45, 77); (46, 110); (47, 86); (48, 93); (49, 45); (51, 17); (54, 8); (56, 7); (57, 150); (59, 25); (62, 51); (63, 129); (65, 130); (67, 122); (73, 148); (75, 18); (77, 46); (79, 26); (80, 117); (81, 1); (84, 72); (86, 60); (89, 15); (90, 89); (91, 73); (93, 12); (94, 84); (102, 141); (104, 142); (107, 71); (108, 152); (112, 92); (113, 83); (115, 23); (116, 32); (118, 50); (119, 43); (121, 34); (124, 13); (125, 74); (127, 149); (128, 90); (129, 57); (130, 103); (131, 134); (132, 155); (134, 98); (139, 24);

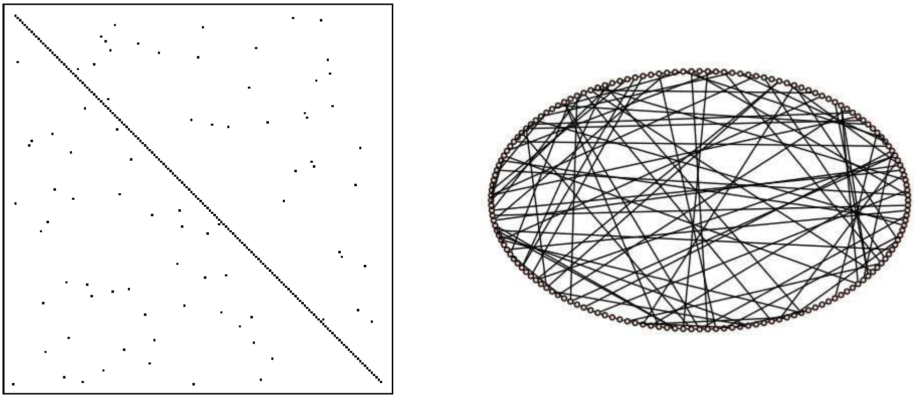

Fig. 7. Matrix representation and graph representation of the matrix T chosen for F-FCSR-H v3

(140, 61); (141, 104); (144, 48); (145, 14); (148, 112); (150, 59); (153, 39); (156, 22); (157, 107); (158, 30); (159, 78) };

– Otherwise, $t_{i,j} = 0$.

– The corresponding q value is (in decimal notation):

$$q = 1741618736723237862812353996255699689552526450883$$

– The set J (for the first part of the Key/IV setup) is:

$$J = \{3, 22, 43, 64, 83, 103, 123, 143\}$$

– The 8 subfilters $F_0, \cdots, F_7$ are given by:

$$\begin{aligned} F_0 &= \{1, 15, 28, 46, 59, 79, 93, 115, 128, 141, 158\} \\ F_1 &= \{2, 16, 31, 47, 62, 80, 94, 116, 129, 144, 159\} \\ F_2 &= \{4, 18, 36, 48, 63, 81, 102, 118, 130, 145\} \\ F_3 &= \{5, 19, 39, 49, 65, 84, 104, 119, 131, 148\} \\ F_4 &= \{9, 20, 40, 51, 67, 86, 107, 121, 132, 150\} \\ F_5 &= \{11, 21, 42, 54, 73, 89, 108, 124, 134, 153\} \\ F_6 &= \{12, 23, 44, 56, 75, 90, 112, 125, 139, 156\} \\ F_7 &= \{14, 25, 45, 57, 77, 91, 113, 127, 140, 157\} \end{aligned}$$

B Description of the Transition Matrix for F-FCSR-16 v3

Input parameters: $k = 128$, $v = 128$, $\ell = 130$, $n = 256$, $u = 16$, $d = 28$.

We give here a description of the transition matrix $T = (t_{i,j})_{0 \le i,j < 256}$ (see Fig. 8 for graphic representations):

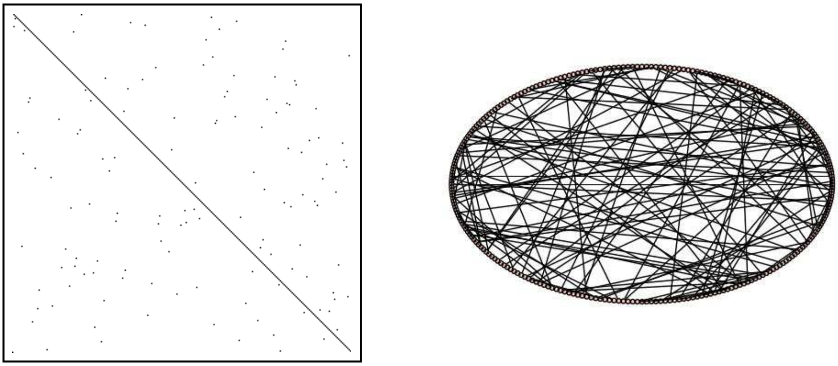

Fig. 8. Matrix representation and graph representation of the matrix T chosen for F-FCSR-16 v3

- For all $0 \leq i < 256$, $t_{i,i+1 \mod 256} = 1$;
- For all $(i, j) \in S$, $t_{i,j} = 1$, where $S = \{$ (0, 52); (2, 150); (3, 2); (5, 169); (6, 89); (8, 100); (9, 1); (11, 156); (12, 9); (13, 46); (19, 146); (20, 206); (26, 204); (31, 254); (32, 151); (38, 144); (40, 108); (46, 167); (47, 198); (48, 70); (49, 98); (50, 213); (53, 214); (56, 87); (57, 55); (58, 162); (62, 160); (63, 13); (64, 192); (65, 59); (66, 12); (67, 207); (68, 209); (71, 229); (73, 84); (74, 199); (77, 168); (78, 122); (79, 35); (80, 154); (82, 153); (85, 188); (87, 51); (89, 4); (90, 49); (93, 231); (95, 224); (97, 249); (101, 208); (102, 120); (104, 218); (105, 8); (108, 77); (109, 68); (110, 250); (113, 237); (115, 252); (116, 17); (118, 73); (119, 182); (123, 29); (124, 234); (127, 138); (132, 190); (134, 244); (136, 219); (141, 228); (142, 205); (143, 58); (144, 230); (145, 210); (146, 44); (147, 137); (148, 130); (150, 79); (152, 111); (153, 172); (154, 141); (156, 78); (157, 131); (158, 110); (159, 127); (170, 189); (171, 112); (174, 217); (175, 7); (176, 187); (177, 40); (179, 118); (181, 195); (184, 48); (186, 64); (189, 246); (190, 47); (191, 37); (192, 211); (193, 85); (194, 181); (195, 61); (196, 54); (198, 222); (199, 83); (203, 105); (204, 201); (205, 43); (206, 139); (208, 20); (210, 242); (211, 124); (213, 253); (215, 243); (216, 69); (218, 176); (220, 30); (222, 19); (223, 232); (224, 239); (225, 220); (227, 102); (231, 185); (232, 15); (234, 152); (236, 62); (238, 245); (242, 197); (245, 235); (246, 171); (247, 67); (253, 26); (254, 202) $\}$;
- Otherwise, $t_{i,j} = 0$.

- The corresponding q value is (in hexadecimal notation):

$$q = (B085834B6BFAE1541C54F7D84F42084C \\ B0568496DDD0FEA5E99AA79C022023241)$$

- The set J (for the first part of the Key/IV setup) is:

$$J = \{10, 27, 43, 59, 75, 91, 107, 122, 139, 155, 172, 187, 202, 219, 235, 251\}$$

– The 16 subfilters $F_0, \cdots, F_{15}$ are given by:

$$
\begin{aligned}
F_0 &= \{0, 40, 68, 101, 134, 158, 193, 218, 253\}\\
F_1 &= \{2, 46, 71, 102, 136, 159, 194, 220, 254\}\\
F_2 &= \{3, 47, 73, 104, 141, 170, 195, 222\}\\
F_3 &= \{5, 48, 74, 105, 142, 171, 196, 223\}\\
F_4 &= \{6, 49, 77, 108, 143, 174, 198, 224\}\\
F_5 &= \{8, 50, 78, 109, 144, 175, 199, 225\}\\
F_6 &= \{9, 53, 79, 110, 145, 176, 203, 227\}\\
F_7 &= \{11, 56, 80, 113, 146, 177, 204, 231\}\\
F_8 &= \{12, 57, 82, 115, 147, 179, 205, 232\}\\
F_9 &= \{13, 58, 85, 116, 148, 181, 206, 234\}\\
F_{10} &= \{19, 62, 87, 118, 150, 184, 208, 236\}\\
F_{11} &= \{20, 63, 89, 119, 152, 186, 210, 238\}\\
F_{12} &= \{26, 64, 90, 123, 153, 189, 211, 242\}\\
F_{13} &= \{31, 65, 93, 124, 154, 190, 213, 245\}\\
F_{14} &= \{32, 66, 95, 127, 156, 191, 215, 246\}\\
F_{15} &= \{38, 67, 97, 132, 157, 192, 216, 247\}
\end{aligned}
$$