Construction of Transition Matrices for Binary FCSRs

Zhiqiang Lin[1], Dingyi Pei[2] and Dongdai Lin[1]

1 the State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences,
Beijing 100093, China
2 College of Mathematics and Information Science, Guangzhou University,
Guangzhou 510006, China
linzhiqiang@iie.ac.cn; ddlin@iie.ac.cn; gztcdpei@scut.edu.cn

Abstract. Stream ciphers based on Linear Feedback Shift Registers (LFSRs) have faced algebraic attacks. To avoid this kind of attacks, Feedback with Carry Shift Registers (FCSRs) have been proposed as an alternative. In order to eliminate a so-called LFSRization weakness, FCSRs have been implemented using ring representation instead of the Galois one. A ring FCSR is determined by its transition matrix A. Its connection integer, which is related to the properties of the output sequences, is $q = \det(I - 2A)$. In this paper, we show how to calculate the determinant $\det(I - 2A)$ of transition matrices with a critical path of length 1 and fan-out 2. Moreover, we propose algorithms to construct such transition matrices (binary case) based on searching target connection integers.

Keywords: stream cipher, l-sequences, 2-adic ring, FCSRs, transition matrix

1 Introduction

Linear Feedback Shift Registers (LFSRs) are widely used in information theory, coding theory and cryptography. However, for cryptographic applications their linear structure has the drawback that it may help algebraic attacks. Klapper and Goresky have introduced Feedback with Carry Shift Registers (FCSRs) as an alternative to LFSRs ([10],[11],[12]). FCSRs share many of the good properties of LFSRs, such as proven large period and good statistical properties, but also is possible to recover the structure of a sequence as for LFSRs ([13]). The main difference is the fact that the elementary additions in LFSRs are additions modulo 2, but in FCSRs are addition of integers with propagation of carries. This makes FCSRs to have quadratic transition functions and provide an intrinsic nonlinearity.

FCSRs have two traditional representations: Fibonacci or Galois representations ([7]). In Fibonacci mode all the feedback bits influence a single cell, while in the Galois mode a single feedback bit influences all the carry cells. The Galois

representation is more efficient than the Fibonacci one for hardware implementations, because its feedback computations are performed in parallel. By using a linear filter on the cells of a Galois FCSR automaton, Arnault et al have proposed family of hardware stream ciphers: F-FCSR ([1]). They are resistant to many regular attacks. Furthermore, a software version of FCSR-based stream cipher: the X-FCSR family have been proposed by using word-ring FCSRs ([5]).

However, in Galois mode due to the dependency between the binary carries and the single feedback bit, it is possible to maintain linear behavior for several clock cycles in its main register. Hell and Johansson have exploited this LFSRization of FCSR to mount a very powerful attack against all the version of stream ciphers based on Galois FCSR ([9], [18]).

Arnault et al have responded to this attack by introducing a new FCSR representation called ring FCSR ([2]). In this new mode, any binary cell can be used as a feedback bit for any other cells in the main register. The LFSRization behavior seems to have disappeared when this new mode is used.

1.1 Ring FCSR

Definition 1. *A ring FCSR is composed of a main shift register of n binary cells $m = (m_0, m_1, \ldots, m_{n-1})^T$ (T indicates the column vector.) and a carry register of n integer cells $c = (c_0, c_1, \ldots, c_{n-1})^T$. It is updated using the following relations:*

$$\begin{cases} m(t+1) = Am(t) + c(t) \ (mod\ 2) \\ c(t+1) = Am(t) + c(t) \ (div\ 2) \ (X \ div\ 2 = \dfrac{X - (X \ (mod\ 2))}{2}), \end{cases}$$

where $A = (a_{l,d})_{1 \le l,d \le n}$ is an $n \times n$ matrix with entries in $\{0, 1\}$, called transition matrix, of this form:

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

It is easy to see that ring FCSRs are totally determined by their transition matrices through Definition 1. As the main register of an FCSR is by definition a shif register, the over-diagonal of the transition matrix A is full of ones, i.e. for all $1 \le l \le n$ we have $a_{l,l+1 \ (\mathrm{mod}\ n)} = 1$. If $a_{l,d}$ is 1, then m_{d-1} is used to feedback to m_{l-1}.

An output sequence of an FCSR can be seen as a power series in the ring of 2-adic integers. The maximal period FCSR-sequences, called l-sequences, are similar to m-sequences and also have good properties ([13]).

Theorem 1. *([2]) Let $\mathcal{F}$ be a ring FCSR with transition matrix A. Let $q = det(I - 2A)$ be an integer, where I is an $n \times n$ identity matrix. q is called the connection integer of $\mathcal{F}$. Then an output sequence of $\mathcal{F}$ is an l-sequence with period $|q| - 1$ if and only if $|q|$ is prime and 2 is a primitive root of $|q|$.*

A widely used approach to construct stream ciphers is to combine an FCSR automaton with a filtering function (such as F-FCSRs). The filtering function extracts keystream bits from the states of the main register. Hence the period of the keystream is always a factor of $|q| - 1$. In order to guarantee the period as large as possible, we usually assume that $(|q| - 1)/2$ is a prime, i.e. $(|q| - 1)/2$ is a Sophie-Germain prime and $|q|$ is called its associated safe prime. Therefore, for the design of stream ciphers, $|q|$ must satisfy the following conditions:

C_1 $|q|$ is a prime with 2 as its primitive root.
C_2 $(|q| - 1)/2$ is also a prime.

Such $|q|$ is called a safe prime.

Recently, another attack on FCSRs based stream ciphers is presented in ([20]). It is a distinguishing attack based on an inherent linear bias of l-sequences ([19]). Wang et al have shown the attack on the F-FCSR-H v3, an F-FCSR that uses a ring FCSR, with an online time complexity of only $2^{37.2}$. The off-line time complexity (for finding a linear relation) is $2^{56.2}$. It breaches the exhaustive search complexity limit. However, this attack does not indicate that FCSRs are useless but limits the usefulness of FCSRs for cryptography, such as not to use a linear filter. So when constructing ring FCSRs, we do not take into account this attack.

In ([15]) a class of ring FCSRs called "defective FCSRs" are found not to remove the LFSRization behavior. A feature of them is that feedbacks are all from a few blocks of serial simple shifting cells. So we should avoid to construct "defective FCSRs".

1.2 Design criteria of hardware ciphers

Ring FCSRs are suitable for hardware implementations ([2, 3]). The cells of the main register can be built as a cascade of flip flops. The feedbacks are implemented by adders-with-carry (can be used to sum up two 2-adic integers). Good hardware designs are required to

- Critical path length: The critical path length is the maximum number of adders the signal have to pass through. If this number is low, the circuit can be clocked at a higher rate.
- Fan-out: A given signal should drive minimum adder number.
- Cost: The number of adders must be as small as possible to lower consumption. However, more adders may need to ensure the complexity when using for cryptography.

The optimal criteria is a critical path of length 1 and fan-out 2. A transition matrix A with these criteria can be described by the following conditions:

C_3 A is an $n \times n$ matrix with entries in $\{0, 1\}$, and the over-diagonal entries $a_{l,l+1 \pmod n}$ $(1 \leq l \leq n)$ must be 1.

C_4 The number of nonzero entries for a given row or a given column must be at most 2.

Another design criterion is called diffusion delay. This criterion represents the diffusion capacity of an FCSR. This parameter is important for cryptographic purpose where small differences in keys or in messages are required to have a large impact. Moreover, a lower diffusion delay would speed up the so-called initialization phase in stream cipher designs.

Definition 2. *([4]) Let $\mathcal{F}$ be a ring FCSR with transition matrix A. Denote by D the digraph consists of n points called vertices defined by A, i.e., if $a_{l,d} \neq 0$ then there exists a directed edge from vertex d and to vertex l. The diffusion delay is equal to the diameter of D.*

Considering the Fibonacci and Galois representations, the associated diffusion delay is $n-1$ because the cells on each side m_0 and m_{n-1} require $n-1$ clocks to mix together. The ring representation allows to achieve a better diffusion delay.

1.3 Construction of ring FCSRs

A ring FCSR suitable for hardware implementations should satisfy the conditions from C_1 to C_4. Normally the number of its feedbacks is about $f = n/2$ for a tradeoff between consumption and safety.

In ([3]), Arnault et al have proposed an algorithm to construct ring FCSRs. The algorithm has two loops. In its first loop, the algorithm randomly pick a matrix A, with $f-1$ feedbacks, satisfying the conditions C_3 and C_4 at first. Then it computes $\det(I-2A)$ and the cofactor matrix of $(I-2A)$. In its second loop, the last feedback is also placed to satisfy the condition C_4. Then using a property of determinant one could obtain roughly $n^2 - n \cdot f$ connection integers and test if one of them satisfies also the condition C_1 (C_2 is not considered there!). The complexities of the computations of the determinant and the cofactor matrix are both $O(n^3)$. In addition, since there may be some equal cofactors in the cofactor matrix, the number of connection integers for testing in the second loop may be less than $n^2 - n \cdot f$ in one iteration. Hence, we need more efficient algorithms.

([14],[17]) have presented algorithms to construct ternary ring FCSRs (a kind of extensional ring FCSRs also suitable for hardware implementations) satisfying the conditions C_3 and C_4 with connection integers specified. The authors have given an affirmative answer to the following conjecture: for any given connection integer there exist ternary transition matrices with a critical path of length 1 and fan-out 2. This construction is based on a correspondence between the so-called non-adjacent form (NAF) of integers and the matrices $(I-2A)$. However, this method seems hard to graft onto the binary case.

In this paper, we show how to calculate the determinant $\det(I-2A)$, where A is a transition matrix with a critical path of length 1 and fan-out 2. According

to this calculation, we propose a method of constructing binary ring FCSRs. We first construct a specific initial matrix (satisfying the conditions C_3 and C_4). Then based on its connection integer and another parameter, we seek a target integer (satisfying the conditions C_1 and C_2) and immediately get an expected binary ring FCSR by this integer. In the search part, this method only needs to do some additions and simple tests, thus efficient.

This paper is organized as follows. In Section 2 we show how to calculate the determinant $\det(I-2A)$ and propose a generalized algorithm for constructing ternary ring FCSRs with given connection integers. Section 3 shows the major results, algorithms to construct binary ring FCSRs. Finally, this paper is concluded in Section 4.

2 Calculation of the determinant $\det(I-2A)$

2.1 Calculation method

Let $\mathcal{F}$ be a ring FCSR with a critical path of length 1 and fan-out 2, and let A be its transition matrix. Then A satisfies the conditions C_3 and C_4. Suppose there is no binary cell feedback to itself, that is $a_{l,l}=0$ for all $1\le l\le n$, and suppose there are f adders ($f\le n$): $a_{l_1,d_1}=1, a_{l_2,d_2}=1,\ldots,a_{l_f,d_f}=1$ ($d_i\neq l_i$ and $d_i\neq l_i+1$ for $1\le i\le f$) in A. Let $1\le l_1<l_2<\cdots<l_f\le n$ without loss of generality.

Denote the matrix $B=I-2A$ and call it connection matrix. Let $B=(b_{l,d})_{1\le l,d\le n}$. Then we have

- $b_{l,l}=1$ $(1\le l\le n)$ and $b_{l,l+1 \pmod n}=-2$ $(1\le l\le n)$. We call them diagonal entries and shifts respectively;
- $b_{l_1,d_1}=-2, b_{l_2,d_2}=-2,\ldots,b_{l_f,d_f}=-2$, called feedbacks;
- $b_{l,d}=0$ for others.

The determinant of B can be calculated as follows:

Theorem 2. *Let B be a connection matrix. The feedbacks in B are: $b_{l_1,d_1}=-2, b_{l_2,d_2}=-2,\ldots,b_{l_f,d_f}=-2$. Then we have*

$$det(B)=-2^n+1+\sum_{(i_1,i_2,\ldots,i_t)\subseteq(1,2,\ldots,f)} S_{i_1,i_2,\ldots,i_t}, \tag{1}$$

where $(i_1<i_2<\ldots<i_t)$ runs over all subsets of $(1,2,\ldots,f)$ and

$$S_{i_1,i_2,\ldots,i_t}=\begin{cases}(-1)^{\epsilon(d_{i_1},d_{i_2},\ldots,d_{i_t})+t}\cdot 2^{t+\sum_{r=1}^{t}(l_{i_r}-d'_{i_r})} & \text{if } d'_{i_1}\le l_{i_1}<d'_{i_2}\le l_{i_2}<\cdots<d'_{i_t}\le l_{i_t}\\ (-1)^{\epsilon(d_{i_1},d_{i_2},\ldots,d_{i_t})-1}\cdot 2^{n+t+\sum_{r=1}^{t}(l_{i_r}-d'_{i_r})} & \text{if } l_{i_1}<d'_{i_1}\le l_{i_2}<d'_{i_2}\le\cdots\le l_{i_t}<d'_{i_t}\\ 0 & \text{otherwise,}\end{cases}$$

where $\epsilon(\sigma)$ is the inverse number of permutation σ and $(d'_{i_1},d'_{i_2},\ldots,d'_{i_t})$ is the sequential permutation of $(d_{i_1},d_{i_2},\ldots,d_{i_t})$. For convenience, we call the two nonzero cases of $S_{i_1,i_2,\ldots,i_t}$ "Case I" and "Case II" respectively.

Proof. We have

$$\det(B) = \sum_{\sigma=(j_1,j_2,\ldots,j_n)} (-1)^{\epsilon(\sigma)} \cdot b_{1,j_1} b_{2,j_2} \cdots b_{n,j_n}, \tag{2}$$

where $\sigma = (j_1, j_2, \ldots, j_n)$ runs over all possible permutations of $(1, 2, \ldots, n)$.

A nonzero term in (2) only consists of diagonal entries, shifts and feedbacks. It is easy to see that there are only two nonzero terms in (2) without any feedback: $(-1)^{\epsilon(1,2,\ldots,n)} \cdot b_{1,1} b_{2,2} \cdots b_{n,n} = 1$ and $(-1)^{\epsilon(2,3,\ldots,n,1)} \cdot b_{1,2} b_{2,3} \cdots b_{n,1} = -2^n$.

Suppose in a nonzero term $(-1)^{\epsilon(\sigma)} \cdot b_{1,j_1} b_{2,j_2} \cdots b_{n,j_n}$ there are t $(1 \le t \le f)$ feedbacks: $b_{l_{i_1},d_{i_1}}, \ldots, b_{l_{i_t},d_{i_t}}$, where $(i_1 < i_2 < \ldots < i_t)$ is a subset of $(1, 2, \ldots, f)$. Then except these feedbacks, the left $n - t$ entries can only be either diagonal entries or shifts. Suppose $(d'_{i_1}, d'_{i_2}, \ldots, d'_{i_t})$ is the sequential permutation of $(d_{i_1}, d_{i_2}, \ldots, d_{i_t})$. We have:

Case I: If $d'_{i_1} \le l_{i_1} < d'_{i_2} \le l_{i_2} < \cdots < d'_{i_t} \le l_{i_t}$. Since $j_{l_{i_r}} = d_{i_r}$ for $1 \le r \le t$, $j_{d_{i_r}}$ can only be $d_{i_r} + 1$. It implies that $j_l = l + 1$ for $l = d'_{i_r}, d'_{i_r} + 1, \ldots, l_{i_r} - 1$ successively. We also have $j_{d'_{i_r}-1} = d'_{i_r} - 1$. It implies that $j_l = l$ for $l = d'_{i_r} - 1, d'_{i_r} - 2, \ldots, l_{i_{r-1}} + 1$ $(2 \le r \le t)$ successively and for $l = 1, 2, d'_{i_1} - 1, l_{i_t} + 1, \ldots, n$. So this term is only one possibility:

$$j_l = \begin{cases} d_{i_r} & l = l_{i_r} \ (1 \le r \le t) \text{ (feedbacks)} \\ l + 1 & l = d'_{i_r}, d'_{i_r} + 1, \ldots, l_{i_r} - 1 \ (1 \le r \le t) \text{ (shifts)} \\ l & \text{otherwise (diagonal entries).} \end{cases}$$

To compute the inverse number of σ, we first sort $(d_{i_1}, d_{i_2}, \ldots, d_{i_t})$, then sort $(d'_{i_r} + 1, d'_{i_r} + 2, \ldots, l_{i_r}, d'_{i_r})$ $(1 \le r \le t)$ respectively. So we have

$$\begin{aligned} \epsilon(\sigma) &= \epsilon(d_{i_1}, d_{i_2}, \ldots, d_{i_t}) + \sum_{r=1}^{t} \epsilon(d'_{i_r} + 1, \ldots, l_{i_r}, d'_{i_r}) \\ &= \epsilon(d_{i_1}, d_{i_2}, \ldots, d_{i_t}) + \sum_{r=1}^{t} (l_{i_r} - d'_{i_r}) \end{aligned}$$

Hence

$$\begin{aligned} & (-1)^{\epsilon(\sigma)} \cdot b_{1,j_1} b_{2,j_2} \cdots b_{n,j_n} \\ &= (-1)^{\epsilon(d_{i_1},d_{i_2},\ldots,d_{i_t}) + \sum_{r=1}^{t} (l_{i_r} - d'_{i_r})} \cdot (-2)^{t + \sum_{r=1}^{t} (l_{i_r} - d'_{i_r})} \\ &= (-1)^{\epsilon(d_{i_1},d_{i_2},\ldots,d_{i_t})+t} \cdot 2^{t + \sum_{r=1}^{t} (l_{i_r} - d'_{i_r})}. \end{aligned}$$

Case II: If $l_{i_1} < d'_{i_1} \le l_{i_2} < d'_{i_2} \le \cdots \le l_{i_t} < d'_{i_t}$, similar to Case I, the nonzero term is only one possibility:

$$j_l = \begin{cases} d_{i_r} & l = l_{i_r} \ (1 \le r \le t) \text{ (feedbacks)} \\ l + 1 & l = d'_{i_r}, d'_{i_r} + 1, \ldots, l_{i_{r+1}} - 1 \ (1 \le r \le t-1) \text{ and } l = 1, 2, \ldots, l_{i_1} - 1, d'_{i_t}, d'_{i_t} + 1, \ldots, n \text{ (shifts)} \\ l & \text{otherwise (diagonal entries).} \end{cases}$$

To compute the inverse number of σ, we first sort $(d_{i_1}, d_{i_2}, \ldots, d_{i_t})$ and reorder it as $(d'_{i_t}, d'_{i_1}, d'_{i_2}, \ldots, d'_{i_{t-1}})$, then sort $(d'_{i_r} + 1, d'_{i_r} + 2, \ldots, l_{i_{r+1}}, d'_{i_r})$

$(1 \le r \le t-1)$ and $(2, 3, \ldots, l_{i_1}, d'_{i_1}, d'_{i_t}+1, d'_{i_t}+2, \ldots, n, 1)$ respectively. So we have

$$\begin{aligned}\epsilon(\sigma) &= \epsilon(d_{i_1}, d_{i_2}, \ldots, d_{i_t}) + t - 1 + \epsilon(2, 3, \ldots, l_{i_1}, d'_{i_t}, d'_{i_t}+1, d'_{i_t}+2, \ldots, n, 1) + \sum_{r=1}^{t-1} \epsilon(d'_{i_r}+1, \ldots, l_{i_{r+1}}, d'_{i_r}) \\ &= \epsilon(d_{i_1}, d_{i_2}, \ldots, d_{i_t}) + t - 1 + n - d'_{i_t} + l_{i_1} + \sum_{r=1}^{t-1} (l_{i_{r+1}} - d'_{i_r}) \\ &= \epsilon(d_{i_1}, d_{i_2}, \ldots, d_{i_t}) + t - 1 + n + \sum_{r=1}^{t} (l_{i_r} - d'_{i_r}).\end{aligned}$$

Hence

$$\begin{aligned}&(-1)^{\epsilon(\sigma)} \cdot b_{1,j_1} b_{2,j_2} \cdots b_{n,j_n} \\ &= (-1)^{\epsilon(d_{i_1}, d_{i_2}, \ldots, d_{i_t}) + t - 1 + n + \sum_{r=1}^{t} (l_{i_r} - d'_{i_r})} \cdot (-2)^{t + n - d'_{i_t} + l_{i_1} + \sum_{r=1}^{t-1} (l_{i_{r+1}} - d'_{i_r})} \\ &= (-1)^{\epsilon(d_{i_1}, d_{i_2}, \ldots, d_{i_t}) - 1} \cdot 2^{t + n + \sum_{r=1}^{t} (l_{i_r} - d'_{i_r})}.\end{aligned}$$

Otherwise: If l_{i_r} and d'_{i_r} are not interleaving, then there exists k $(1 \le k \le t-1)$ such that no l_{i_r} $(1 \le r \le t)$ satisfies that $d'_{i_k} \le l_{i_r} \le d'_{i_{k+1}}$, or no l_{i_r} $(1 \le r \le t)$ satisfies that $1 \le l_{i_r} \le d'_{i_1}$ or $d'_{i_t} \le l_{i_r} \le n$. In the first situation, it implies that $j_l = l + 1$ for $l = d'_{i_k}, d'_{i_k}+1, \ldots, d'_{i_{k+1}}$ successively. So $j_{d'_{i_{k+1}-1}} = d'_{i_{k+1}}$. It is a contradiction. In the second situation, there exists a similar contradiction.

Therefore, a nonzero term in (2) can only be two cases: Case I and Case II. □

According to (1), $\det(B)$ can be calculated by the coordinates of the feedbacks. If the feedbacks are randomly selected we have to compute $f!$ terms and their sum. However, $f!$ is too big even if f is not big. For example, $f = 64$ and $64!$ is very large. In order to reduce computational effort, we will set up some qualifications to the selection of feedbacks. A simplest case is as follows:

Proposition 1. *Let B be a connection matrix with feedbacks $b_{l_1,d_1} = -2, b_{l_2,d_2} = -2, \ldots, b_{l_f,d_f} = -2$, where $d_i \ne l_i$ and $d_i \ne l_i + 1$ for $1 \le i \le f$. If there exist integers $1 \le u < v \le n$ that $l_i \in \{u, u+1, \ldots, v\}$ and $d_i \in (\{1, 2, \ldots, n\}/\{u, u+1, \ldots, v\})$ for all $1 \le i \le f$, or $d_i \in \{u, u+1, \ldots, v\}$ and $l_i \in (\{1, 2, \ldots, n\}/\{u, u+1, \ldots, v\})$ for all $1 \le i \le f$, then*

$$det(B) = -2^n + 1 - \sum_{i=1}^{f} 2^{(l_i - d_i \ (mod\ n)) + 1}.$$

Proof. According to (1), for any $S_{i_1,i_2,\ldots,i_t}$ $(t \ge 2)$, we have $l_{i_r} \in \{u, u+1, \ldots, v\}$ and $d_{i_r} \in (\{1, 2, \ldots, n\}/\{u, u+1, \ldots, v\})$ for al $1 \le r \le t$, or $d_{i_r} \in \{u, u+1, \ldots, v\}$ and $l_{i_r} \in (\{1, 2, \ldots, n\}/\{u, u+1, \ldots, v\})$ for all $1 \le r \le t$. It implies

that $S_{i_1,i_2,\ldots,i_t} = 0$. Hence

$$\begin{aligned}\det(B) &= -2^n + 1 + \sum_{i=1}^{f} S_i \\ &= -2^n + 1 - \sum_{i=1}^{f} 2^{(l_i - d_i \pmod n)+1}.\end{aligned}$$

□

This property can be used to construct ternary ring FCSRs.

2.2 Construction of ternary ring FCSRs with given connection integers

In ([3]), a kind of 2-adic automaton, extending the entries of the transition matrix to $\{-1, 0, 1\}$, called ternary FCSRs, is proposed for hardware implementations. Corresponding to the -1 in transition matrices, one has introduced a subtracter-with-carry to compute the difference between two 2-adic integers.

Algorithm 1 proposes a method of constructing ternary ring FCSRs for a given integer q. The non-adjacent form (NAF) of q is an expression $q = \sum_{i=0}^{n} q_i 2^i$ where $q_i \in \{-1, 0, 1\}$ and no two consecutive digits q_i are both nonzero. NAF(q) can be efficiently computed using an algorithm proposed in ([8]). Algorithm 1 is a generalization of the method proposed in ([14],[17]). In fact, if we let $1 \le d_f < d_{f-1} < \cdots < d_1 < n/2 < l_1 < l_2 < \cdots < l_f < n$, then it is the case in ([14],[17]). An efficient operation of step 4 and step 8 is as follows:

1. Randomly choose $1 \le f_0 \le f - 1$.
2. $(l_i, d_i) = \begin{cases} (s_i + \lfloor \frac{n-s_i}{2} \rfloor, 1 + \lfloor \frac{n-s_i}{2} \rfloor) & 1 \le i \le f_0 \\ (1 + \lceil \frac{s_i}{2} \rceil, n + 2 - s_i + \lceil \frac{s_i}{2} \rceil) & f_0 < i \le f, \end{cases}$

where $\lfloor x \rfloor$ denotes the largest integer smaller than x and $\lceil x \rceil$ denotes the smallest integer larger than x.

Theorem 3. *Let A be a transition matrix constructed by Algorithm 1 with a given positive odd q, then $det(I - 2A) = -q$.*

Proof. If $q_0 = -1$, the proof is similar to Proposition 1, replacing the feedbacks of the connection matrix B by $b_{l_i,d_i} = -2q_i$. If $q_0 = 1$, in step 7 let $b_{n,n} = -1$, then the term $(-1)^{\epsilon(1,2,\ldots,n)} \cdot b_{1,1} b_{2,2} \cdots b_{n,n} = -1$ and the sum of the nonzero terms without feedbacks is $-2^n - 1$. Moreover, if $l_i > d_i$, the term S_i must contain $b_{n,n} = -1$. So it will change the sign. The rest of the proof is according to Proposition 1. □

Example 1. Let $q = 347$. Then NAF(347) $= 2^9 - 2^7 - 2^5 - 2^2 - 1$. Let the feedbacks be $a_{8,4} = -1, a_{7,6} = -1, a_{1,3} = -1$. Then the transition matrix is

Algorithm 1 Construction of ternary ring FCSRs with a given connection integer

Input: A positive odd q.

Output: A transition matrix A with a critical path of length 1, a fan-out of 2 and $\det(I-2A)=-q$.

1: Compute NAF$(q)=2^n+\sum_{i=1}^{f}q_i2^{s_i}+q_0$, where $q_i=\pm1$ $(0\le i\le f)$, $n\ge s_f+2$ and $s_{i+1}\ge s_i+2$ $(1\le i\le f-1)$.

2: $A\leftarrow(a_{l,d})_{1\le l,d\le n}$ with $a_{l,d}\leftarrow\begin{cases}1 \text{ if } d\equiv l+1 \pmod n\\ 0 \text{ otherwise}\end{cases}$

3: **if** q_0=-1 **then**

4: Found (l_i,d_i) $(1\le i\le f)$ such that $l_i-d_i(\mathrm{mod}\ n)=s_i-1$ and there exist integers $1\le u<v\le n$ that all $l_i\in\{u,u+1,\ldots,v\}$ and all $d_i\in(\{1,2,\ldots,n\}/\{u,u+1,\ldots,v\})$, or all $d_i\in\{u,u+1,\ldots,v\}$ and all $l_i\in(\{1,2,\ldots,n\}/\{u,u+1,\ldots,v\})$.

5: $a_{l_i,d_i}\leftarrow q_i$ for $1\le i\le f$

6: **else**

7: $a_{n,n}\leftarrow 1$

8: Found (l_i,d_i) $(1\le i\le f)$ such that $l_i-d_i(\mathrm{mod}\ n)=s_i-1$ and there exist integers $1\le u<v\le n-1$ that all $l_i\in\{u,u+1,\ldots,v\}$ and all $d_i\in(\{1,2,\ldots,n-1\}/\{u,u+1,\ldots,v\})$, or all $d_i\in\{u,u+1,\ldots,v\}$ and all $l_i\in(\{1,2,\ldots,n-1\}/\{u,u+1,\ldots,v\})$.

9: **for** $1\le i\le f$ **do**

10: **if** $l_i>d_i$ **then**

11: $a_{l_i,d_i}\leftarrow -q_i$

12: **else**

13: $a_{l_i,d_i}\leftarrow q_i$

14: **end if**

15: **end for**

16: **end if**

17: **return** A

$$A=\begin{pmatrix}
0&1&-1&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&-1&0&1&0\\
0&0&0&-1&0&0&0&0&1\\
1&0&0&0&0&0&0&0&0
\end{pmatrix}$$

and $\det(I-2A)=-q$. □

A ternary ring FCSRs constructed by Algorithm 1 is a "defective FCSR" proposed in ([14]), with the number of feedbacks less than $n/2$. So we have to use methods presented in ([17]) to add some adders and subtracters for improving its security. We have no intention to traverse the methods here.

3 Construction of binary transition matrices

3.1 Main idea

To construct a binary ring FCSR, the method proposed for the ternary case is hard to be used, because there is no binary signed digit representation of integers like the non-adjacent form and the elementary transformations of connection matrices proposed in ([17]) for adding feedbacks are not suitable for the binary case. Therefore, we have to consider some more complicated cases than Proposition 1.

Proposition 2. *Let B be a connection matrix with f_1 feedbacks in the lower triangular: b_{l_i,d_i} $(1 \le i \le f_1)$, where $l_i > d_i$ and any two different (l_i, d_i), (l_j, d_j) $(l_j > l_i)$ have only two forms $d_j < d_i < l_i < l_j$ or $d_i < l_i < d_j < l_j$, and f_2 feedbacks in the upper triangular: $b_{l'_i,d'_i}$ $(l'_i < d'_i,\ 1 \le i \le f_2)$. Consider term $S_{i_1,i_2,\ldots,i_t}$ of (1) in Case I. Then for any upper triangular feedback $b_{l'_\mu,d'_\mu}$ in this term, there exists a lower triangular feedback b_{l_ν,d_ν} in this term that $d_\nu \le l'_\mu < d'_\mu \le l_\nu$.*

Proof. Suppose there is a upper triangular feedback $b_{l_{i_k},d'_{i_{\tilde{k}}}}$ in the term $S_{i_1,i_2,\ldots,i_t}$ of (1) in Case I, where $1 \le k < \tilde{k} \le t$. Then the number of $l_{i_r} < l_{i_k}$ is $k-1$, and the number of $d'_{i_r} \le l_{i_k}$ is k. It implies that there is at least one $d'_{i_{\tilde{\tau}}}$ of a feedback $b_{l'_{i_{\tilde{\tau}}},d'_{i_{\tilde{\tau}}}}$, where $l'_{i_{\tilde{\tau}}} > l_{i_k}$. If $l'_{i_{\tilde{\tau}}} \ge d'_{i_{\tilde{k}}}$, then we have $d'_{i_{\tilde{\tau}}} \le l_{i_k} < d'_{i_{\tilde{k}}} \le l'_{i_{\tilde{\tau}}}$. Otherwise, $l_{i_k} < l'_{i_{\tilde{\tau}}} < d'_{i_{\tilde{k}}}$. Suppose the number of such $d'_{i_{\tilde{\tau}}}$ is τ $(\tau \ge 1)$ and the maximum of their corresponding $l'_{i_{\tilde{\tau}}}$ is l_{i_λ}. Then the number of $l_{i_r} < l_{i_\lambda}$ which have yet to match feedbacks is $\lambda - \tau - 1$, and the number of $d'_{i_r} \le l_{i_\lambda}$ which have yet to match feedbacks is $\lambda - \tau$. Hence there is at least one $d'_{i_{\tilde{\tau}}}$ of a feedback $b_{l'_{i_{\tilde{\tau}}},d'_{i_{\tilde{\tau}}}}$, where $l'_{i_{\tilde{\tau}}} > l_{i_\lambda}$. If $l_{i_k} \le d'_{i_{\tilde{\tau}}} < d'_{i_{\tilde{k}}}$ then for $b_{l'_{i_{\tilde{\tau}}},d'_{i_{\tilde{\tau}}}}$ and $b_{l_{i_\lambda},d_{i_\lambda}}$ we have $d_{i_\lambda} < d'_{i_{\tilde{\tau}}} < l_{i_\lambda} < l'_{i_{\tilde{\tau}}}$. It is a contradiction. Then $d'_{i_{\tilde{\tau}}} < l_{i_k} < d'_{i_{\tilde{k}}} \le l'_{i_{\tilde{\tau}}}$. □

Proposition 3. *Let B be a connection matrix. Let $S_{i_1,i_2,\ldots,i_{t_1}}$ and $S_{j_1,j_2,\ldots,j_{t_2}}$ be two nonzero terms of (1) in Case I. If $l_{i_{t_1}} < d'_{j_1}$, then we have*

$$S_{i_1,i_2,\ldots,i_{t_1},j_1,j_2,\ldots,j_{t_2}} = S_{i_1,i_2,\ldots,i_{t_1}} \cdot S_{j_1,j_2,\ldots,j_{t_2}}.$$

Proof. Suppose $(d'_{i_1}, d'_{i_2}, \ldots, d'_{i_{t_1}})$ and $(d'_{j_1}, d'_{j_2}, \ldots, d'_{j_{t_2}})$ are the sequential permutations of $(d_{i_1}, d_{i_2}, \ldots, d_{i_{t_1}})$ and $(d_{j_1}, d_{j_2}, \ldots, d_{j_{t_2}})$ respectively. If $l_{i_{t_1}} < d'_{j_1}$, then $d'_{i_{t_1}} < d'_{j_1}$ and we have

$$\begin{aligned}
&S_{i_1,i_2,\ldots,i_{t_1},j_1,j_2,\ldots,j_{t_2}} \\
&= (-1)^{t_1+t_2+\epsilon(d_{i_1},d_{i_2},\ldots,d_{i_{t_1}},d_{j_1},d_{j_2},\ldots,d_{j_{t_2}})} \cdot 2^{t_1+t_2+\sum_{r=1}^{t_1}(l_{i_r}-d'_{i_r})+\sum_{r=1}^{t_2}(l_{j_r}-d'_{j_r})} \\
&= (-1)^{t_1+\epsilon(d_{i_1},d_{i_2},\ldots,d_{i_{t_1}})} \cdot 2^{t_1+\sum_{r=1}^{t_1}(l_{i_r}-d'_{i_r})} \cdot (-1)^{t_2+\epsilon(d_{j_1},d_{j_2},\ldots,d_{j_{t_2}})} \cdot 2^{t_2+\sum_{r=1}^{t_2}(l_{j_r}-d'_{j_r})} \\
&= S_{i_1,i_2,\ldots,i_{t_1}} \cdot S_{j_1,j_2,\ldots,j_{t_2}}.
\end{aligned}$$

□

According to Proposition 2 and Proposition 3, we propose a method of constructing binary ring FCSRs. As is shown in Algorithm 2, we first construct a transition matrix A_0 with f_1 feedbacks. The coordinate (i, j) of any feedback satisfies $1 \leq i, j < x$. We compute $q_0 = \det(I - 2A_0)$ and the sum p of all $S_{i_1,i_2,\ldots,i_{t_1}}$ of (1) in Case I of $(I - 2A_0)$. Then in the loop, we randomly choose a vector $(i_1, i_2, \ldots, i_{f_2}) \subseteq (1, 2, \ldots, (n - x + 1)/2)$, compute $q = q_0 - (p + 1) \sum_{r=1}^{f_2} 4^{i_r}$, and test whether $|q|$ is a safe prime and 2 is its primitive root. The test of primitive can be reduced to test whether $2^{\frac{|q|-1}{2}} \equiv 1$, because there are only two possible values of the order of 2 modulo a safe prime $|q|$, i.e. $(|q| - 1)$ and $(|q| - 1)/2$. If q passes the tests, then we set up f_2 feedbacks corresponding to the chosen vector.

For a pre-constructed A_0, we should require $\gcd(q_0, p+1) = 1$. Otherwise, $|q|$ can never be a prime. Another requirement is $q < -2^n$. In fact, an n length ring FCSR may have its connection integer of about $n/2$ length. For example, consider an n length ring FCSR with only two feedbacks $a_{n,\frac{n}{2}+1} = 1$ and $a_{\frac{n}{2},1} = 1$, then its connection integer is $2^{\frac{n}{2}+1} + 1$. According to Theorem 1, the output sequences of such an FCSR are with period smaller than expected. Hence we propose some tests to avoid this problem.

The correctness of Algorithm 2 is shown in the following theorem:

Theorem 4. *Let A be a transition matrix constructed by Algorithm 2, then $det(I - 2A) = q$.*

Proof. Let $B = I - 2A$. There are f_1 feedbacks $b_{l,d} = -2$ where $1 \leq l, d < x$ and f_2 feedbacks $b_{\frac{n+x-1}{2}+j_r, \frac{n+x+1}{2}-j_r} = -2$ for $1 \leq r \leq f_2$ where $(j_1, j_2, \ldots, j_{f_2}) \subseteq (1, 2, \ldots, (n - x + 1)/2)$ (suppose $j_1 < j_2 < \cdots < j_{f_2}$). Firstly, suppose a term $S_{i_1,i_2,\ldots,i_t}$ of (1) includes k ($k \geq 2$) feedbacks: $b_{\frac{n+x-1}{2}+j_{r_1}, \frac{n+x+1}{2}-j_{r_1}}, \ldots, b_{\frac{n+x-1}{2}+j_{r_k}, \frac{n+x+1}{2}-j_{r_k}}$, where $(r_1, \ldots, r_k) \subseteq (1, \ldots, f_2)$ and $r_1 < r_2 < \cdots < r_k$. Then we have $\cdots < (d'_{i_{t-k+1}} = \frac{n+x+1}{2} - j_{r_k}) < \cdots < (d'_{i_t} = \frac{n+x+1}{2} - j_{r_1}) < (l_{i_{t-k+1}} = \frac{n+x-1}{2} + j_{r_1}) < \cdots < (l_{i_t} = \frac{n+x-1}{2} + j_{r_k})$, thus $S_{i_1,i_2,\ldots,i_t} = 0$. Hence a nonzero term $S_{i_1,i_2,\ldots,i_t}$ of (1) can not include more than one $b_{\frac{n+x-1}{2}+j, \frac{n+x+1}{2}-j}$ $(j = 1, 2, \ldots, (n-x+1)/2)$.

Suppose a nonzero term $S_{i_1,i_2,\ldots,i_t}$ includes $b_{\frac{n+x-1}{2}+j, \frac{n+x+1}{2}-j}$, then we have $l_{i_t} = \frac{n+x-1}{2} + j$ and $d'_{i_t} = d_{i_t} = \frac{n+x+1}{2} - j$. So if $t > 1$, the term $S_{i_1,i_2,\ldots,i_{t-1}}$ must be in Case I and the feedbacks in this term must all of $b_{l,d} = -2$ where $1 \leq l, d < x$. On the other hand, suppose $S_{i_1,i_2,\ldots,i_{t'}}$ is a nonzero term of (1) in Case I where $l_{i_{t'}} < x$ and let $l_{i_{t'+1}} = \frac{n+x-1}{2} + j$, $d_{i_{t'+1}} = \frac{n+x+1}{2} - j$ $(1 \leq j \leq (n - x + 1)/2)$. According to Proposition 3, we have

$$\begin{aligned} & S_{i_1,i_2,\ldots,i_{t'},i_{t'+1}} \\ &= S_{i_1,i_2,\ldots,i_{t'}} \cdot (-2^{l_{i_{t'+1}} - d_{i_{t'+1}} + 1}) \\ &= S_{i_1,i_2,\ldots,i_{t'}} \cdot (-2^{2j}) \\ &= -4^j \cdot S_{i_1,i_2,\ldots,i_{t'}}. \end{aligned} \tag{3}$$

The above argument implies that any nonzero term of (1) in Case II only have feedbacks such as $b_{l,d}$ where $1 \leq l, d < x$. Denote the sum of them by $S(\text{II})$.

Denote the sum of all nonzero terms of (1) in Case I by $S(\mathrm{I})$. Suppose at the end of the loop we have chose $(j_1, j_2, \ldots, j_{f_2})$ in step 4, according to (3) we have

$$\begin{aligned}
&\det(B)\\
&= -2^n + 1 + S(\mathrm{I}) + S(\mathrm{II})\\
&= -2^n + 1 + S(\mathrm{II}) + p - p \cdot \sum_{r=1}^{f_2} 4^{j_r} - \sum_{r=1}^{f_2} 4^{j_r}\\
&= q_0 - (p+1) \cdot \sum_{r=1}^{f_2} 4^{j_r}\\
&= q.
\end{aligned}$$

□

The efficiency of Algorithm 2 will be further discussed in the next subsection.

3.2 Practical constructions

In Algorithm 2, after the pre-construction we search a safe prime with 2 is its primitive root by the loop from step 3 to step 12. In each iteration, there are some additions and tests. The efficiency is roughly equal to a random search of a such integer. The density of such integers can be estimated according to Sophie-Germain Prime Density Conjecture and Artin's Conjecture in number theory. By Sophie-Germain Prime Density Conjecture there is about $\frac{1.32N}{\ln^2 N}$ Sophie-Germain primes $\leq N$. Then there is about $\frac{1.32(N'-1)/2}{\ln^2 (N'-1)/2}$ safe primes $\leq N'$. Artin's Conjecture argues that there is about $1/3$ primes such that 2 is their primitive root.

For instance, we estimate the number of safe primes with 2 is its primitive root in range $(2^{160}, 2^{161})$. It is about $\frac{1}{3} \cdot (\frac{1.32 \cdot 2^{160}}{\ln^2 2^{160}} - \frac{1.32 \cdot 2^{159}}{\ln^2 2^{159}})$. Hence the density of such integers in range $(2^{160}, 2^{161})$ is about 0.000018. We do a simulation: Randomly select 2^{25} integers from range $(2^{160}, 2^{161})$. In these integers, there are 854 safe primes with primitive root 2. The density is about 0.000025.

So we should select appropriate x and f_2 so that the search space is large enough comparing to the estimated density. For example, when $n = 160$, let $x = 81$ an $f_2 = 28$, then there are $\binom{(n-x+1)/2}{f_2} = \binom{40}{28}$ possible integers for search. It is much larger than the necessity.

An remaining problem is that we should design an effectiveness pre-construction in which the sum of all $S_{i_1,i_2,\ldots,i_t}$ of (1) in Case I is easy to be calculated. A most natural idea is let all feedbacks $b_{l,d}$ where $1 \leq d < l < x$ meet the conditions of Proposition 2 and let $l - d$ be small. Then according to Proposition 2 and Proposition 3 we may easily calculate the sum. A simplest method is presented in Algorithm 3.

The computations of Algorithm 3 are simple except for the computation of a determinant in step 6. However, on average we need a few iterations to finish the pre-construction, because the pass probability of the test in step 7 can be estimated as follows: Suppose the probability $q_0 < -2^n$ is $1/2$. Since the probabilities of $q_0 \not\equiv 0 \pmod 3$ and $q_0 \not\equiv 0 \pmod 7$ are $1/2$ and $3/4$ respectively, the pass probability is about $3/16$. Therefore Algorithm 3 is effectiveness.

The following theorem proves the correctness of Algorithm 3:

Theorem 5. *Let A be a transition matrix constructed by Algorithm 3 and $B = I - 2A$. Then the sum $S(I)$ of all nonzero terms of (1) in Case I is p.*

Proof. We can see that the lower triangular feedbacks of B meet the conditions of Proposition 2. Since any upper triangular feedback $b_{l,d}$ has $d - l \geq 3$, then any nonzero term in Case I includes no upper triangular feedback according to Proposition 2. On the other hand, suppose a term $S_{i_1,i_2,\ldots,i_k}$ of (1) contains just k $(1 \leq k \leq g)$ lower triangular feedbacks: $b_{l_{i_1},d_{i_1}}, \ldots, b_{l_{i_k},d_{i_k}}$, where $(i_1, \ldots, i_k) \subseteq (1, \ldots, g)$ and $i_1 < i_2 < \cdots < i_k$, then we have $d_{i_1} < l_{i_1} < d_{i_2} < l_{i_2} < \cdots < d_{i_k} < l_{i_k}$, thus in Case I. Therefore, according to Proposition 3, we have

$$\begin{aligned} S(I) &= \sum_{(i_1,\ldots,i_k)\subseteq(1,\ldots,g)} S_{i_1,i_2,\ldots,i_k} \\ &= \sum_{(i_1,\ldots,i_k)\subseteq(1,\ldots,g)} S_{i_1} \cdot Si_2 \cdot \cdots \cdot S_{i_k} \\ &= (S_1 + 1) \cdot (S_2 + 1) \cdot \cdots \cdot (S_g + 1) - 1 \\ &= (\prod_{i=1}^{g} (-2^{l_i - d_i + 1} + 1)) - 1 \\ &= 3^{\delta_1} \cdot 7^{\delta_2} - 1 \\ &= p. \end{aligned}$$

□

Now we can use Algorithm 2 and Algorithm 3 to construct binary ring FCSRs for hardware implementations. To verify the effectiveness of our construction, we do this construction 1000 times by randomly selecting n from range $[128, 256]$ and choosing appropriate integers x, g, f_1, f_2. Every time we can successfully get an expected binary ring FCSR. We present two examples below. They can be applied to the stream cipher F-FCSR-H v3 and the stream cipher F-FCSR-16 v3 respectively.

Example 2. A ring FCSR of size 160 bits for F-FCSR-H v3

The transition matrix $A = (a_{l,d})_{1 \leq l,d \leq 160}$ is given by:

$$\begin{cases} a_{l,l+1 \pmod{160}} = 1 & \text{for } 1 \leq l \leq 160 \\ a_{l,d} = 1 & \text{for } (l,d) \in \mathcal{U} \\ a_{l,d} = 0 & \text{otherwise} \end{cases}$$

where $\mathcal{U}$ is the set:

$$\left\{\begin{array}{ccccccccc}
(1,35) & (2,26) & (3,2) & (4,49) & (7,6) & (9,62) & (10,38) & (11,24) & (12,11) \\
(13,30) & (14,45) & (15,32) & (18,17) & (19,46) & (20,27) & (22,58) & (23,44) & (24,23) \\
(25,37) & (27,25) & (28,56) & (29,42) & (31,29) & (33,40) & (34,33) & (35,39) & (36,43) \\
(37,36) & (38,60) & (40,54) & (41,55) & (42,41) & (43,48) & (48,71) & (49,47) & (51,50) \\
(52,69) & (53,52) & (55,63) & (56,64) & (58,57) & (59,78) & (60,67) & (63,61) & (64,68) \\
(65,75) & (66,76) & (68,66) & (69,73) & (70,74) & (71,80) & (72,79) & (74,72) & (79,77) \\
(123,118) & (125,116) & (126,115) & (127,114) & (128,113) & (129,112) & (131,110) & (132,109) & (133,108) \\
(134,107) & (135,106) & (136,105) & (137,104) & (138,103) & (140,101) & (144,97) & (145,96) & (146,95) \\
(147,94) & (148,93) & (149,92) & (152,89) & (153,88) & (154,87) & (155,86) & (157,84) & (159,82) \\
(160,81) & & & & & & & &
\end{array}\right\}$$

This FCSR has a connection integer $q = -148731335080631408441305456521194031482433940481 9$ which is safe prime with 2 as its primitive root. It has a cost of 82 adders-with-carry, a critical path of length 1 and fan-out 2, and a diffusion delay of 56. □

Example 3. A ring FCSR of size 256 bits for F-FCSR-16 v3

The transition matrix $A = (a_{l,d})_{1 \leq l,d \leq 256}$ is given by:

$$\begin{cases} a_{l,l+1 \pmod{256}} = 1 & \text{for } 1 \leq i \leq 256 \\ a_{l,d} = 1 & \text{for } (l,d) \in \mathcal{U} \\ a_{l,d} = 0 & \text{otherwise} \end{cases}$$

where $\mathcal{U}$ is the set:

$$\left\{\begin{array}{ccccccccc}
(2,86) & (3,2) & (6,4) & (7,46) & (12,10) & (15,51) & (17,15) & (18,69) & (19,48) \\
(20,18) & (22,59) & (24,22) & (26,25) & (27,54) & (28,38) & (31,30) & (34,33) & (37,52) \\
(38,37) & (39,77) & (41,40) & (43,62) & (44,43) & (45,101) & (46,75) & (47,89) & (48,57) \\
(50,49) & (53,76) & (55,53) & (57,67) & (58,85) & (59,82) & (60,58) & (63,80) & (64,96) \\
(65,99) & (66,65) & (67,100) & (70,81) & (71,70) & (72,90) & (73,88) & (74,72) & (78,83) \\
(79,78) & (83,117) & (86,84) & (87,93) & (88,97) & (89,87) & (91,95) & (92,102) & (93,91) \\
(94,110) & (95,104) & (96,105) & (97,111) & (98,106) & (99,109) & (100,98) & (103,107) & (104,116) \\
(105,103) & (106,120) & (107,113) & (108,121) & (109,114) & (110,108) & (113,122) & (114,112) & (115,119) \\
(116,124) & (117,115) & (118,127) & (119,118) & (121,128) & (122,126) & (124,123) & (127,125) & (194,191) \\
(195,190) & (196,189) & (197,188) & (198,187) & (200,185) & (201,184) & (202,183) & (204,181) & (205,180) \\
(206,179) & (209,176) & (210,175) & (211,174) & (213,172) & (214,171) & (215,170) & (216,169) & (218,167) \\
(219,166) & (220,165) & (222,163) & (223,162) & (224,161) & (226,159) & (228,157) & (229,156) & (230,155) \\
(232,153) & (233,152) & (234,151) & (236,149) & (237,148) & (238,147) & (239,146) & (240,145) & (241,144) \\
(242,143) & (243,142) & (245,140) & (246,139) & (247,138) & (248,137) & (249,136) & (250,135) & (251,134) \\
(252,133) & (253,132) & (254,131) & (256,129) & & & & &
\end{array}\right\}$$

This FCSR has a connection integer $q = -155290126080730714984253542403912423188027751232308607544737047876915834077259$ which is safe prime with 2 as its primitive root. It has a cost of 130 adders-with-carry, a critical path of length 1 and fan-out 2, and a diffusion delay of 89. □

4 Conclusion

Ring FCSRs are applied to hardware stream ciphers. In this paper, we have presented how to calculate the determinant $\det(I-2A)$ for transition matrices of ring FCSRs by observing their feedbacks. According to the calculation method, we have proposed algorithms of constructing binary ring FCSRs suitable for hardware implementations. The efficiency of the construction is near to randomly search safe primes with primitive root 2. In addition, a generalized algorithm for constructing ternary ring FCSRs with given connection integers also has been proposed.

Interestingly enough, we found that these results can be trivially grafted onto the construction of ring LFSRs. The problem of construction of ring LFSRs have been proposed in ([4],[16]).

References

1. F. Arnault and T. P. Berger, "F-FCSR: Design of a new class of stream ciphers," in Fast Software Encryption. Berlin, Germany: Springer-Verlag, 2005, pp. 83-97.
2. F. Arnault, T. Berger, C. Lauradoux, M. Minier, and B. Pousse, "A new approach for FCSRs," in Selected Areas in Cryptography (Lecture Notes in Computer Science), vol. 5867, M. J. Jacobson, Jr., V. Rijmen, and R. Safavi-Naini, Eds. New York, NY, USA: Springer-Verlag, 2009, pp. 433-448.
3. F. Arnault, T. P. Berger, and B. Pousse, "A matrix approach for FCSR automata," Cryptography Commun., vol. 3, no. 2, pp. 109C139, Jun. 2011.
4. F. Arnault, T. Berger, M. Minier, and B. Pousse, "Revisiting LFSRs for cryptographic applications," IEEE Trans. Inf. Theory, vol. 57, no. 12, pp. 8095-8113, Dec. 2011.
5. T. Berger, M. Minier, and B. Pousse, "Software oriented stream ciphers based upon FCSRs in diversified mode," in Progress in Cryptology-INDOCRYPT 2009, pp. 119-135.
6. S. Fischer, W. Meier, and D. Stegemann, "Equivalent representations of the F-FCSR keystream generator," in Proc. ECRYPT Netw. Excellence, SASC Workshop, Feb. 2008, pp. 87-94. [Online]. Available: http://www.ecrypt.eu.org/stvl/sasc2008/
7. M. Goresky and A. M. Klapper, "Fibonacci and Galois representations of feedback-with-carry shift registers," IEEE Trans. Inf. Theory, vol. 48, no. 11, pp. 2826C2836, Nov. 2002.
8. D. Hankerson, S. Vanstone, A. Menezes. "Guide to Elliptic Curve Cryptography," New York: Springer, 2004.
9. M. Hell and T. Johansson, "Breaking the stream ciphers F-FCSR-H and F-FCSR-16 in real time," J. Cryptol., vol. 24, no. 3, pp. 427-445, 2011.
10. A. Klapper and M. Goresky, "2-adic shift registers," in Fast Software Encryption, vol. 809, R. Anderson, Ed. Berlin, Germany: Springer-Verlag, 1994, pp. 174-178.
11. A. Klapper and M. Goresky, "Feedback shift registers, 2-adic span, and combiners with memory," J. Cryptol., vol. 10, no. 2, pp. 111-147, Mar. 1997.
12. A. Klapper, "A survey of feedback with carry shift registers," in Sequences and Their Applications (Lecture Notes in Computer Science), vol. 3486, T. Helleseth, D. Sarwate, H.-Y. Song, and K. Yang, Eds. Berlin, Germany: Springer-Verlag, 2005, pp. 56-71.

13. A. Klapper and M. Goresky, "large period nearly debruijn FCSR sequences," advances in cryptologyeurocrypt'95, Springer-Berlin Heidelberg, 1995 pp. 263-273.
14. L. Zhiqiang and P. Dingyi, "Constructing a ternary FCSR with a given connection integer," Tech. Rep. 2011/358. [Online]. Available: http://eprint.iacr.org/
15. L. Zhiqiang, K. Lishan, L. Dongdai, and G. Jian: "On the LFSRization of a Class of FCSR Automata," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences Vol. 98, no. 1, pp. 434-440, 2015.
16. G. Mrugalski, J. Rajski, and J. Tyszer, "Ring generatorsNew devices for embedded test applications," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 23, no. 9, pp. 1306-1320, Sep. 2004.
17. P. Dingyi, L. Zhiqiang and Z. Xiaolei, "Construction of Transition Matrices for Ternary Ring Feedback with Carry Shift Registers," IEEE Trans. Inf. Theory, vol. 61, no. 5, pp. 2042-2951, 2015.
18. P. Stankovski, M. Hell, and T. Johansson, "An efficient state recovery attack on the X-FCSR family of stream ciphers," J. Cryptol., vol. 27, no. 1, pp. 1-22, 2014.
19. T. Tian and W.-F. Qi, "Linearity properties of binary FCSR sequences," Designs, Codes Cryptography, vol. 52, no. 3, pp. 249-262, Sep. 2009.
20. H. Wang, P. Stankovski, and T. Johansson, "A generalized birthday approach for efficiently finding linear relations in ℓ-sequences," Designs, Codes Cryptography, vol. 74, no. 1, pp. 41-57, 2015.

Algorithm 2 Construction of binary ring FCSRs for hardware implementations

Input: n the length of the FCSR, x ($x < n$) an integer where $n - x$ is odd, f_1 the number of feedbacks for the pre-construction and f_2 ($f_2 < \frac{n+x-1}{2}$) the number of feedbacks for randomly selections.

Output: A binary transition matrix A with $f_1 + f_2$ feedbacks and meeting the conditions from C_1 to C_4.

$A \leftarrow (a_{l,d})_{1 \le l,d \le n}$ with $a_{l,d} \leftarrow \begin{cases} 1 \text{ if } l \equiv d+1 \pmod{n} \\ 0 \text{ otherwise} \end{cases}$

Pre-construction:

- Choose f_1 feedbacks $a_{l,d} \leftarrow 1$ meeting condition C_4, where $1 \le l, d < x$
- $q_0 \leftarrow \det(I - 2A)$
- **if** $q_0 > -2^n$ **then**
  Redo pre-construction
- **end if**
- Compute p: the sum of all $S_{i_1,i_2,\ldots,i_t}$ of (1) in Case I
- **if** $\gcd(q_0, p+1) \neq 1 || q_0 - \frac{1}{3}(p+1)(4^{f_2+1} - 4) > -2^n$ **then**
  Redo pre-construction
- **end if**
- **for** $1 \le i \le \frac{n+x-1}{2}$ **do**
  $p_i \leftarrow -4^i \cdot (p+1)$
- **end for**

loop
  Randomly choose $(i_1, i_2, \ldots, i_{f_2}) \subseteq (1, 2, \ldots, (n-x+1)/2)$
  Compute $q = q_0 + \sum_{r=1}^{f_2} p_{i_r}$
  if $|q|$ is a safe prime & $2^{\frac{|q|-1}{2}} \not\equiv 1$ **then**
    for $1 \le r \le f_2$ **do**
      $a_{\frac{n+x-1}{2}+i_r, \frac{n+x+1}{2}-i_r} \leftarrow 1$
    end for
    break
  end if
end loop
return A, q

Algorithm 3 A practical pre-construction

1: Randomly choose $1 \leq d_1 < l_1 < d_2 < l_2 < \cdots < d_g < l_g < x$, where $l_i - d_i = 1$ or 2 $(1 \leq i \leq g)$ and g is even. Count δ_1 and δ_2 the numbers of $l_i - g_i = 1$ and 2 $(1 \leq i \leq g)$ respectively.
2: **for** $1 \leq i \leq g$ **do**
3: $\quad a_{l_i,d_i} \leftarrow 1$
4: **end for**
5: Randomly choose $f_1 - g$ feedbacks $a_{l,d} \leftarrow 1$ satisfying condition C_4, where $l \in (\{1,\ldots,x-1\}/\{l_1,\ldots,l_g\})$, $d \in (\{1,\ldots,x-1\}/\{d_1,\ldots,d_g\})$ and $d - l \geq 3$
6: $q_0 \leftarrow \det(I - 2A)$
7: **if** $q_0 > -2^n || q_0 \equiv 0 \pmod 3 || q_0 \equiv 0 \pmod 7$ **then**
8: $\quad$ **Goto** step 1
9: **end if**
10: $p \leftarrow 3^{\delta_1} \cdot 7^{\delta_2} - 1$
11: **for** $1 \leq i \leq \frac{n+x-1}{2}$ **do**
12: $\quad p_i \leftarrow -4^i \cdot (p+1)$
13: **end for**