Some Primitives Based on Jumping LFSRs with Determined Period

Arash Mirzaei, Mahdi Sajadieh, and Mohammad Dakhilalian

[1] Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran
[2]
Department of Electrical Engineering , Department of Electrical Engineering, Islamic Azad University, Khorasgan Branch, Iran
arash_mirzaei@ec.iut.ac.ir,
m.sajadieh@khuisf.ac.ir,
mdalian@cc.iut.ac.ir,

Abstract. Linear feedback shift registers (LFSRs) are used in many stream ciphers because of their maximal period and good statistical properties. Due to the linearity of the LFSR, its output cannot be directly used as the keystream. Different methods have been proposed to introduce the nonlinearity to the LFSR output. Irregular clocking is one of the methods to do this but the stream ciphers based on this method, are vulnerable to the side-channel attacks. In addition, the generation rate of the irregular clocked LFSRs is less than the corresponding regular ones. Jumping is a method of irregular clocking for LFSRs which may have non of the mentioned flaws but its output period cannot be determined.

In this paper, using the jumping LFSRs, some new primitives will be proposed. Structures of the new primitives have the determined lower bound of period and the resistance against the side-channel attacks. In some of the proposed structures, the lower bound of period can be determined without knowledge of the jump index. These structures are applicable when the calculation of the jump index is infeasible. The proposed structures can be used as primitives to design the software oriented and hardware oriented stream ciphers.

Keywords: Stream ciphers, LFSR, Jump index, Period

1 Introduction

In cryptography, the stream cipher is a symmetric key cipher that used to provide security to a communication channel. These types of ciphers produce a stream of pseudorandom bits, which is called the keystream. The keystream is often bitwise XORed with the plaintext (the ciphertext) to produce the ciphertext (the plaintext). Since a shared secret key K and a public initialization vector IV is used to initialize the stream cipher, generated keystream is the same at the sender and the receiver side.

Many stream ciphers have focused on the bit oriented linear feedback shift registers (LFSRs) because of their hardware implementation speed. In addition, the LFSR output can have the maximal period and good statistical properties [12]. Since the bit oriented LFSRs produce only one bit of output per iteration, their software implementation is not efficient for the modern processors. To address this drawback, word oriented LFSRs have been mentioned and recently many stream ciphers have been proposed based on them [1].

Since the output bits of the LFSRs (bit oriented or word oriented) are linearly dependent they cannot be directly used as a keystream. Thus non-linearity shall be introduced into the LFSR output. To do this, one way is to apply irregular clocking. Unfortunately, the LFSRs that use the irregular clocking are vulnerable to the timing, power and other side-channel attacks. In addition, the generation rate of these LFSRs is less than the regularly clocked ones. To address these flaws, in [7] a method called jumping was suggested for the bit oriented LFSRs. The jumping method is an efficient way to let a bit oriented LFSR, irregularly clock but without having to step through all the intermediate states. Number of the intermediate states is called the jump index of LFSR.

The most important disadvantage of the jumping method is that the jumping LFSR period depends on the jump index value and the jump controlling sequence which is often a pseudo-random sequence. Thus, determining the jumping LFSR period is not straightforward. In [2, 3], a method has been suggested to design the stream cipher Mickey which can be used to determine the lower bound for the jumping LFSR period regardless of the jump controlling sequence. This method is just applicable to the bit oriented LFSRs and moreover, its resistance against the side-channel attacks has not been proved.

In [13], the jumping method has been extended to a type of the word oriented LFSRs called σ-LFSRs. The idea of the jumping method can be also easily extended to the other types of the word oriented LFSRs. For the jumping word oriented LFSRs, there are cases (depends on the LFSR length) which finding the jump index is plausible for them [13]. For these cases, determining the period is not possible. Furthermore, when the jump index value is known, no method has been suggested to determine the period or even a lower bound of the word oriented LFSRs period.

In this paper, using more than one jumping LFSR, some structures will be proposed which will have the determined lower bound of the period if the jump index value is known. Furthermore, some structures with the determined lower bound of the period will be proposed for cases where the jump index cannot be calculated for them. All of the proposed structures in this paper have resistance against the side-channel attacks and can be used as primitives to design the word oriented and the bit oriented stream ciphers. It should be noted that among the possible attacks, the focus of the paper is on side-channel attacks because irregular clocked LFSRs are potentially vulnerable to these types of attacks. Resistance of the new structures against other types of attacks depends on the properties of stream ciphers using the new structures as primitives.

The rest of the paper is organized as follows. In Section 2, some types of LFSRs are described. A brief description of the jumping method and its flaws are presented in Section 3. We present our new structures and their properties in Section 4. Finally, the conclusion is presented in Section 5.

2 Brief Description of LFSRs

In this section the bit oriented LFSRs and some types of the word oriented LFSRs are described which can be used in the new structures that will be proposed in Section 4. A word oriented LFSR is a linear shift register based on the linear recurrence relation ([9])

$$\mathbf{s}_{t+n} = \mathbf{s}_{t+n-1}\mathbf{C}_{n-1} + \mathbf{s}_{t+n-2}\mathbf{C}_{n-2} + \cdots + \mathbf{s}_t\mathbf{C}_0, \qquad t = 0, 1, \cdots \tag{1}$$

where each $\mathbf{C}_i$ is a binary $w \times w$ matrix.

The state of the shift register is of the length $w \times n$ bits and at time t is denoted by $\mathbf{S}_t = (\mathbf{s}_t, \mathbf{s}_{t+1}, \cdots, \mathbf{s}_{t+n-1})$. The vector $(\mathbf{s}_0, \mathbf{s}_1, \cdots, \mathbf{s}_{n-1})$ is called the initial value. The initial value is assumed to be nonzero. According to the relation (1), the transition matrix of an LFSR which transforms $\mathbf{S}_t$ to $\mathbf{S}_{t+1} = \mathbf{S}_t.\mathbf{T}$ is of the form

$$\mathbf{T} = \begin{pmatrix} \mathbf{0}_{w\times w} & \mathbf{0}_{w\times w} & \cdots & \mathbf{0}_{w\times w} & \mathbf{C}_0 \\ \mathbf{I}_{w\times w} & \mathbf{0}_{w\times w} & \cdots & \mathbf{0}_{w\times w} & \mathbf{C}_1 \\ \mathbf{0}_{w\times w} & \mathbf{I}_{w\times w} & \cdots & \mathbf{0}_{w\times w} & \mathbf{C}_2 \\ \vdots & & \ddots & & \\ \mathbf{0}_{w\times w} & \mathbf{0}_{w\times w} & \cdots & \mathbf{I}_{w\times w} & \mathbf{C}_{n-1} \end{pmatrix} \tag{2}$$

If the output sequence generated by an LFSR attains the maximal period, the LFSR is called primitive. Theorem 1 determines the necessary and sufficient conditions for a word oriented LFSR to be a primitive one.

Theorem 1 ([9]).
Let the sequence $\boldsymbol{s}^{\infty} = \boldsymbol{s}_0, \boldsymbol{s}_1, \cdots$ be generated by an LFSR with linear recurrence $\boldsymbol{s}_{t+n} = \boldsymbol{s}_{t+n-1}\boldsymbol{C}_{n-1} + \boldsymbol{s}_{t+n-2}\boldsymbol{C}_{n-2} + \cdots + \boldsymbol{s}_t\boldsymbol{C}_0$ where $\boldsymbol{C}_0$ is an invertible binary matrix and $\boldsymbol{C}_l = (c_l^{ij})_{w\times w}$ for $l = 0, 1, \cdots, n-1$,

$$\boldsymbol{F}(x) = (f^{ij}(x))_{w\times w}$$

be the corresponding polynomial matrix of f(x) where

$$f^{ij}(x) = \delta^{ij}x^n + \sum_{l=0}^{n-1} c_l^{ij}x^l \in F_2[x], \quad \delta^{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{3}$$

Then LFSR will have the maximal period if and only if the determinant of the $\boldsymbol{F}(x)$ ($|\boldsymbol{F}(x)|$) is a primitive polynomial of degree $w \times n$ over $\mathbb{F}_2$.

Bit oriented LFSR is a special case of word oriented LFSR with $w = 1$ where it can be efficiently implemented in hardware. Thus, the bit oriented LFSR is a shift register based on the linear recurrence relation

$$s_{t+n} = s_{t+n-1}c_{n-1} + s_{t+n-2}c_{n-2} + \cdots + s_t c_0, \qquad t = 0, 1, \cdots \tag{4}$$

where each c_i is a bit 0 or 1.
For bit oriented LFSRs the transition matrix is an $n \times n$ binary matrix of the form

$$\mathbf{T} = \begin{pmatrix} 0 & 0 & \cdots & 0 & c_0 \\ 1 & 0 & \cdots & 0 & c_1 \\ 0 & 1 & \cdots & 0 & c_2 \\ \vdots & & \ddots & & \\ 0 & 0 & \cdots & 1 & c_{n-1} \end{pmatrix} \tag{5}$$

Based on Theorem 1 a bit oriented LFSR will be a primitive one if the characteristic polynomial of its transition matrix is a primitive polynomial of degree n over $\mathbb{F}_2$.

Different types of the word oriented LFSRs are distinguished from each other based on the form of the coefficients $(\mathbf{C}_i, i = 0, 1, \cdots, n-1)$. In the reminder of this section, some types of the word oriented LFSRs are described. It should be noted that all types of the word oriented LFSRs can be used in the new structures that will be proposed in Section 4.

TGFSR (Twisted Generalized Feedback Shift Register) [8] is a word oriented linear feedback shift register based on the linear recurrence relation

$$\mathbf{s}_{t+n} = \mathbf{s}_{t+m} + \mathbf{s}_t.\mathbf{A} (t = 0, 1, \cdots) \tag{6}$$

where $\mathbf{A}$ is an invertible $w \times w$ binary matrix, m is a positive number between 1 and $n-1$ and $\mathbf{s}_t$ denotes a word and is regarded as a row vector of length w over $\mathbb{F}_2$. According to the relation (1), for the TGFSR $\mathbf{C}_0 = \mathbf{A}$, $\mathbf{C}_m = \mathbf{I}_{w\times w}$ and $\mathbf{C}_i = \mathbf{0}_{w\times w}$ for $i = 1, 2, \cdots, m-1, m+1, \cdots, n-1$.

By choosing suitable values for n, m and $\mathbf{A}$, the state of the TGFSR attains the maximal period $2^{n\times w} - 1$. It means the minimum value for k which satisfies $\mathbf{s}_{t+k} = \mathbf{s}_t$, for $t = 0, 1, \cdots$ is $2^{w\times n} - 1$. The generated sequence of a TGFSR is denoted by $S(n, m, \mathbf{A}) = \mathbf{s}_0, \mathbf{s}_1, \cdots$. Theorem 2 determines the necessary and sufficient conditions for a TGFSR generator to produce the maximal period sequence.

Theorem 2 ([8]).
Let $\phi_{\boldsymbol{A}}(t)$ be the characteristic polynomial of the matrix $\boldsymbol{A}$. The period of $S(n, m, \boldsymbol{A})$ is $2^{w\times n} - 1$ words if and only if $\phi_{\boldsymbol{A}}(t^n + t^m)$ is a primitive polynomial of degree $n \times w$.

The matrix $\mathbf{A}$ should be chosen so that $\mathbf{s}_t.\mathbf{A}$ can be calculated efficiently in the modern processors. The proposed $\mathbf{A}$ in [8] is a matrix of the form

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & & \\ 0 & 0 & 0 & \cdots & 1 \\ a_0 & a_1 & a_2 & \cdots & a_{w-1} \end{pmatrix} \tag{7}$$

For this matrix $\phi_A(t) = t^w + \sum_{i=0}^{w-1} a_i t^i$ and $\mathbf{s}.\mathbf{A}$ can be calculated by ($\mathbf{s}$ is a binary vector with content $(s_0, s_1, \cdots, s_{w-1})$ and $\mathbf{a}$ is the binary vector $(a_0, a_1, \cdots, a_{w-1})$)

$$\mathbf{s}.\mathbf{A} = \begin{cases} \mathbf{s} \ggg 1, & s_{w-1} = 0 \\ \mathbf{s} \ggg 1 \oplus \mathbf{a} & s_{w-1} = 1 \end{cases} \tag{8}$$

Another type of the word oriented LFSRs is σ-LFSR. σ-LFSRs are linear feedback shift registers based on the linear recurrence relation (1).

In σ-LFSRs $\mathbf{C}_i$ should be corresponding to one of the following operations ($\mathbf{s} = (s_0, \cdots, s_{w-1})$ and $\mathbf{v} = (v_0, \cdots, v_{w-1})$ are binary words of length w bits).

- AND with constant $\mathbf{v}$ which is denoted by $\Lambda_{\mathbf{v}}(\mathbf{s}) = (s_0 v_0, s_1 v_1, \cdots, s_{w-1} v_{w-1})$.
- Left shift by k bits which is denoted by $L_k(\mathbf{s}) = (s_k, \cdots, s_w, 0, 0, \cdots, 0)$.
- Right shift by k bits which is denoted by $R_k(\mathbf{s}) = (0, 0, \cdots, 0, s_0, \cdots, s_{w-k-1})$.
- Circular operation by k bits which is denoted by $\sigma^k(\mathbf{s}) = (s_{w-k}, \cdots, s_{w-1}, s_0, \cdots, s_{w-k-1})$.
- LR shift combination operation which is denoted by $LR_{k,l}(\mathbf{s}) = L_k(\mathbf{s}) \oplus R_l(\mathbf{s})$.

The word oriented LFSRs which have been used in Sosemanuk [4] or SNOW [5] stream ciphers have neither the TGFSR nor the σ-LFSR form. For these types of the word oriented LFSRs, calculation of $\mathbf{s}_{t+i}.\mathbf{C}_i$ is corresponding to multiplication of $\mathbf{s}_{t+i}$ by α_i in $\mathbb{F}_{2^w}$ where α_i is an element of $\mathbb{F}_{2^w}$. $\mathbf{C}_i = \mathbf{I}_{w\times w}$ and $\mathbf{C}_i = \mathbf{0}_{w\times w}$ are corresponding to $\alpha_i = 1$ and $\alpha_i = 0$, respectively.

The coefficients $\mathbf{C}_i \neq \mathbf{I}_{w\times w}, \mathbf{0}_{w\times w}$ should be chosen such that the multiplication of $\mathbf{s}_{t+i}$ by corresponding α_i can be efficiently implemented in software. For Sosemanuk and SNOW stream ciphers the length of the each word is 32 bits and the coefficients which are neither $\mathbf{0}_{32\times 32}$ nor $\mathbf{I}_{32\times 32}$, have been chosen such that the multiplication of $\mathbf{s}_{t+i}$ by $\mathbf{C}_i$ can be calculated using a shift operation of $\mathbf{s}_{t+i}$ and then XORing of the result with a word which is loaded from a lookup table. Fig. 1 shows the LFSR used in Sosemanuk stream cipher.

3 Jumping LFSR

The output bits of an LFSR are linearly dependent and cannot be directly used as a keystream. One of the ways to generate a non-linear sequence from an LFSR output is irregular clocking of the LFSR. It means the number of applied clocks to the LFSR between producing two consecutive outputs is not constant and depends on a controlling sequence. Thus it is sometimes necessary to apply more than one clock to the LFSR to produce only one output word. This property

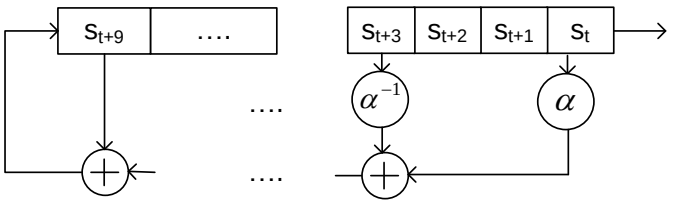

Fig. 1. The LFSR used in Sosemanuk stream cipher.

results in a reduction of the output generation rate of an irregular clocked LFSR compared with a regular one. Furthermore, irregular clocked LFSRs are vulnerable against side-channel attacks.

An efficient way to let an LFSR move to a state more than one step further without having to traverse consecutive intermediate states has been proposed in [6]. In this case, it is said that the LFSR jumps. This method is explained in the following.

Let $\mathbf{T}$ denotes the transition matrix of an LFSR (or generally a linear finite state machine). Thus, applying one regular clock to the LFSR corresponds to the multiplication of the LFSR state by $\mathbf{T}$. If a positive J is found such that $\mathbf{T}^J = \mathbf{T} + \mathbf{I}$, then it is achieved the same result if the state vector is multiplied either by $\mathbf{T}^J$ or by $\mathbf{T} + \mathbf{I}$. J is called the jump index of $\mathbf{T}$.

Multiplication of the LFSR state vector by $\mathbf{T}+\mathbf{I}$ is corresponding to XORing the LFSR state vector with the resultant state from applying one regular clock to the LFSR. Multiplication of the LFSR state vector by $\mathbf{T}^J$ is corresponding to applying J regular clocks to the LFSR. Since $\mathbf{T}^J = \mathbf{T} + \mathbf{I}$, the results of the corresponding transformations are the same. Thus, when according to the controlling sequence, it is necessary that the LFSR irregularly clocks (jumps), the transformation corresponding to $\mathbf{T} + \mathbf{I}$ can be applied to the LFSR. In this case, the LFSR jumps to a state J steps further without having to traverse consecutive intermediate states. Fig. 2 and Fig. 3 show the applying $\mathbf{T}$ and $\mathbf{T} + \mathbf{I}$ to an LFSR, respectively.

Fig. 4 shows a jumping LFSR which is controlled by a controlling sequence. $\mathbf{cs} = cs_0, cs_1, \cdots$ is the binary controlling sequence. For $cs_t = 1$ ($cs_t = 0$) the feedbacks which have been shown above the registers are connected (disconnected) and the LFSR jumps (regularly clocks).

If $f(x)$ is the characteristic polynomial of $\mathbf{T}$ and the jump index of $\mathbf{T}$ is J, it can be shown that $f(x)$ should satisfy $f(x)|x^J + x + 1$. For a primitive $f(x)$ of degree n, the jump index always exists and $n \leq J < 2^n - 1$. To determine the period of a jumping LFSR output, all the states for the jump controlling sequence should be considered. Among them, the length of the shortest jump controlling sequence which results in the LFSR state repetition, determines the period of

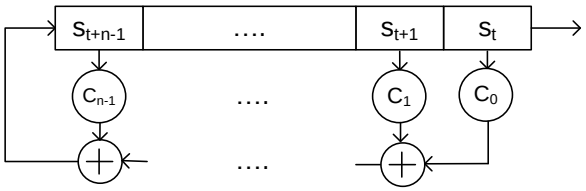

Fig. 2. An LFSR in the regular clocking case.

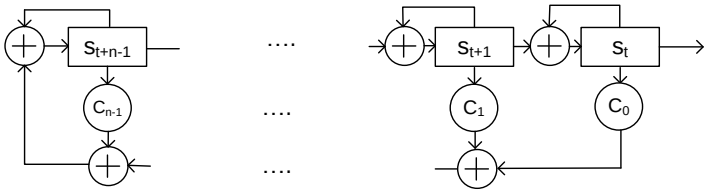

Fig. 3. An LFSR in the jumping case.

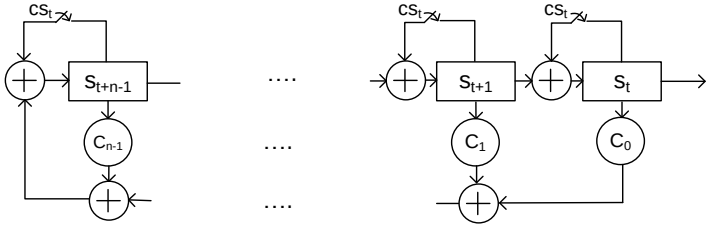

Fig. 4. A jumping LFSR controlled by the controlling sequence cs.

the jumping LFSR output. Equivalently the output period of a jumping LFSR with primitive characteristic polynomial is p words where p is the minimum value for $a+b$ $(a+b>0)$ which satisfies $a+bJ=0 \ mod \ 2^{w\times n}-1$. In these $a+b$ consecutive clocks the LFSR clocks regularly a times and jumps b times.

In [2], by applying a simple algebraic trick, a method has been proposed to find the bit oriented LFSRs of degree n with known jump index which cannot be extended to the word oriented LFSRs. In this method J is chosen to be $2^{\frac{n}{2}}-j$ where j is a small positive number. So the jumping LFSR period will be approximately $2^{\frac{n}{2}}$. These LFSRs have been used in Mickey stream ciphers family [3].

In [13], the word oriented jumping LFSRs have been proposed where σ-LFSRs were used as the jumping LFSRs. This method can be extended to the other types of word oriented LFSRs which were stated in Section 2.

If the σ-LFSR has the maximum period $2^{n\times w}-1$, there exist a jump index J such that $\mathbf{T}^J=\mathbf{T}+\mathbf{I}$. In a jumping σ-LFSR, according to the controlling sequence, in the jumping case the transition matrix changes from $\mathbf{T}$ to $\mathbf{T}+\mathbf{I}$.

Finding the jump index is the final problem which has been mentioned in [13]. Let $f(x)$ be the characteristic polynomial of the LFSR transition matrix. Then the jump index can be computed using DLP (discrete logarithm problem) $J=log_x(x+1)$ in the finite field $\mathbb{F}_{2^{n\times w}}$. First Pohlig-Hellman method [10] is used to reduce the DLP in $\mathbb{F}^*$ to the DLP in groups of prime group of order p which p divides $|\mathbb{F}^*|=2^{n\times w}-1$. For each p, the jump index module p is calculated using

$$(x+1)^{\frac{|\mathbb{F}^*|}{p}}=(x^{\frac{|\mathbb{F}^*|}{p}})^{(J \ mod \ p)} \ mod \ f(x) \tag{9}$$

The above equation for all factors of $|\mathbb{F}^*|$ should be solved. Then using the Chinese Reminder Theorem, J can be calculated. The Pollard's Rho method [11] can be used to solve the equations of the form (9). The complexity of the calculation is approximately of order $\sqrt{p}$.

The period of the jumping LFSR output is an important parameter which [11] has proposed no method to calculate it. To avoid the short period of the jumping LFSR output, in [13] it has been just suggested to choose the jumping LFSRs which for them the characteristic polynomial of $\mathbf{T}$ $(f(x))$ and $\mathbf{T}+\mathbf{I}$ $(f(x+1))$ are primitive. The polynomial $f(x+1)$ is called the dual of the polynomial $f(x)$

It should be noted that the primitivity of $f(x)$ and $f(x+1)$ are necessary but not sufficient conditions to have an output sequence with long period. It means for some values of the jump index, there exist some states for the controlling sequence that result short period for the σ-LFSR. For example assume a σ-LFSR of length $n\times w$ bits (a σ-LFSR consists of n words of length w bits) and $f(x)$ and $f(x+1)$ are primitive for this σ-LFSR. Let the jump index for the σ-LFSR is equal to $2^{n\times w-1}-1$. The controlling sequence is shown with $\mathbf{cs}$. Then $cs_i=0$ and $cs_i=1$ cause to $\mathbf{T}$ and $\mathbf{T}+\mathbf{I}$ are applied to the σ-LFSR state, respectively. For a controlling sequence equals to $(\overline{110})=110110110\cdots$, state of the σ-LFSR repeats after each 3 consecutive clocks or equivalently the period of the σ-LFSR would be at most 3 words.

Furthermore, if $|\mathbb{F}^*|$ has a large factor (e.g. $p > 2^{128}$), then finding the jump index would be infeasible. For these cases the jump index value is unknown and so obviously the LFSR period cannot be calculated. Since now, it has not been proposed any structure based on the word-oriented jumping LFSRs with determined period. In Section 4 some structures with the determined lower bound of the period based on more than one jumping LFSR will be proposed. For some of the proposed structures, knowledge about the jump index value is not necessary and so these structures are suitable for cases where finding the jump index would be infeasible for them.

4 New Structures

In this section some new structures are introduced which can be used as primitives to design new stream ciphers. The new structures are based on more than one LFSR and their periods can be determined. In the new structures at each clock only one LFSR jumps and the other LFSRs regularly clock. So all of them have resistance against the side channel attacks if constructions of the used LFSRs are the same. Resistance against other attacks is not investigated because it depends on the properties of the stream ciphers using the mentioned structures.

The LFSRs used in the new structures can be word oriented LFSRs as well as bit oriented ones. Fig 5 shows the first structure. ($\overline{cs_t}$) denotes the complement of the bit cs_t.

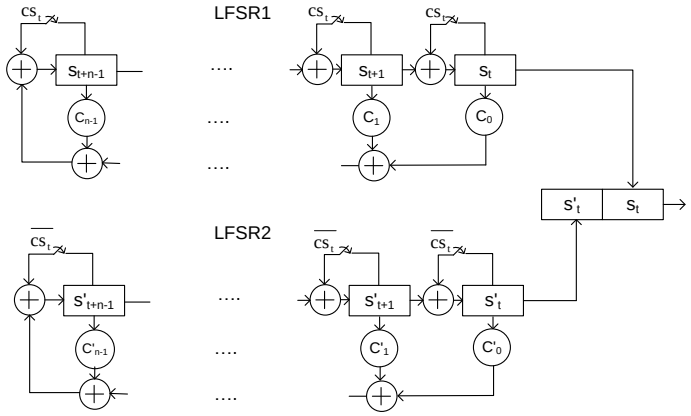

Fig. 5. New structure with two jumping LFSRs.

Each LFSR produces one word per clock and so in each clock two words are outputs of the structure. In this structure when one of the LFSRs regularly

clocks (the one with $cs_t = 0$) the other LFSR jumps (the one with $cs_t = 1$). Now we want to find the lower bound of the structure output period. The transition matrix of LFSR1 and LFSR2 are denoted by $\mathbf{T}_1$ and $\mathbf{T}_2$, respectively. The characteristic polynomial of $\mathbf{T}_1$ and $\mathbf{T}_2$ are $f_1(x)$ and $f_2(x)$, respectively. Jump index of LFSR1 and LFSR2 are denoted by J_1 and J_2, respectively. Theorem 3 determines the lower bound of structure output period.

Theorem 3. *In the structure shown in Fig. 5, if the state of LFSR1 and LFSR2 are nonzero and $f_1(x)$ and $f_2(x)$ are primitive and the greatest common divisor (GCD) of $J_1 \times J_2 - 1$ and $2^{w\times n} - 1$ is d, then the lower bound of the structure output period is $2 \times \frac{2^{n\times w}-1}{d}$ words.*

Proof. Assume that in a period of the structure output, the controlling sequence has a_1 bits with value 1 and a_2 bits with value 0. So in these $a_1 + a_2$ consecutive clocks, LFSR1 regularly clocks a_2 times and jumps a_1 times. Obviously in these clocks, LFSR2 regularly clocks a_1 times and jumps a_2 times. Applying these $a_1 + a_2$ clocks, results in the repetition of the LFSR1 state and also LFSR2 state. Thus, respect to the primitivity of $f_1(x)$ for the repetition of LFSR1 state, we have the following relation:

$$a_2 + a_1 \times J_1 = 0 \ mod \ 2^{w\times n} - 1 \tag{10}$$

Similarly for LFSR2 we have:

$$a_1 + a_2 \times J_2 = 0 \ mod \ 2^{w\times n} - 1 \tag{11}$$

Multiplying the relation (10) by J_2 and subtracting relation (11) from the result we have:

$$a_1 \times (J_1 \times J_2 - 1) = 0 \ mod \ 2^{w\times n} - 1 \tag{12}$$

Similarly, by multiplying the relation (11) by J_1 and subtracting the relation (10) from the result we have:

$$a_2 \times (J_1 \times J_2 - 1) = 0 \ mod \ 2^{w\times n} - 1 \tag{13}$$

Adding the relations (12) and (13) results in

$$(a_1 + a_2) \times (J_1 \times J_2 - 1) = 0 \ mod \ 2^{w\times n} - 1 \tag{14}$$

If the GCD of $J_1 \times J_2 - 1$ and $2^{w\times n} - 1$ is d, then $a_1 + a_2$ ($a_1 + a_2 > 0$) should be a positive multiple of $\frac{2^{w\times n}-1}{d}$. So the minimum value for $a_1 + a_2$ is $\frac{2^{w\times n}-1}{d}$ and in each clock two words are produced. So the lower bound of the output period is $2 \times \frac{2^{w\times n}-1}{d}$ words.

Example 4. Consider LFSR1 as a TGFSR with $n = 5$, $w = 32$, $m = 4$, $\mathbf{a} = 0xDFB93BEF$ and LFSR2 as a TGFSR with $n = 5$, $w = 32$, $m = 2$, $\mathbf{a} = 0xEEE6BBFD$. The characteristic polynomial of LFSR1 and LFSR2 and also

their duals are stated in the Appendix. The factorization of $2^{n\times w}-1 = 2^{160}-1$ is:

$$3\times 11\times 17\times 25\times 31\times 41\times 257\times 61681\times 65537\times 414721\times 4278255361\times 44479210368001 \tag{15}$$

Jump index values of LFSR1 and LFSR2 are as following:

$$\begin{aligned} J_1 &= 340282367000166625996085689103316680701, \\ J_2 &= 491651825044167539119904468755442744072168407 06 \end{aligned} \tag{16}$$

For the mentioned structure, the GCD of $J_1 \times J_2 - 1$ and $2^{w\times n}-1$ is $3\times 5\times 17\times 257\times 65537 = 2^{32}-1$. So the minimum value for the structure output period is $2\times \frac{(2^{160}-1)}{(2^{32}-1)} \simeq 2^{129}$ words. Note that the initial state of LFSR1 and LFSR2 should be nonzero.

Corollary 5. *In the structure shown in Fig 5 if the state of LFSR1 and LFSR2 are nonzero, $f_1(x)$ and $f_2(x)$ are primitive and $J_1 \times J_2 - 1$ is prime relative to $2^{w\times n}-1$, then the lower bound of the structure output period would be $2\times(2^{w\times n}-1)$ words.*

Proof. If $J_1 \times J_2 - 1$ is relatively prime to $2^{w\times n}-1$ or equivalently the GCD of $J_1 \times J_2 - 1$ and $2^{w\times n}-1$ is 1 then according to Theorem 3 the lower bound of structure output period would be $2\times(2^{w\times n}-1)$ words.

Fig. 6 shows the another structure. In this structure three jumping LFSRs are used. The transition matrix of LFSR1, LFSR2 and LFSR3 are denoted by $\mathbf{T}_1$, $\mathbf{T}_2$ and $\mathbf{T}_3$, respectively. The characteristic polynomial of $\mathbf{T}_1$, $\mathbf{T}_2$ and $\mathbf{T}_3$ are $f_1(x)$, $f_2(x)$ and $f_3(x)$, respectively. The jump index of LFSR1, LFSR2 and LFSR3 are denoted by J_1, J_2 and J_3, respectively. At each clock only one of the LFSRs jumps and the others regularly clock. It means that the controlling sequence, $\mathbf{cs}_t$, results in three dependent binary sequences, $cs1_t$, $cs2_t$ and $cs3_t$ with condition $cs1_t + cs2_t + cs3_t = 1$. For example based on the controlling sequence if the first LFSR jumps, then the second and the third LFSR regularly clock. Theorem 6 determines the period of the mentioned structure output.

Theorem 6. *In the structure shown in Fig. 6 assume that the state of LFSR1, LFSR2 and LFSR3 are nonzero and $f_1(x)$, $f_2(x)$ and $f_3(x)$ are primitive. If the GCD of $J_1\times J_2\times J_3 - J_1 - J_2 - J_3 + 2$ and $2^{w\times n}-1$ is d, then the lower bound of the structure output period would be $3\times\frac{2^{w\times n}-1}{d}$ words.*

Proof. Assume that in a period, LFSR1, LFSR2 and LFSR3 jump a_1, a_2 and a_3 times, respectively or equivalently LFSR1, LFSR2 and LFSR3 regularly clock $a_2 + a_3$, $a_1 + a_3$ and $a_1 + a_2$ times, respectively. Applying these $a_1 + a_2 + a_3$ clocks to the structure results in the repetition of the states of LFSR1, LFSR2 and LFSR3. So respect to the primitivity of $f_1(x)$, $f_2(x)$ and $f_3(x)$ we have the following three equations:

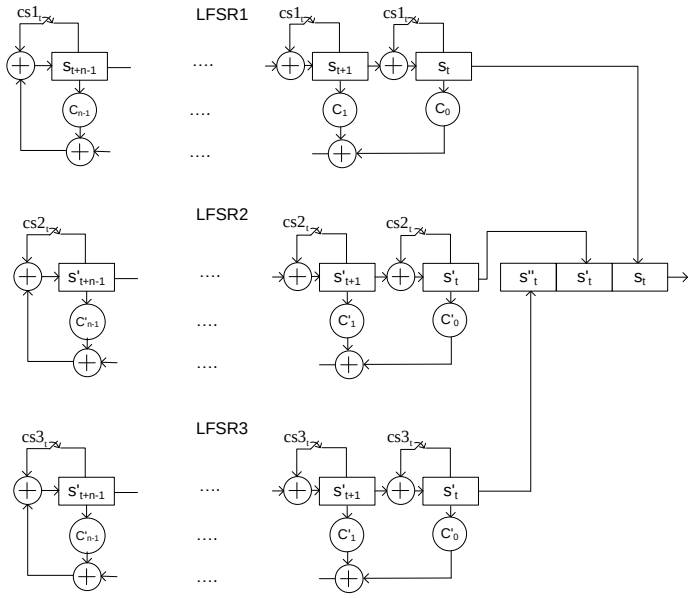

Fig. 6. New structure with three jumping LFSRs.

$$\begin{aligned} a_1 \times J_1 + a_2 + a_3 &= 0 \; mod \; 2^{w\times n} - 1 \\ a_1 + a_2 \times J_2 + a_3 &= 0 \; mod \; 2^{w\times n} - 1 \\ a_1 + a_2 + a_3 \times J_3 &= 0 \; mod \; 2^{w\times n} - 1 \end{aligned} \tag{17}$$

Combining the above equations we have the following equation:

$$a_i \times [J_1 \times J_2 \times J_3 - J_1 - J_2 - J_3 + 2] = 0 \; mod \; 2^{w\times n} - 1, i = 1, 2, 3 \tag{18}$$

So we have:

$$(a_1 + a_2 + a_3) \times [J_1 \times J_2 \times J_3 - J_1 - J_2 - J_3 + 2] = 0 \; mod \; 2^{w\times n} - 1 \tag{19}$$

If the GCD of $J_1 \times J_2 \times J_3 - J_1 - J_2 - J_3 + 2$ and $2^{w\times n} - 1$ is d, then $a_1 + a_2 + a_3 (a_1 + a_2 + a_3 > 0)$ should be a positive multiple of $\frac{2^{w\times n}-1}{d}$. So the minimum value for $a_1 + a_2 + a_3$ is $\frac{2^{w\times n}-1}{d}$ and in each clock three words are produced. So the lower bound of the structure output period is $3 \times \frac{2^{w\times n}-1}{d}$ words.

Corollary 7. *In the structure shown in Fig. 6, if the state of LFSR1, LFSR2 and LFSR3 are nonzero, $f_1(x)$, $f_2(x)$ and $f_3(x)$ are primitive and $J_1 \times J_2 \times J_3 - J_1 - J_2 - J_3 + 2$ is prime relative to $2^{w\times n} - 1$, then the lower bound of the structure output period would be $3 \times (2^{w\times n} - 1)$ words.*

Proof. The proof is similar to Corollary 5.

It should be noted that in the mentioned structures, it is only sufficient to find the value of the jump indices modulo the factors of $2^{w\times n} - 1$. In Section 3 the method of finding the jump index module p_i was proposed (p_i is a prime factor of $2^{w\times n} - 1$). In the mentioned method, equation 9 should be solved for all prime factors of $2^{w\times n} - 1$. The complexity of solving the equation using Pollard's Rho method is of order $\sqrt{p}_i$. Thus if $2^{w\times n} - 1$ has a factor with large value (e.g. $p_i > 2^{128}$) then finding the jump index would be infeasible.

Example 8. Consider a structure with parameters $n = 32$ and $w = 16$. For this structure, there is a prime number of length 206 bits which is a factor of $2^{w\times n} - 1$.

Now some structures will be proposed where it is unnecessary to calculate the jump index of the structures LFSRs to find the lower bound of structures output period. Suppose that in the first structure of this section LFSR1 and LFSR2 are the same and the transition matrix of that is denoted by $\mathbf{T}$. The characteristic polynomial of $\mathbf{T}$ and the jump index of this LFSR are denoted by $f(x)$ and J, respectively. The following corollary determines the period of such a structure.

Corollary 9. *In the structure shown in Fig. 5 with two same LFSRs, if the state of the LFSRs are nonzero and $f(x)$ is primitive and the GCD of $J^2 - 1$ and $2^{w\times n} - 1$ is d then the period of the structure output would be $2 \times \frac{2^{w\times n}-1}{d}$ words.*

Proof. Theorem 6 with $J_1 = J_2$ results in Corollary 9.

According to Corollary 9 the period of the structure output would be $2 \times \frac{2^{w \times n}-1}{d}$ words, if the GCD of $(J-1) \times (J+1)$ and $2^{w \times n} - 1$ is d. Now for any factor p of $2^{w \times n} - 1$ the question is that $J-1$ or $J+1$ has this factor or not. To check this, we have Theorem 10.

Theorem 10. *Suppose that the characteristic polynomial of an LFSR, $f(x)$ is primitive and jump index of this LFSR is denoted by J. If p is a factor of $2^{w \times n}-1$ and p satisfies $[x^i(x+1)]^{\frac{2^{w \times n}-1}{p}} \neq 1 \ mod\ f(x)$, then p would not be a factor of $J+i$ (i is an integer number).*

Proof. Suppose that p is a factor of $2^{w \times n} - 1$ and $J + i$, then we have

$$[x^i(x+1)]^{\frac{2^{w \times n}-1}{p}} = [(x^i x^J)]^{\frac{2^{w \times n}-1}{p}} = [x^{kp}]^{\frac{2^{w \times n}-1}{p}} = [x^{2^{w \times n}-1}]^k = 1 \ \ mod\ f(x) \tag{20}$$

which leads to a contradiction. Thus the theorem was proved.

We emphasize that the order of calculation of $[x^i(x+1)]^{\frac{2^{w \times n}-1}{p}} mod\ f(x)$ is $w \times n$ which is feasible for usual LFSRs.

Corollary 11. *In the structure shown in Fig. 5 with two same LFSRs, suppose that the characteristic polynomial of the LFSR, $f(x)$ is primitive and jump index of this LFSR is denoted by J. Assume $2^{n \times w} - 1 = p_1^{\alpha_1} \times p_2^{\alpha_2} \times \cdots \times p_m^{\alpha_m}$ and $\mathcal{I}$ is the set that $p_k \in \mathcal{I} \Leftrightarrow [x(x+1)]^{\frac{2^{w \times n}-1}{p_k}} = 1\ mod\ f(x)$ or $[x^{-1}(x+1)]^{\frac{2^{w \times n}-1}{p_k}} = 1\ mod\ f(x)$. The lower bound of the structure output period is $2 \times \frac{2^{w \times n}-1}{\prod_{k \in \mathcal{I}} p_k^{\alpha_k}}$ words.*

Proof. Using Corollary 9 and Theorem 10 (with $i = \pm 1$), this corollary is proved.

Example 12. Consider the structure shown in Fig 5 where LFSR1 and LFSR2 are the same σ-LFSRs with parameters $n = 32$ and $w = 16$. For these σ-LFSRs $\mathbf{C}_0$ corresponds to $LR_{1,1}$, $\mathbf{C}_1$ corresponds to Λ_{0x06E9} and $\mathbf{C}_i$ is an all zero matrix for $i = 2, 3, \cdots, 31$. The characteristic polynomial of LFSR1 and its dual are stated in the Appendix. $2^{w \times n} - 1$ is factored as following: $3 \times 5 \times 17 \times 257 \times$
$641 \times 65537 \times 274177 \times$
$6700417 \times 67280421310721 \times 1238926361552897 \times$
$59649589127497217 \times 5704689200685129054721 \times$
$934616397153579777691635581996068965840512375416381885802803 21$
By using Corollary 11 for the mentioned structure, the GCD of $J+1$ and $2^{512}-1$ is 3. It means $p = 3$ is the only factor of $2^{512}-1$ which satisfies $[x(x+1)]^{\frac{2^{w \times n}-1}{p}} = 1$. The greatest common divisor of $J-1$ and $2^{w \times n} - 1$ is 1 or equivalently all of the factors of $2^{512} - 1$ satisfy $[x^{-1}(x+1)]^{\frac{2^{w \times n}-1}{p}} \neq 1$. Thus the structure output period would be $2 \times \frac{2^{512}-1}{3}$ words.

In the above example, it should be noted that the period was determined without knowledge of the LFSR jump index value. It is noted that if two LFSRs

are not the same, the characteristic polynomials of the LFSRs are different and we cannot use Corollary 11 to calculate the lower bound of output period.

For the structure shown in Fig. 6 we have the similar results. Suppose that in this structure LFSR1, LFSR2 and LFSR3 are the same and the transition matrix of that is denoted by $\mathbf{T}$. The characteristic polynomial of $\mathbf{T}$ and the jump index of this LFSR are denoted by $f(x)$ and J, respectively. The following corollary determines the period of such a structure output.

Corollary 13. *In the structure shown in Fig. 6, if the LFSRs states are nonzero, $f(x)$ is primitive and the GCD of $J^3 - 3 \times J + 2$ and $2^{w\times n} - 1$ is d then the period of structure output would be $3 \times \frac{2^{w\times n}-1}{d}$ words.*

Proof. Corollary 7 with $J_1 = J_2 = J_3$ results in Corollary 11.

The structures introduced in this section can be used as primitives to design pseudo-random generators. A simple method to design a pseudo-random generator by utilizing one of the proposed structures is the use of combining function. As an example, the combining function can be a combination of XOR, rotation and multiplication modulo 2^w which is applied on output words of the structure LFSRs (A memory can be included in the combining function). Jump controlling sequence for the proposed structure can be produced from combination of one (or more) LFSRs word(s) and output sequence.

5 Conclusion

In this paper, using the jumping LFSRs, some structures were proposed where the lower bound of their output period can be calculated and resist against side-channel attacks. In these structures more than one LFSR were used which in each clock only one of them jumped and the others were regularly clocked. In the first structure only two n words LFSRs were used. Let jump indices of two LFSRs are J_1 and J_2. It was proved that the lower bound period of this structure is $2 \times \frac{2^{w\times n}-1}{d}$ words where d is the GCD of $J_1 \times J_2 - 1$ and $2^{w\times n} - 1$.
In the other structure, we used two same LFSRs. In this structure, it is not necessary to calculate the value of jump index (it is note that the jump index calculation is not simple for some LFSRs). For this structure it is sufficient to check that $J^2 - 1$ is relatively prime to the prime factors of $2^{w\times n} - 1$ and the output period is $2 \times \frac{2^{w\times n}-1}{d}$ where d is the GCD of $J^2 - 1$ and $2^{w\times n} - 1$. Moreover, we can extend the proposed structures to the cases with more than two LFSRs. We can use these proposed structures instead of the used LFSRs in the some current stream ciphers or in the design of the future stream ciphers (especially the stream ciphers with determined period).

References

1. Ecrypt, eSTREAM: ECRYPT Stream Cipher Project,, 2004-2008. http://www.ecrypt.eu.org/stream/.
2. S.H. Babbage and M.W. Dodd. Finding characteristic polynomials with jump indices, 2006. http://eprint.iacr.org/2006/010.
3. S.H. Babbage and M.W. Dodd. The stream cipher Mickey-128 2.0, 2007. ECRYPT Stream Cipher Project.
4. C. Berbain, O. Billet, and et al. Sosemanuk, a fast software-oriented stream cipher, 2007. ECRYPT Stream Cipher Project.
5. P. Ekdahl and T. Johansson. A new version of the stream cipher Snow. In *SAC' 02*, volume 2595, pages 47–61. Springer-Verlag, 2003.
6. C.J.A. Jansen. Stream cipher design: Make your lfsrs jump!, 2004. ECRYPT SASC workshop.
7. C.J.A. Jansen, T. Helleseth, and A Kholosha. Cascade jump controlled sequence generator and Pomaranch stream cipher (version 3), 2007. eSTREAM, ECRYPT Stream Cipher Project.
8. M. Matsumoto and Y. Kurita. Twisted GFSR Generators. *ACM transactions on modeling and simulation (TOMACS)*, 2(3):179–194, 1992.
9. H. Niederreiter. The multiple-recursive matrix method for pseudorandom number generation. *Finite Fields and Their Applications*, 1(1):3–30, 1995.
10. M.E. Pohlig, S.C.and Hellman. An improved algorithm for computing logarithms over gf(p) and its cryptographic significance. *IEEE Transactions on Information Theory*, 24:106–110, 1978.
11. J.M. Pollard. Monte carlo methods for index computation (mod p). *Mathematics of Computation*, 32(143):918–924, 1978.
12. Rueppel R.A. *Analysis and Design of Stream Ciphers*. Springer-Verlag, 1986.
13. G. Zeng, Y. Yang, W. Han, and S. Fan. Word oriented cascade jump σlfsr. In *AAECC ' 09*, volume 5527, pages 127–136. Springer-Verlag, 2009.

A Characteristic Polynomial of LFSRs in Examples

The characteristic polynomial of an LFSR is stated by $f(x) = \sum_{i \in I} x^i$. For LFSR1 in the Example 1, the set I is as following:

$I_{LFSR1} = \{$
$0, 4, 5, 12, 13, 14, 15, 16, 21, 25, 26, 29, 31, 33, 34, 35, 42, 44, 45, 46, 47, 50, 53, 54,$
$55, 56, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 73, 75, 76, 77, 78, 79, 80, 84, 92, 94, 96,$
$97, 98, 99, 101, 104, 108, 111, 113, 115, 116, 120, 121, 124, 125, 127, 130, 131, 135, 139,$
$143, 144, 147, 149, 151, 152, 153, 154, 155, 160\}$

and for its dual the set I is:

$I_{dualLFSR1} = \{$
$0, 1, 3, 4, 6, 8, 9, 10, 11, 12, 14, 21, 23, 24, 30, 31, 32, 33, 34, 36, 39, 40, 42, 45, 49,$
$50, 54, 55, 56, 57, 58, 62, 65, 66, 70, 76, 77, 78, 79, 81, 82, 84, 87, 88, 89, 91, 92, 95, 97,$
$98, 100, 108, 109, 113, 118, 119, 120, 121, 122, 123, 124, 126, 127, 129, 130, 131, 134, 135,$
$139, 140, 141, 142, 143, 145, 147, 150, 151, 155, 160\}$

For LFSR2 which was mentioned in the Example 1, the set I is as following:

$I_{dualLFSR1} = \{$
$0, 2, 4, 5, 8, 12, 13, 16, 21, 22, 24, 25, 28, 29, 30, 32, 34, 36, 40, 44, 45, 47, 48, 49,$
$54, 55, 57, 60, 62, 63, 64, 67, 71, 72, 73, 76, 78, 80, 81, 83, 85, 87, 88, 90, 96, 98, 100, 102,$
$103, 104, 105, 106, 107, 108, 111, 112, 113, 115, 118, 119, 120, 121, 124, 126, 129, 131, 132,$
$133, 134, 135, 137, 142, 143, 145, 146, 149, 152, 155, 160\}$

and for its dual the set I is:

$I_{dualLFSR2} = \{$
$0, 1, 4, 8, 9, 10, 12, 13, 14, 16, 18, 19, 20, 24, 31, 37, 38, 42, 46, 47, 48, 52, 53, 54, 55, 57,$
$58, 59, 60, 61, 62, 63, 65, 67, 70, 71, 73, 77, 78, 79, 82, 85, 89, 94, 99, 100, 101, 103, 109, 111,$
$112, 117, 118, 119, 121, 122, 126, 128, 129, 130, 132, 136, 137, 138, 141, 143, 144, 145, 147,$
$148, 149, 153, 154, 155, 160\}$

For the LFSR used in the Example 3, the set I is as following:

$I_{LFSR3} = \{$
$0, 35, 66, 70, 97, 99, 101, 103, 130, 132, 134, 161, 163, 167, 198, 225, 227, 231, 256, 258, 260,$
$291, 295, 322, 324, 326, 355, 357, 384, 388, 419, 448, 450, 481, 512\}$

and for its dual the set I is:

$I_{dualLFSR3} = \{$
$0, 1, 3, 32, 33, 34, 35, 66, 67, 70, 98, 99, 130, 131, 162, 163, 192, 197, 199, 228, 229, 230, 231,$
$263, 294, 295, 321, 323, 324, 325, 326, 352, 353, 354, 355, 356, 357, 387, 388, 418, 419, 448, 449, 450,$
$, 480, 481, 512\}$