Journal of Computing and Security

July 2015, Volume 2, Number 3 (pp. 175–184)

http://www.jcomsec.org

Primitives Based on Jumping LFSRs with Determined Period

Mahdi Sajadieh [a], Arash Mirzaei [b], Mohammad Dakhilalian [b]

[a] *Department of Electrical Engineering, Islamic Azad University, Isfahan (Khorasgan) Branch, Iran.*

[b] *Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran.*

ARTICLE INFO.

Article history:
Received: 06 June 2015
Revised: 15 February 2016
Accepted: 25 April 2016
Published Online: 24 August 2016

Keywords:
Stream Ciphers, LFSR, Jump Index, Period

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 utilize LFSRs in construction of stream ciphers. Jumping is one of these methods which is used in some stream ciphers like Mickey and Pomaranch but the period for this method cannot be generally determined. In this paper, using the jumping LFSRs, some new primitives are proposed. According to the properties of these new primitives, a lower bound for their period can be computed. In some of the proposed primitives, this lower bound can be determined without the knowledge of the jump index. These primitives are applicable when the calculation of the jump index is infeasible. The proposed primitives can be used as building blocks to design the software oriented and the hardware oriented stream ciphers.

1 Introduction

In cryptography, stream ciphers are symmetric key ciphers employed to provide confidentiality for communication channels. These types of ciphers produce a stream of pseudo-random bits, which is called 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 bit oriented linear feedback shift registers (LFSRs) because of their hardware implementation speed. In addition, the LFSR output can have maximal period and good statistical properties [1]. Since 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 were introduced and many stream ciphers have recently been proposed based on them [2].

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 timing, power and other side-channel attacks [3]. In addition, the bit (or word) generation rate of these LFSRs is less than that of regularly clocked ones. To address these flaws, in [3] a method called jumping was suggested for 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. The number of the intermediate states is called the jump index of LFSR.

* Corresponding author.
Email addresses: m.sajadieh@khuisf.ac.ir (M. Sajadieh), arash_mirzaei@ec.iut.ac.ir (A. Mirzaei), mdalian@cc.iut.ac.ir (M. Dakhilalian)

ISSN: 2322-4460

The most important disadvantage of 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 [4, 5], 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 bit oriented LFSRs and moreover, its resistance against side-channel attacks has not been proved.

In [6], jumping method was extended to a type of word oriented LFSRs called σ-LFSRs. The idea of jumping method can easily be extended to the other types of word oriented LFSRs. For the jumping word oriented LFSRs, depending on the LFSR length, there exist cases where finding the jump index is plausible [6]. For these cases, determining the period is not possible. Moreover, even when the jump index value is known, no method has been suggested to determine the exact value or even a lower bound for the period of jumping for the word oriented LFSRs.

In this paper, using more than one jumping LFSR, some primitives are proposed for which the lower bound of the period can be determined if the jump index value is known. Furthermore, some primitives with the determined lower bound of period are proposed for cases which the jump index cannot be calculated for. All of the proposed primitives in this paper have resistance against side-channel attacks and can be used as building blocks to design word oriented and 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 primitives against other types of attacks depends on the properties of stream ciphers use the new primitives as building blocks.

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 represented in Section 2. We propose our new primitives and their properties in Section 4.

2 Brief Description of LFSRs

In this section bit oriented LFSRs and some types of word oriented LFSRs are described all of which can be used in the new primitives proposed in Section 4. A word oriented LFSR is a linear shift register based on the linear recurrence Equation (1) [7]

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

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

The state of the nw-bit shift register at time t is denoted by $\mathbf{S}_t = (\mathbf{s}_t, \mathbf{s}_{t+1}, \cdots, \mathbf{s}_{t+n-1})$ and the nonzero 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 Equation (1), the transition matrix of an LFSR which transforms $\mathbf{S}_t$ to $\mathbf{S}_{t+1}$ is as follows

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

If the characteristic polynomial of the LFSR is primitive, its output sequence attains the maximal period. Theorem 1 determines the necessary and sufficient conditions for a word oriented LFSR to have the maximal period.

Theorem 1. *[7]*

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* $l = 0, 1, \cdots, n-1$, *and*

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

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 $\boldsymbol{F}(x)$ *(*$|\boldsymbol{F}(x)|$*) is a primitive polynomial of degree* wn *over* $\mathbb{F}_2$.

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

$$s_{t+n} = s_{t+n-1}c_{n-1} + s_{t+n-2}c_{n-2} + \cdots + s_tc_0, \quad t = 0, 1, \cdots \tag{4}$$

where each c_i equals 0 or 1.

For bit oriented LFSRs the transition matrix is an $n \times n$ binary matrix of the form

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

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

Different types of 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 remainder of this section, some types of word oriented LFSRs are described. It should be noted that all types of word oriented LFSRs can be used in the new primitives of Section 4.

2.1 TGFSR

TGFSR (Twisted Generalized Feedback Shift Register) [8] is a word oriented linear feedback shift register based on the linear recurrence Equation (1)

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

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 Equation (1), for 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 output of TGFSR attains the maximal period. It means the minimum value for k which satisfies $\mathbf{s}_{t+k} = \mathbf{s}_t$, for $t = 0, 1, \cdots$ is $2^{nw} - 1$. 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 $2^{nw} - 1$ words if and only if $\phi_{\boldsymbol{A}}(t^n + t^m)$ is a primitive polynomial of degree nw.

$\mathbf{A}$ 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

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

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

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

where $\mathbf{s}$ is a binary vector with content $(s_0, s_1, \cdots, s_{w-1})$ and $\mathbf{a}$ is the binary vector $(a_0, a_1, \cdots, a_{w-1})$ and also $\mathbf{s} \ggg 1$ shows one bit circular shift of vector $\mathbf{s}$ to the right.

2.2 σ-LFSR

Another type of word oriented LFSRs is σ-LFSR. σ-LFSRs are linear feedback shift registers based on the linear recurrence Equation (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 right shift 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 $LR_{k,l}(\mathbf{s}) = L_k(\mathbf{s}) \oplus R_l(\mathbf{s})$.

2.3 Other Word Oriented LFSRs

Word oriented LFSRs which have been used in Sosemanuk [9] or SNOW [10] stream ciphers have neither the TGFSR nor the σ-LFSR form. For these types of word oriented LFSRs, calculation of $\mathbf{s}_{t+i}.\mathbf{C}_i$ corresponds 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 $\mathbf{C}_i = \mathbf{0}_{w\times w}$ correspond 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 each word is 32 bits and the coefficients which are neither $\mathbf{0}_{32\times32}$ nor $\mathbf{I}_{32\times32}$, 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 the result with a word which is loaded from a lookup table.

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 results in a reduction of the output generation rate of an irregular clocked LFSR compared with a regular one.

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 [11]. In this case, it is said that the LFSR jumps. This method is explained as follows.

Let $\mathbf{T}$ denote the transition matrix of an LFSR (or generally a linear finite state machine). Thus, applying one regular clock to the LFSR corresponds to multiplication of the LFSR state by $\mathbf{T}$. If a positive J is found such that $\mathbf{T}^J = \mathbf{T} + \mathbf{I}$, then the same result is obtained 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 $\mathbf{T} + \mathbf{I}$ corresponds 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 $\mathbf{T}^J$ corresponds 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. Figure 1 and Figure 2 show the application of $\mathbf{T}$ and $\mathbf{T} + \mathbf{I}$ to an LFSR, respectively.

Figure 3 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 are above the registers are connected (disconnected) and the LFSR jumps (regularly clocks).

Figure 1. An LFSR in the Regular Clocking Case[3]

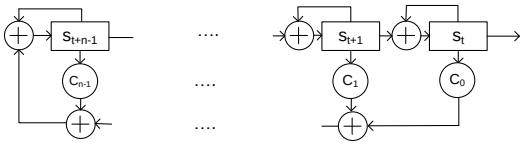

Figure 2. An LFSR in the Jumping Case[3]

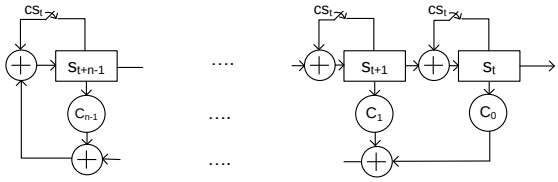

Figure 3. A Jumping LFSR Controlled by the Controlling Sequence cs.

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 of 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 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 $a + bJ = 0 \ mod \ 2^{nw} - 1$. In these $a + b$ consecutive clocks, the LFSR clocks regularly a times and jumps b times.

In [4], by applying a simple algebraic trick, a method has been proposed to find bit oriented LFSRs of degree n with known jump index which cannot be extended to 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 is approximately $2^{\frac{n}{2}}$. These LFSRs have been used in Mickey stream ciphers family [5].

In [6], word oriented jumping LFSRs were proposed

where σ-LFSRs were used as the jumping LFSRs. This method can be extended to the other types of word oriented LFSRs stated in Section 2.

If the σ-LFSR has the maximum period $2^{nw}-1$, a jump index J exists 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 [6]. 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 modulo $2^{nw}-1$. At first Pohlig-Hellman method [12] is used to reduce the DLP in $\mathbb{F}^*$ to the DLP in groups of prime group of order p which p divides $|\mathbb{F}^*| = 2^{nw}-1$. For each p, the jump index module p is calculated using

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

The above equation for all factors of $|\mathbb{F}^*|$ should be solved. Then, using the Chinese Remainder Theorem, J can be calculated. The Pollard's Rho method [13] 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 but no method has been proposed in [13] to calculate it. In order to avoid jumping LFSR output with short period, [6] has suggested that jumping LFSRs be chosen in such a way that 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, some states exist for the controlling sequence that result in short period for the σ-LFSR. For example, assume a σ-LFSR of length nw 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 be equal to $2^{nw-1}-1$. The controlling sequence is shown with $\mathbf{cs}$. Then $cs_i = 0$ and $cs_i = 1$ cause $\mathbf{T}$ and $\mathbf{T}+\mathbf{I}$ to be 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 is unknown and so obviously the LFSR period cannot be calculated. So far, no structure based on the word-oriented jumping LFSRs with determined period has been proposed. In Section 4 some primitives with the determined lower bound of the period based on more than one jumping LFSR are proposed. For some of the proposed primitives, knowledge about the jump index value is not necessary and so these primitives are suitable for the cases for which finding the jump index are infeasible.

4 Some New Primitives Based on Several Jumping LFSRs

In this section, some new primitives are introduced which can be used as building blocks to design new stream ciphers. The new primitives are based on more than one LFSR. In the new primitives at each clock only one LFSR jumps and the other LFSRs regularly clock. So, if constructions of the used LFSRs are similar (*e.g.* all used LFSRs are TGFSRs with the same size), the number of operations per clock does not change with the value of the controlling sequence. This is why all of the new primitives are resistant against side channel attacks. It should be noted that a jumping LFSR does not have this property. In other words, the number of operations per clock for a jumping LFSR changes with the value of the controlling sequence. Resistance against other attacks is not investigated because it depends on the properties of the stream ciphers using the mentioned primitives. Also for these primitives, the lower bound of the period can be determined.

In Section 4.1, two new primitives called J2L and J3L are presented for which 2 LFSRs and 3 LFSRs are used, respectively. For these primitives, the lower bound of the output period can be determined if the jump index of each LFSR is known. Also in Section 4.2 two new primitives called J2SL and J3SL are proposed which are special cases of J2L and J3L, respectively. For these primitives, to determine the lower bound of output period, finding the exact value of jump index for LFSRs is not required.

4.1 J2L and J3L

Figure 4 shows the first primitive. Hereinafter, this primitive is called J2L. ($\overline{cs_t}$) denotes the complement of the bit cs_t.

Each LFSR produces one word per clock and so in each clock two words are the outputs of J2L. In J2L, when one of the LFSRs regularly clocks (the one with $cs_t = 0$) the other LFSR jumps (the one with $cs_t = 1$). Now the lower bound of J2L output period is found. 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

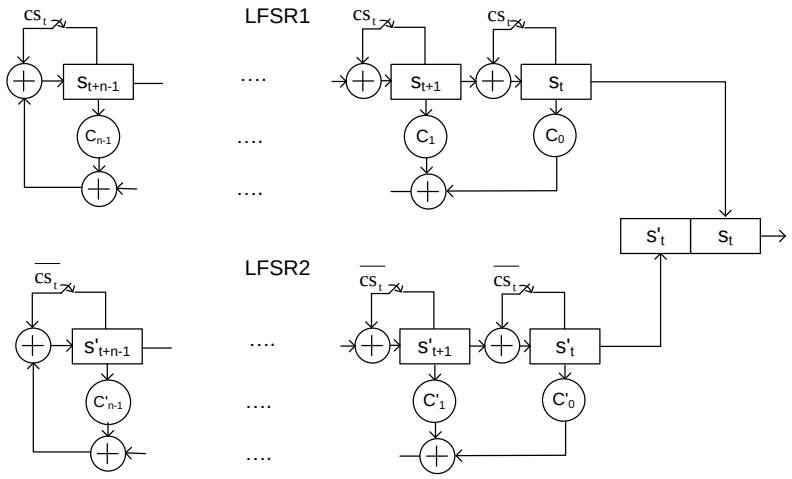

Figure 4. The New Structure With Two Jumping LFSRs.

$f_2(x)$ and also the jump index of LFSR1 and LFSR2 are denoted by J_1 and J_2, respectively. Theorem 3 determines the lower bound of J2L output period.

Theorem 3. *In J2L, 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_1J_2-1 and $2^{nw}-1$ is d, then the lower bound of the output period is $2*\frac{2^{nw}-1}{d}$ words.*

Proof. Assume that in one period of J2L 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. It is obvious that 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, according to the primitivity of $f_1(x)$ for the repetition of LFSR1 state, the following equation is obtained:

$$a_2 + a_1J_1 = 0 \ mod\ 2^{nw}-1 \tag{10}$$

Similarly for LFSR2:

$$a_1 + a_2J_2 = 0 \ mod\ 2^{nw}-1 \tag{11}$$

Equation (12) is resulted by multiplying Equation (10) by J_2 and subtracting Equation (11) from the result:

$$a_1(J_1J_2-1) = 0 \ mod\ 2^{nw}-1 \tag{12}$$

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

$$a_2(J_1J_2-1) = 0 \ mod\ 2^{nw}-1 \tag{13}$$

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

$$(a_1+a_2)(J_1J_2-1) = 0 \ mod\ 2^{nw}-1 \tag{14}$$

If the GCD of J_1J_2-1 and $2^{nw}-1$ is d, then a_1+a_2 ($a_1+a_2>0$) should be a positive multiple of $\frac{2^{nw}-1}{d}$. So, the minimum value for a_1+a_2 is $\frac{2^{nw}-1}{d}$ and in each clock two words are produced. Thus, the lower bound of the output period is $2*\frac{2^{nw}-1}{d}$ words. □

Example 1. 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 $2^{nw}-1=2^{160}-1$ is:

$$2^{160}-1 = 3*11*17*25*31*41*257*61681* 65537*414721*4278255361*44479210368001 \tag{15}$$

Jump index values of LFSR1 and LFSR2 are as following:

$$J_1 = 340282367000166625996085689103316680701, \tag{16}$$

$$J_2 = 491651825044167539119904468755442744072168406706 \tag{17}$$

For mentioned J2L, the GCD of J_1J_2-1 and $2^{nw}-1$ is $3*5*17*257*65537=2^{32}-1$. So the minimum value for the output period is $2*\frac{(2^{160}-1)}{(2^{32}-1)}\simeq 2^{129}$ words.

Corollary 1. *For J2L, if the state of LFSR1 and LFSR2 are nonzero, $f_1(x)$ and $f_2(x)$ are primitive and J_1J_2-1 is co-prime to $2^{nw}-1$, then the lower bound of the output period will be $2*(2^{nw}-1)$ words.*

Proof. If J_1J_2-1 is relatively prime to $2^{nw}-1$ or equivalently the GCD of J_1J_2-1 and $2^{nw}-1$ is 1

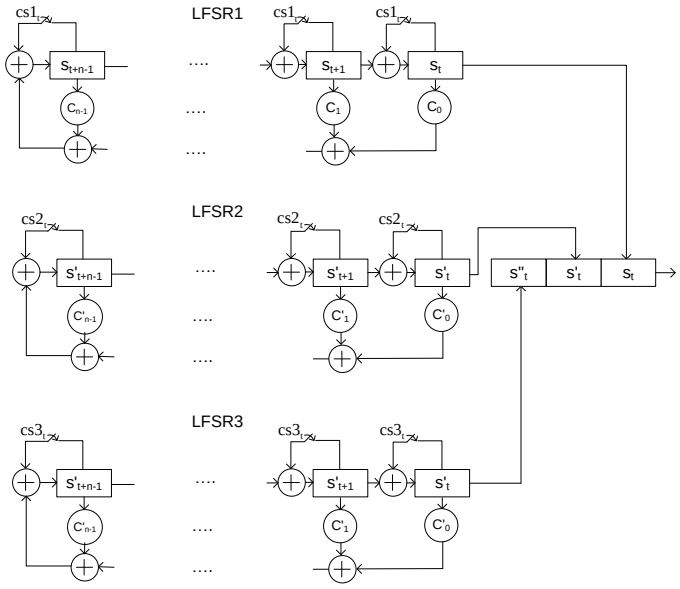

Figure 5. The New Structure With Three Jumping LFSRs.

then according to Theorem 3 the lower bound of J2L output period will be $2 * (2^{nw} - 1)$ words. □

Figure 5 shows another new primitive. This primitive consists of three jumping LFSRs and hereinafter is called J3L. 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 $f_3(x)$ and the jump indexes 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 are regularly clocked. It means that the controlling sequence, $\mathbf{cs}_t$, results in three dependent binary sequences, $cs1_t$, $cs2_t$ and $cs3_t$ with the 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 4 determines the period of J3L output.

Theorem 4. *For J3L assume that the states of LFSR1, LFSR2 and LFSR3 are nonzero and $f_1(x)$, $f_2(x)$ and $f_3(x)$ are primitive. If the GCD of $J_1J_2J_3 - J_1 - J_2 - J_3 + 2$ and $2^{nw} - 1$ is d, then the lower bound of the output period will be $3 * \frac{2^{nw}-1}{d}$ words.*

Proof. Assume that in a period, LFSR1, LFSR2 and LFSR3 jump a_1, a_2 and a_3 times, 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 J3L results in the repetition of the states of LFSR1, LFSR2 and LFSR3. So, according to the primitivity of $f_1(x)$, $f_2(x)$ and $f_3(x)$ the following three equations are concluded:

$$\begin{aligned} a_1J_1 + a_2 + a_3 &= 0 \ mod\ 2^{nw} - 1 \\ a_1 + a_2J_2 + a_3 &= 0 \ mod\ 2^{nw} - 1 \\ a_1 + a_2 + a_3J_3 &= 0 \ mod\ 2^{nw} - 1 \end{aligned} \tag{18}$$

Combining the above equations, the following equation is obtained:

$$a_i[J_1J_2J_3 - J_1 - J_2 - J_3 + 2] = 0 \ mod\ 2^{nw} - 1, i = 1, 2, 3 \tag{19}$$

So:

$$(a_1 + a_2 + a_3)[J_1J_2J_3 - J_1 - J_2 - J_3 + 2] = 0 \ mod\ 2^{nw} - 1 \tag{20}$$

If the GCD of $J_1J_2J_3 - J_1 - J_2 - J_3 + 2$ and $2^{nw} - 1$ is d, then $a_1 + a_2 + a_3 (a_1 + a_2 + a_3 > 0)$ should be a positive multiple of $\frac{2^{nw}-1}{d}$. So, the minimum value for $a_1 + a_2 + a_3$ is $\frac{2^{nw}-1}{d}$ and in each clock three words are produced. Thus, the lower bound of J3L output period is $3 * \frac{2^{nw}-1}{d}$ words. □

Corollary 2. *For J3L, if the state of LFSR1, LFSR2 and LFSR3 are nonzero, $f_1(x)$, $f_2(x)$ and $f_3(x)$ are primitive and $J_1J_2J_3 - J_1 - J_2 - J_3 + 2$ is co-prime to $2^{nw} - 1$, then the lower bound of the output period will be $3 * (2^{nw} - 1)$ words.*

Proof. The proof is similar to Corollary 1. □

4.2 J2SL and J3SL

It should be noted that in J2L and J3L, it is only sufficient to find the value of the jump indices modulo the factors of $2^{nw}-1$. In Section 2 the method of finding the jump index module p_i was proposed (p_i is a prime factor of $2^{nw}-1$). In the mentioned method, Equation (9) should be solved for all prime factors of $2^{nw}-1$. The complexity of solving the equations in Equation (9) using Pollard's Rho method is of order $\sqrt{p_i}$. Thus if $2^{nw}-1$ has a factor with large value (*e.g.* $p_i > 2^{128}$) then finding the jump index will be infeasible.

Example 2. Consider a word LFSR with parameters $n=32$ and $w=16$. For this LFSR, there is a prime number of length 206 bits which is a factor of $2^{nw}-1$ and finding its jump index is infeasible.

Now, some primitives are proposed for which it is unnecessary to calculate the jump index of their LFSRs to find the lower bound of their output period. Suppose that for J2L, LFSR1 and LFSR2 are the same and their transition matrix is denoted by $\mathbf{T}$. Hereinafter, it is called J2SL. 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 J2SL.

Corollary 3. *For J2SL, if the state of the LFSRs are nonzero and $f(x)$ is primitive and the GCD of J^2-1 and $2^{nw}-1$ is d, then the output period will be $2*\frac{2^{nw}-1}{d}$ words.*

Proof. Theorem 4 with $J_1 = J_2$ results in Corollary 3. □

According to Corollary 3, the output period of J2SL will be $2*\frac{2^{nw}-1}{d}$ words, if the GCD of $(J-1)(J+1)$ and $2^{nw}-1$ is d. Now, for any factor p of $2^{nw}-1$ the question is whether $J-1$ or $J+1$ has this factor or not. To check this, Theorem 5 is presented.

Theorem 5. *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 $2^{nw}-1$ and p satisfies $[x^i(x+1)]^{\frac{2^{nw}-1}{p}} \neq 1 \ mod\ f(x)$, then p will not be a factor of $J+i$ (i is an integer number).*

Proof. Suppose that p is a factor of $2^{nw}-1$ and $J+i$, then

$$[x^i(x+1)]^{\frac{2^{nw}-1}{p}} = [(x^i x^J)]^{\frac{2^{nw}-1}{p}} = [x^{kp}]^{\frac{2^{nw}-1}{p}} = [x^{2^{nw}-1}]^k = 1 \ \ mod\ f(x) \tag{21}$$

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

It is emphasized that the order of calculation of $[x^i(x+1)]^{\frac{2^{nw}-1}{p}} mod\ f(x)$ is nw which is feasible for usual LFSRs.

Corollary 4. *For J2SL, suppose that the characteristic polynomial of the LFSR, $f(x)$, is primitive and jump index of this LFSR is denoted by J. Assume $2^{nw}-1 = p_1^{\alpha_1}p_2^{\alpha_2}\cdots p_m^{\alpha_m}$ and $\mathcal{I}$ is the set that $p_k \in \mathcal{I} \Leftrightarrow [x(x+1)]^{\frac{2^{nw}-1}{p_k}} = 1 \ mod\ f(x)$ or $[x^{-1}(x+1)]^{\frac{2^{nw}-1}{p_k}} = 1 \ mod\ f(x)$. The lower bound of the output period is $2*\prod_{k\in\mathcal{I}} p_k^{\alpha_k}$ words.*

Proof. Using Corollary 3 and Theorem 5 (with $i = \pm 1$), this corollary is proved. □

Example 3. For J2SL, consider that 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. $2^{nw}-1$ is factored as following:

$$\begin{aligned}&3*5*17*257*641*65537*274177*\\ &6700417*67280421310721*1238926361552897*\\ &59649589127497217*5704689200685129054721*\\ &9346163971535797776916355819960689658405...\\ &1237541638188580280321\end{aligned} \tag{22}$$

By using Corollary 4 for J2SL, 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 $[x(x+1)]^{\frac{2^{nw}-1}{p}} = 1$. The greatest common divisor of $J-1$ and $2^{nw}-1$ is 1 or equivalently all of the factors of $2^{512}-1$ satisfy $[x^{-1}(x+1)]^{\frac{2^{nw}-1}{p}} \neq 1$. Thus the output period would be $2*\frac{2^{512}-1}{3}$ words.

In the above example, it should be noted that the period was determined without the 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 Corollary 4 cannot be used to calculate the lower bound of output period.

When LFSRs used in J3L are the same, the primitive is called J3SL. J3SL has similar results as J2SL. The transition matrix of LFSRs is denoted by $\mathbf{T}$ and 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 J3SL output.

Corollary 5. *For J3SL, if the LFSRs states are nonzero, $f(x)$ is primitive and the GCD of J^3-3J+2*

and $2^{nw}-1$ *is* d *then the output period will be* $3*\frac{2^{nw}-1}{d}$ *words.*

Proof. Corollary 2 with $J_1 = J_2 = J_3$ results in Corollary 4. □

5 Conclusion

In this paper, using the jumping LFSRs, some primitives were proposed for which the lower bound of output period can be calculated. In these primitives more than one LFSR are used, in each clock only one of which jumps and the others are regularly clocked. In J2L only two $n-$word LFSRs are used. Let jump indices of two LFSRs be J_1 and J_2. It was proved that the lower bound of the period of this primitive is $2*\frac{2^{nw}-1}{d}$ words where d is the GCD of J_1J_2-1 and $2^{nw}-1$.

In J2SL, we used two similar LFSRs. In this primitive, it is not necessary to calculate the jump index in order to compute the period (note that the jump index calculation is not simple for some LFSRs). For this primitive it is sufficient to check that J^2-1 is coprime to the prime factors of $2^{nw}-1$ and the output period is $2*\frac{2^{nw}-1}{d}$ where d is the GCD of J^2-1 and $2^{nw}-1$. Moreover, we can extend the proposed primitives to the cases with more than two LFSRs. We can use these proposed primitives to replace the LFSRs used in some current stream ciphers or to design new stream ciphers (especially the ones with determined period).

References

[1] Rueppel R.A. *Analysis and Design of Stream Ciphers*. Springer-Verlag, 1986.

[2] Ecrypt, eSTREAM: ECRYPT Stream Cipher Project, 2004-2008. URL `http://www.ecrypt.eu.org/stream/`.

[3] C. Jansen, T. Helleseth, and A. Kholosha. *New Stream Cipher Designs: The eSTREAM Finalists*, chapter Cascade Jump Controlled Sequence Generator and Pomaranch Stream Cipher, pages 224–243. Springer Berlin Heidelberg, 2008.

[4] S. Babbage and M. Dodd. Finding Characteristic Polynomials with Jump Indices, 2006. URL `http://eprint.iacr.org/2006/010`.

[5] S. Babbage and M. Dodd. *New Stream Cipher Designs: The eSTREAM Finalists*, chapter The MICKEY Stream Ciphers. Springer Berlin Heidelberg, 2008.

[6] G. Zeng, Y. Yang, W. Han, and S. Fan. Word Oriented Cascade Jump σ-LFSR. In *AAECC '09*, volume 5527, pages 127–136. Springer-Verlag, LNCS, 2009.

[7] H. Niederreiter. The multiple-recursive matrix method for pseudorandom number generation. *Finite Fields and Their Applications*, 1(1):3–30, 1995.

[8] M. Matsumoto and Y. Kurita. Twisted GFSR Generators. *ACM transactions on modeling and simulation (TOMACS)*, 2(3):179–194, 1992.

[9] C. Berbain, O. Billet, Anne Canteaut, N. Courtois, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Pornin, and H. Sibert. *New Stream Cipher Designs: The eSTREAM Finalists*, chapter Sosemanuk, a Fast Software-Oriented Stream Cipher, pages 98–118. Springer Berlin Heidelberg, 2008.

[10] P. Ekdahl and T. Johansson. A New Version of the Stream Cipher SNOW. In *SAC' 02, LNCS*, volume 2595, pages 47–61. Springer-Verlag, 2003.

[11] C. Helleseth, T. Helleseth, C. Jansen, and E. Kholosha. Cascade jump controlled sequence generator. In *In: Symmetric Key Encryption Workshop, Workshop Record, ECRYPT Network of Excellence in Cryptology*, 2005.

[12] 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.

[13] J.M. Pollard. Monte carlo methods for index computation (mod p). *Mathematics of Computation*, 32(143):918–924, 1978.

A Characteristic Polynomial of LFSRs in the 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:

$I_{dualLFSR2} = \{$
$0, 2, 4, 5, 8, 12, 13, 16, 21, 22, 24, 25, 28, 29, 30, 32, 34,$
$36, 40, 44, 45, 47, 48, 49, 54, 55, 57, 60, 62, 63, 64, 67, 71,$
$72, 73, 76, 78, 80, 81, 83, 85, 87, 88, 90, 96, 98, 100, 102,$
$103, 104, 105, 106, 107, 108, 111, 112, 113, 115, 118, 119,$
$120, 121, 124, 126, 129, 131, 132, 133, 134, 135, 137, 142,$
$143, 145, 146, 149, 152, 155, 160\}$

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:

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

Mahdi Sajadieh received the B.Sc., M.Sc. and Phd degrees in Communication Engineering from Isfahan University of Technology (IUT), Isfahan, Iran, in 2004, 2007 and 2012, respectively. He joined Islamic Azad University, Isfahan (Khorasgan) Branch in 2011 and at present time is an Assistant Professor in Electrical Engineering Department. His research interests include cryptography and channel coding.

Arash Mirzaei received the B.Sc. and M.Sc. degrees in Control and Communication Engineering from Isfahan University of Technology (IUT), Isfahan, Iran, in 2007 and 2009, respectively. His research interests include cryptography and data security.

Mohammad Dakhilalian received the B.Sc. and Ph.D. degrees in Electrical Engineering from Isfahan University of Technology (IUT) in 1989 and 1998 respectively and M.Sc. degree in Electrical Engineering from Tarbiat Modarres University in 1993. He was an Assistant Professor of Faculty of Information & Communication Technology, Ministry of ICT, Tehran, Iran in 1999-2001. He joined IUT in 2001 and at present time is an Associate Professor in Electrical and Computer Engineering Department. His current research interests are Cryptography and Data Security.