Construction of Maximum period Linear Feedback Shift Registers (LFSR)
(Primitive polynomials and linear recurring relations)

Babacar Alassane Ndaw.* Djiby Sow.† Mamadou Sanghare.‡

July 7, 2015

Abstract

Feedback Shift Register (FSR) is generally the basic element of pseudo random generators used to generate cryptographic channel or set of sequences for encryption keys. This type of generator is widely used in stream cipher and communication systems such as C.D.M.A (Code Division Multiple Access) , mobile communication systems, ranging and navigating systems, spread spectrum communication systems.

The objective of the present paper is to propose a method for determining linear recurring sequences generating linear feedback shift register (LFSR) from primitive polynomials (and vice-versa). The linear recurring sequences facilitate the construction of maximum length LFSR. It also insists, in the last part, on the cryptographic security of LFSR and indicates some open problems in the area of nonlinear feedback shift registers (NLFSR) based pseudo random generators.

1 Introduction

Contrary to blocks ciphers algorithms, stream cipher algorithms operate on every unit of the plaintext (encryption of a bit/character at time, bit by bit or character by character) ; the bits are encrypted individually. They are generally faster than blocks cipher algorithms, and have less complex circuits. [19][29][33][58][63][70][71][73][74][75][80][82][85][86][87][89][94][96].

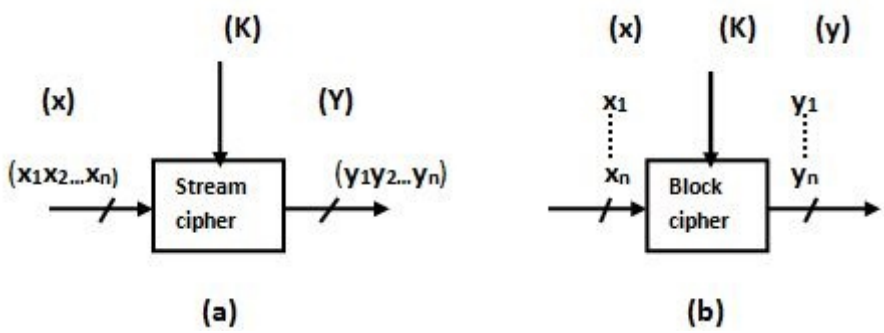


Figure 1: (a) Stream cipher and (b) Block cipher

The stream cipher based on a key generator which generates a bit stream (Key Stream) i.e. a key sequence $K = (k_1, k_2, ..., k_n)$ that combined (XOR function) to bits of plaintext $X = (x_1, x_2, ..., x_n)$ provides ciphertext $Y = (y_1, y_2, ..., y_n)$.

*Head of Central Technical Service of Cipher and Information Security Systems, Presidence of Republic of Senegal, graduated of the Centre of Superior Cryptographic Studies of Paris.

†Professor of Mathematics and Information Security, University of Cheikh Anta Diop, Dakar-Senegal.

‡Professor of Mathematics and Information Security, University of Cheikh Anta Diop, Dakar-Senegal.

- encryption equation $y_i = E_{ki}(x_i) = x_i \oplus k_i$
- decryption equation $x_i = D_{ki}(y_i) = y_i \oplus k_i$

(E_{ki}=encryption function and D_{ki}=decryption function)

Encryption is reciprocal: we encrypt as we decrypt.The key stream generator may be regarded as a finite state machine.

- An error in y_i affects only one bit x_i
- The loss or addition of a bit y_i affects all following bits (x_i) after decryption
- if $k_i = 0, \forall i, X = Y$
- if the following key (k_i) is infinite and completely random, one obtain crypto system key-a-time crypto system (One-Time-Pad) also called VERNAM cipher, name of its inventor Gilbert VERNAM (1917) [29][63][70][74][75][77][93][94][98]which is unconditionally secure against a cipher text only attack, the cryptogram contributes no information about the plaintext.

Generally, the stream cipher is based on the same principle as the "One-Time-Pad" with the only difference that it requires a real random sequence which cannot be produced unless you know the whole sequence.

In default, pseudo-random key sequences generated by a pseudo-random generator is therefore used[37][38] [49][82][99]. Good pseudo-random sequence is one for which, knowing a portion of the sequence, it is extremely difficult in practice to determine the rest of the sequence[15]. A classic method for generating a pseudo random sequence[44][45]is to use a feedback shift register[cf.paragraph 2].

The bit stream (Stream Key) or sequences of keys generated by the key stream generator constitutes a cryptographic chain.

The stream cipher is classified into two (2) categories:

- Synchronous stream cipher:

  In synchronous stream cipher, the flow of bits of the key stream is generated independently both of the bits of the plaintext and the bit stream of the cipher text.The sender and receiver must be synchronized i.e. use the key stream and be in the same condition that the decryption can be done. If there's a loss or addition of bits, the decryption fails. However, changing a bit in the transmission does not interfere in the decryption of the following bits.

  Examples[29][86]: Output Feedback Mode (OFB) for block cipher systems and CTR mode(Counter Mode) are examples of synchronous stream cipher.

- Self synchronizing stream cipher or asynchronous stream cipher:

  In self-synchronizing stream cipher (or asynchronous stream cipher) each bit of the stream generated by the key generator is a function of a fixed number of bits of the preceding cryptogram. In this method, the generators are synchronized automatically.

If some encrypted bits are lost or added in the cipher text,self synchronization is always possible. However, the system is subject to error propagation, and similarly, a modification of the cipher text by the descriptor can lead in an incorrect decryption of several bits.

Example[86]:Cipher Feedback Mode (CFB) transforms block cipher into self asynchronous stream cipher.

The two (2) methods of stream cipher mentioned above are described in detail, in [19][29][39] [70][75][77][79] [82][86].

2 LFSR(Linear Feedback Shift Register)

An example of this type of generator is the FSR (Feedback Shift Register=Shift Registers+a feedback function) .

2.1 Definition:[9][23][44][72][86]

- A flip-flop (position on a delay line or other memory device) is an electronic device capable of storing binary information (bits 0 and 1).
- A shift register of length (n) consists of n-flops interconnected such that the binary state of the memory cell of rank (i) is transmitted to the memory cell of rank $(i+1)$ when a clock signal is applied to the all flip-flops. Each flip-flop may be seen as a stage of the register. The binary information of the last stage is always accessible physically.

A shift register is then constituted of:

- An input which, in shift mode, will advance the bit of a flip-flop to a next flip-flop.
- (n) flops constituting the register stages.
- And an output.

Example of a shift-register of 11 stages:

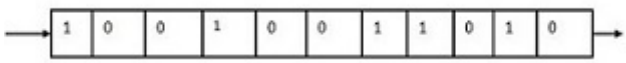


Figure 2: 11 bits LFSR

Most pseudo-random generators are constructed using feedback shift registers (Example:eSTREAM project running from 2004 to 2008 to choose new standard stream ciphers: Sosemanuk, Grain, Mickey, Trivium) . [1][7][8][9][44][66][67][81]

The Feedback Shift Registers constitute the base of pseudo-random generators used for generation of encryption key. This type of generator is largely used in stream cipher.

A Feedback Shift Register (FSR) of size (n) is an automate constructed by a boolean function (f) (ref: Definition 2.2) and a function (F) both with n variables over a field $GF(p)$ such that
$F = \{0,1\}^n \rightarrow \{0,1\}^n$ (often $p = 2$ for binary field where $p = 2^w$ for some extension field of the binary field) .

$$F(x_1, x_2, ..., x_n) = (x_2, x_3, ..., x_n, f(x_1, x_2, ..., x_n)) \tag{1}$$

- F which is the function of the next state, gives the new state of the FSR from the prior state;
- and (f) which is the feedback function calculates the $n-th$ term of the next state;
- If $(x_1, x_2, ..., x_n)$ is the initial state then the application of (f) and (F) give the state sequence:

$$\begin{array}{rl}
F(x_1, x_2, ..., x_n) = & (x_2, x_3, ..., x_n, x_{n+1}); x_{n+1} = f(x_1, x_2, ..., x_n) \\
F(x_2, x_3, ..., x_{n+1}) = & (x_3, x_4, ..., x_{n+1}, x_{n+2}); x_{n+2} = f(x_2, x_3, ..., x_{n+1}) \\
F(x_3, x_4, ..., x_{n+2}) = & (x_4, x_5, ..., x_n, x_{n+1}, x_{n+2}, x_{n+3}); x_{n+3} = f(x_3, x_4, ..., x_{n+2}) \\
\vdots & \vdots
\end{array}$$

The output sequence generated by the FSR:

$$(x_i)_{i \in N} = (x_1, x_2, ..., x_n, x_{n+1}, ...) \tag{2}$$

satisfies the relation of recurrence:

$$x_{i+n} = f(x_i, x_{i+1}, x_{i+2}, ..., x_{i+n-1}) \tag{3}$$

If the feedback function (f) is linear (ref:2.2 Definition) , the FSR is called Linear Feedback Shift Registers(LFSR).Otherwise, it is called Nonlinear Feedback Shift Register (NLFSR) .[35]

2.2 Definition:[1][9][10][19][23][28][29][44][45][56][58][64][65][72][74][79][84][86][87][88][93][94][95]

- A linear feedback shift register of n bit-length (LFSR n-bits) is composed of two parts:
  - One shift register containing a sequence of n bits $(x_1, ..., x_n)$ arranged from left to right which is the initial state of the register;
  - And, a linear feedback function $f(x_1, x_2, ..., x_n)$

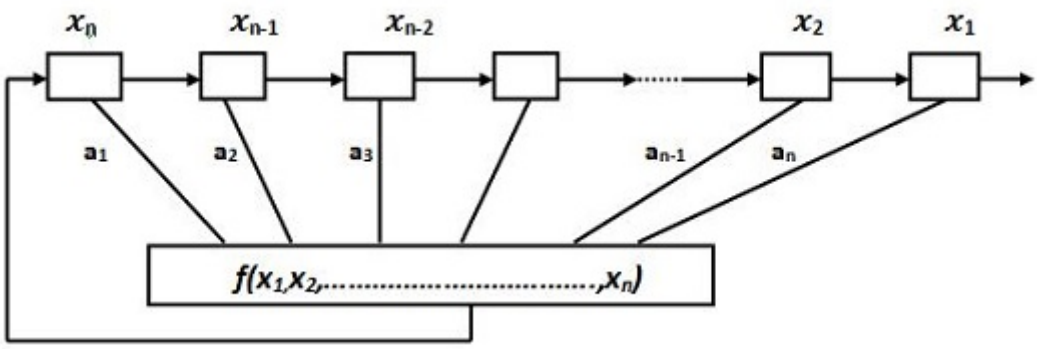


Figure 3: General Scheme of Linear Feedback Shift Register

  - The registry is called by its acronym: LFSR(Linear Feedback Shift Register).
  - At periodic intervals determined by the clock, the content of the stage (i) is transferred into the stage $(i+1)$: a bit is required at any time, and all the bits in the register are shifted forward.
  - The new left most bit is obtained from the other bits in the register with the feedback function $f(x_1, x_2, ..., x_n)$;
  - Output register is 1 bit; the sequence generated is called derivation sequence (output stream) .
  - The period of the LFSR is the length of the sequence generated before it repeats (ref. Definition 2.1).
  - The feedback function $f(x_1, x_2, ..., x_n)$ is such that:

$$\begin{aligned} f(x_1, x_2, ..., x_n) &= a_n x_1 + a_{n-1} x_2 + ... + a_1 x_n \\ &= \sum_{i=1}^{n} a_i x_{n-i+1} \end{aligned} \tag{4}$$

where $a_i = 0$ or $1, \forall i \ 1 \leq i \leq n$, and addition (XOR operation) is over GF(2).

  - $f : GF(2^n) \rightarrow GF(2)$
  - f is a boolean function of (n) variables[16][17][31][29][46][76][79];

– There are 2^{2^n} different boolean functions for (n) variables given.
– The sequence produced by the LFSR satisfy the relation of linear recurrence:

$$x_{n+j} = \sum_{i=1}^{n} a_i x_{n+j-i} \Leftrightarrow x_{n+j} = \sum_{i=0}^{n-1} a_{i+1} x_{n+j-i-1} \tag{5}$$

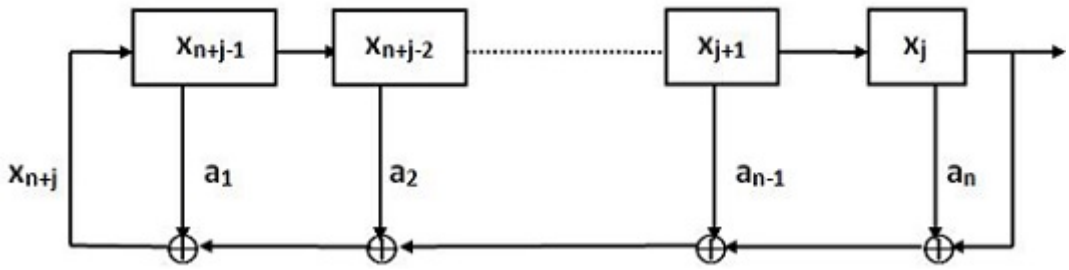


Figure 4: A scheme of LFSR

and the matrix A associated with the linear mapping is:

$$[GF(2)]^n \to [GF(2)]^n$$

$$\begin{bmatrix} x_j \\ x_{j+1} \\ \vdots \\ x_{n+j-2} \\ x_{n+j-1} \end{bmatrix} \longrightarrow \begin{bmatrix} x_{j+1} \\ x_{j+2} \\ \vdots \\ x_{n+j-1} \\ x_{n+j} \end{bmatrix} \tag{6}$$

$$A = \begin{pmatrix} 0 & 1 & 0\ldots & 0 & 0 \\ 0 & 0 & 1\ldots & 0 & 0 \\ & & & & \\ & & & & \\ 0 & 0 & 0\ldots & 0 & 1 \\ a_n & a_{n-1} & \ldots & a_2 & a_1 \end{pmatrix} \tag{7}$$

- Such the configuration, which we'll be interested, is called FIBONACCI's configuration (Fibonacci generator)" [3][29][45][65][79][86]. It is efficient in hardware as it requires only one n-bits LFSR and XOR operations although inefficient in software implementation (LFSR in mode Fibonacci or External-XOR LFSR)unlike its other counterpart called GALOIS configuration (LFSR in mode Galois or internal-XOR LFSR) "discussed in[29][45][79][86].

- The Fibonacci generator or External-XOR LFSR is based on the Fibonacci's sequences modulo the maximum value desired:

$$x_n = (x_{n-1} + x_{n-2}) \pmod m \text{ with } x_0 \text{ and } x_1 \text{ in input} \tag{8}$$

Alternatively, we can use this form called "generalized Fibonacci" recurrences to generate pseudo-random numbers[3][18][60]:

$$x_n = \pm(x_{n-s} \pm x_{n-k}) \pmod m \text{ with } x_0....x_{k-1} \text{ in input as words of w bits; generally } m = 2^w. \tag{9}$$

The quality of the generator depends of the coefficients (k),(s) which must be carefully chosen and the values used for the initial state of the generator. This generator is against very simple to implement and consumes little resources.

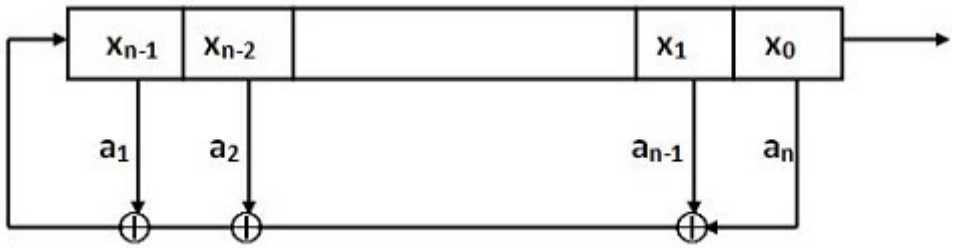


Figure 5: LFSR in mode Fibonacci

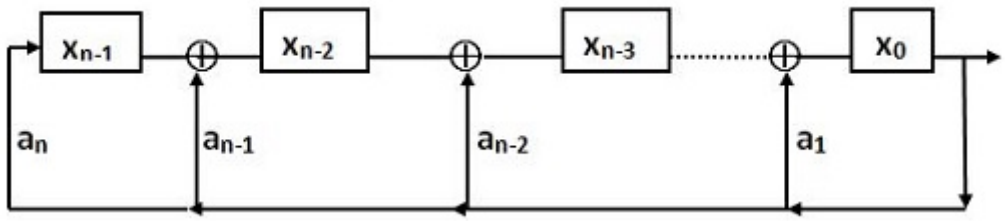


Figure 6: LFSR in mode Galois

- We must differentiate this one from the linear congruential generator[34][59][65][74][79][86][94]that produces pseudo-random sequences of the form:

$$x_n = (ax_{n-1} + b) \pmod m, \text{ where } a,\ b \text{ and } m \text{ are integers.} \tag{10}$$

  x_n is the (n)-th bit in the sequence; x_0= the seed, the period of the generator is less than m.

  If a, b, and m are carefully chosen, then the generator will be said to be "maximum period (m) (e.g.if $b \wedge m = 1, (b)$ and (m) are coprime) , if $b = 0$, the generator is said to be homogeneous congruential multiplicative.[65][34]

  The linear congruential generators are fast and require little bit operations, but it has been proved that they cannot be used in cryptography for stream cipher. Indeed, they can be predicted and therefore are decryptable[86]. It is valid for:

  - The Quadratic generators:

$$x_n = (ax_{n-1}^2 + bx_{n-1} + c) \pmod m \tag{11}$$

  - The cubic generators:

$$x_n = (ax_{n-1}^3 + bx_{n-1}^2 + cx_{n-1} + d) \pmod m \tag{12}$$

  discussed in[86] who notes that the combination of linear congruential generator providing long periods were not also proved safe cryptographically.

- The LFSR are used extensively in stream cipher because they are easily implemented in hardware as well as in software. Referring to the above definitions, it is possible to generalize the LFSR, in any finite $GF(p)$. On the software aspect, it is used finite field of the form $GF(2^n)$ with $n = 8, 32, 64, ...$

2.3 Examples

Example 1: One maximal-period n-bits LFSR

A maximal-period n-bits LFSR on $GF(2)$with maximal period $T = 2^n - 1$ is a register which can theoretically generate a pseudo-random sequence of length $T = 2^n - 1$ bits before the repetition(and not 2^n the null sequence(000...000) is not considered. The resulting output sequence is called an "m-sequence".

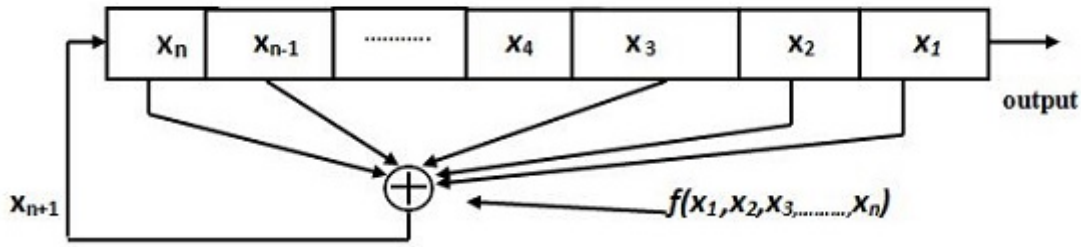
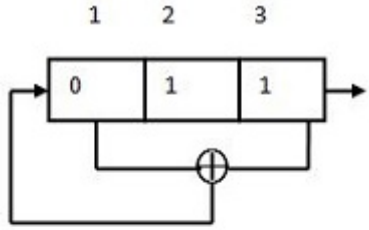


Figure 7: n-bits LFSR

Example 2: Maximal period 3-bits LFSR: There are two possible loops:

- the stages 1 and 3: $x_{n+1} = x_n + x_{n-2} \pmod 2$ (recurrence equation)
- the stages 2 and 3: $x_{n+1} = x_{n-1} + x_{n-2} \pmod 2$ (recurrence equation)
- The loop 1 and 2 is forbidden: the LFSR loop after two steps as shown in figure 10 (no maximal period)

Loop 1 and 3: Maximal period. The maximal period is: $T = 2^3 - 1 = 8 - 1 = 7$.
The recurrence equation is: $x_{n+1} = x_n + x_{n-2} \pmod 2$.



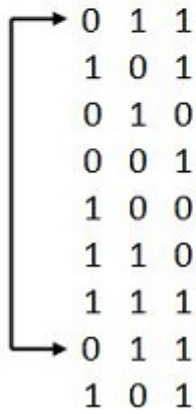


Figure 8: 3-bits LFSR

Key stream: 1101001
Loop 2 and 3: Maximal period
The recurrence equation is: $x_{n+1} = x_{n-1} + x_{n-2} \pmod 2$

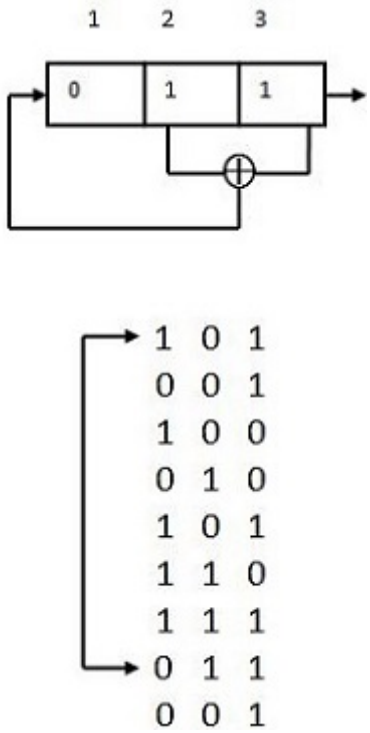


Figure 9: 3 bits LFSR

Key stream: 1100101
Loop 1 and 2:
The recurrence equation is: $x_{n+1} = x_n + x_{n-1} \pmod 2$

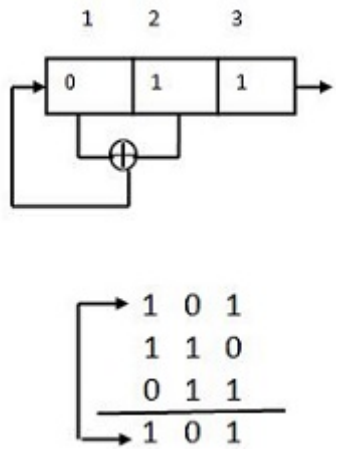


Figure 10: 3-bits LFSR

Example 3: Maximal period 4-bits LFSR
The maximal period is $T = 2^4 - 1 = 15$ with:
Loops 1 and 4: Maximal-period
The recurrence equation is: $x_{n+1} = x_n + x_{n-3} \pmod 2$

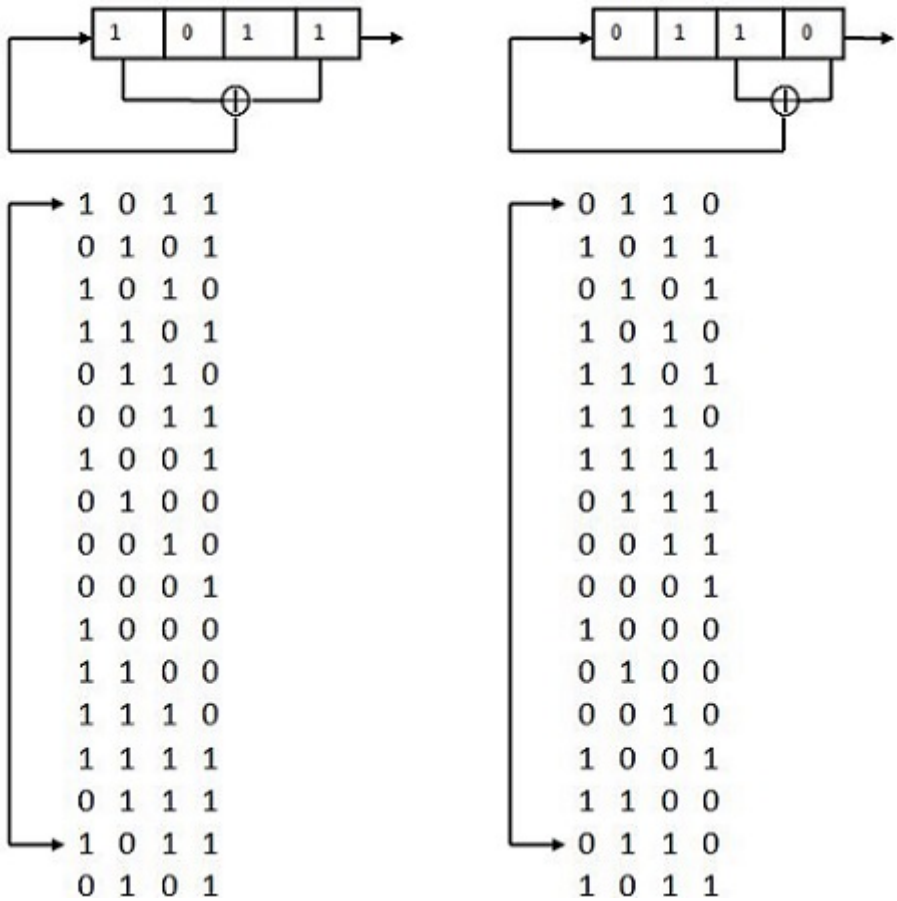


Figure 11: 4-bits LFSR

Keys stream: 110101100100011 and 011010111100010
Loop 3 and 4: Maximal period
The recurrence equation is: $x_{n+1} = x_{n-2} + x_{n-3} \pmod 2$
Example 4: maximal period 6-bits LFSR
The maximal period is obtained by: $T = 2^6 - 1 = 63$

- the loop 1 and 6: $x_{n+1} = x_n + x_{n-5}$;
- the loop 5 and 6: $x_{n+1} = x_{n-4} + x_{n-5}$;

(Loop 5 and 6: Maximal period)

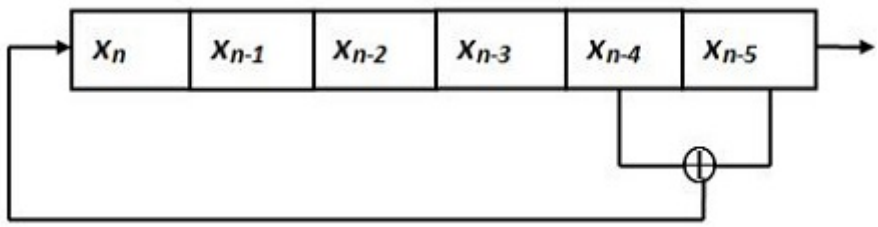


Figure 12: 6-bits LFSR

Example 5: maximal period 31-bits LFSR
The maximal period is obtained by: $T = 2^{31} - 1 = 2.147.483.647$

- the loop 3 and 31: $x_{n+1} = x_{n-2} + x_{n-30} \pmod 2$;
- the loop 28 and 31: $x_{n+1} = x_{n-27} + x_{n-30} \pmod 2$;

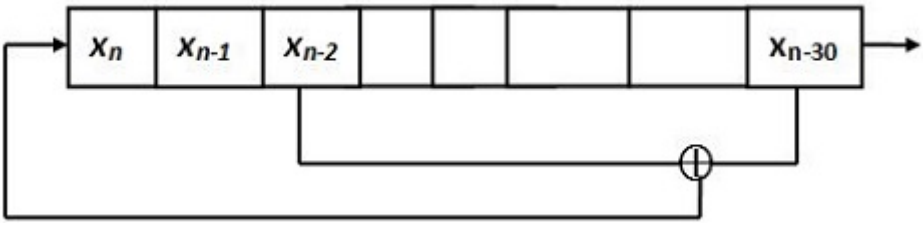


Figure 13: 31-bits LFSR

2.4 Theorem: [1][9][23][45][58][72][86][88]

For an n-bits LFSR be maximal period,it is necessary that the polynomial formed from derivation sequence must be a primitive polynomial of degree (n) in $GF(2)$.

- We thus associate with an n-bits LFSR, a primitive polynomial generator:

$$f(x) = 1 + \sum_{i=1}^{n} a_i x^i = 1 + a_1 x + a_2 x^2 + \ldots\ldots + a_n x^n \tag{13}$$

- A primitive polynomial of degree (n) is an irreducible polynomial of degree (n) which divides $(x^{2^n-1} + 1)$.

Example: the polynomial $f(x) = x^3 + x + 1$ of degree 3 is primitive over $GF(2)$, it divides $x^7 + 1$, $(T = 2^n - 1 = 2^3 - 1) \longrightarrow x^7 + 1 = (x^3 + x + 1)(x^4 + x^2 + x + 1)$.

- if any polynomial $f(x)$ associated to a LFSR is primitive over $GF(2)$ then any non-zero initial state produces a sequence of maximal period $T = 2^n - 1$.

2.5 Definition 3:

Let f be a polynomial $f(x)$ in $F_2[X]$. Its order, denoted $ord(f)$ is the smallest integer (t) such that $x^t \equiv 1 \bmod f(x)$.

2.6 Definition 4:

Let $f(x)$ be an irreducible polynomial of degree (n) in $F_2[X]$. It is primitive if its order is$(2^n - 1)$.

So, we want to build an optimal n bit LFSR (in relation to the period after production), we must ensure that the feedback polynomial chosen is of degree n and primitive.

We will be sure to obtain a maximal period, but taking the precaution of using a non-zero initial state.

Another advantage of feedback primitive polynomials is the statistical quality of sequence produced.

2.7 Definition 5:[9]

Let $f(x)$ be an irreducible polynomial over $([GF(2^n)])^*$. It is called primitive if one of its roots generates the multiplicative subgroup $[GF(2^n)]$, with $[GF(2^n)] = F_2[X]/f(x)$(polynomials reduced modulo$f(x)$). First recall that the multiplicative subgroup of a finite field is cyclic. In other words,$\forall \alpha \in [GF(2^n)]$, we have $\alpha^{2^n-1} = 1$.

Let (α) a root of (f), then we have $f(\alpha) = 0$. If α generates the multiplicative group, the elements $\alpha, \alpha^2, \alpha^3 ... \alpha^{2^n-1}$correspond to all nonzero elements of the field, and there are $2^n - 1$ distinct elements $(\alpha^{2^n-1} = 1$ since $\alpha \in ([GF(2^n)])^*$ cyclic).

2.8 Definition 6:[9][88]

In fact, we can define the order of an element α as the smallest (t) such that $\alpha^t = 1$. What we want to know is if the order of α is equal to $2^n - 1$ ($f(x) =$ primitive polynomial) or not ($f(x) =$ non-primitive polynomial); it must be remembered that for an n-bit LFSR, $(2^n - 1)$ and $f(x)$ divides $(x^{2^n-1} + 1)$.

Example: Let $f(x) = x^3 + x + 1$ irreducible, and (α) is as $f(\alpha) = 0 \Rightarrow \alpha^3 + \alpha + 1 = 0 \Rightarrow \alpha^3 = \alpha + 1$.
Let us determine the powers of α: $\alpha^1 = \alpha$
$\alpha^2 = \alpha^2$
$\alpha^3 = \alpha + 1$
$\alpha^5 = \alpha^3 + \alpha^2 = \alpha^2 + \alpha + 1$
$\alpha^6 = \alpha^3 + \alpha^2 + \alpha = \alpha^2 + 1$
$\alpha^7 = \alpha^3 + \alpha = 1 \Rightarrow \alpha^{2^3} - 1$ with $T = 7$.
We can conclude that the polynomial $f(x) = x^3 + x + 1$ is primitive.

3 Primitive feedback polynomials and linear recurrences for constructing maximal-period LFSR

With the primitive polynomial, we can identify the linear recurrence equation associated to the $n-$bits LFSR and vice versa.

Let the following register and the primitive polynomial defined in paragraph 2.4 (13):

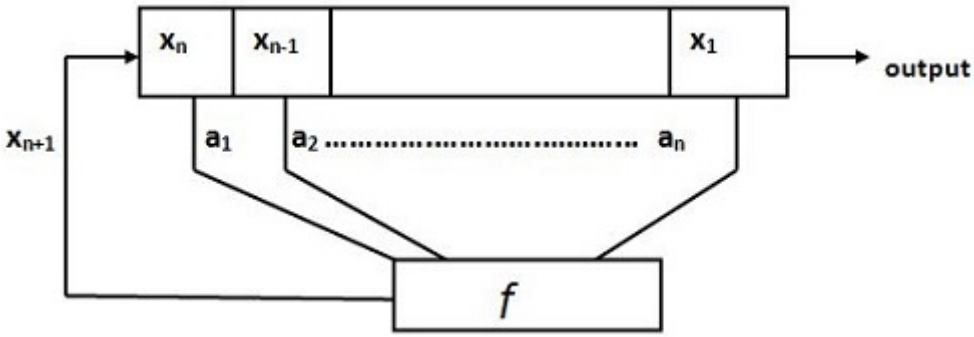


Figure 14: n-bits LFSR

Then, our linear recurrence can be expressed in this form:

$$\begin{array}{rcl} x_{n+1} & = & a_1x_n + a_2x_{n-1} + ... + a_nx_1 = \sum_{k=1}^{n-1} a_kx_{n-k+1} \\ \vdots & & \vdots \qquad\qquad\qquad \vdots \\ x_{n+i} & = & a_1x_{n+i-1} + a_2x_{n+i-2} + ... + a_nx_i = \sum_{k=1}^{n} a_kx_{n-k+i} \end{array}$$

We will consider in our study, the simplest form corresponding to the single equation:

$$x_{n+1} = f(x_1, x_2, ..., x_n) = \sum_{k=1}^{n} a_kx_{n-k+1} \tag{14}$$

3.1 Maximal Period

In the case whether the polynomial generator is primitive, for all initial state of n non-zero bits, T is the maximal period: $T = 2^n - 1$.

3.2 Identification equation

With our primitive polynomial and the linear recurrence, the identification equation is:

$$\begin{aligned} f(x) &= 1 + \sum_{k=1}^{n} a_k x^k = 1 + a_1 x + a_2 x^2 + ... + a_n x^n \\ x_{n+1} &= \sum_{k=1}^{n} a_k x_{n-k+1} \end{aligned} \tag{15}$$

The identification equation help us to determine the coefficients a_k and therefore the recurrence equation from the polynomial(and vice versa).

$$x_{n+1} \Leftrightarrow f(x) \tag{16}$$

1. Application to 4-bits LFSR:

   Let $f(x) = x^4 + x + 1$

   The maximal period: $n = 4$ and $T = 2^n - 1 = 2^4 - 1 = 15$.

   The identification equation:

$$\begin{aligned} f(x) &= x^4 + x + 1 \\ &= 1 + \sum_{k=1}^{4} a_k x^k \\ &= 1 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 \end{aligned} \tag{17}$$

   We can identify then: $a_3 = a_2 = 0$ and $a_4 = a_1 = 1$.
   But $x_{n+1} = \sum_{k=1}^{4} a_k x_{n-k+1} = a_1 x_n + a_2 x_{n-1} + a_3 x_{n-2} + a_4 x_{n-3}$, then recurrence equation is:

$$\begin{aligned} x_{n+1} &= a_1 x_n + a_4 x_{n-3} \pmod 2 \\ x_{n+1} &= x_n + x_{n-3} \pmod 2 \end{aligned} \tag{18}$$

   we obtain the following maximal-period 4 bits LFSR:

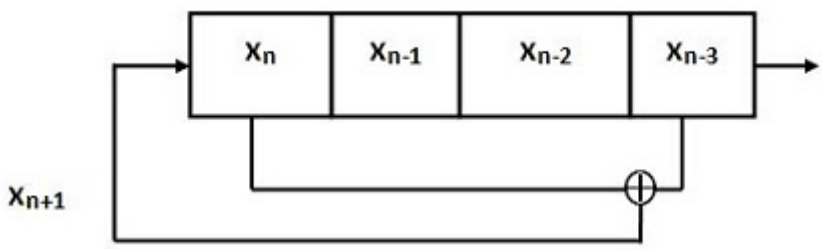


Figure 15: 4 bits LFSR

Remark: If we make this change of variable $x = \dfrac{1}{x}$, we have another possible register Indeed if $f(x)$ primitive, $f'(x) = x^n f(\frac{1}{x})$ is also primitive. Assuming $f(x) = x^n + x^s + 1 \Rightarrow$f$'(x) = x^n(\frac{1}{x}) = x^n(\dfrac{1}{x^n} + \dfrac{1}{x^s} + 1) = x^n + x^{n-s} + 1$ which is also primitive (reciprocal polynomial).

The two polynomials can be used to build register for applications. (More generally $f(x) = 1 + \sum_{k=1}^{n} a_k x^k$ and $f'(x) = x^n + \sum_{k=1}^{n} a_k x^{n-k}$)

With the polynomial$f(x) = x^4 + x + 1$, we have$f'(x) = x^4 + x^3 + 1 = 1 + \sum_{k=1}^{4} a_k x^k = 1 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$.

By using the identification equation, we have:

- $a_4 = a_3 = 1$ and $a_2 = a_1 = 0$
- $x_{n+1} = \sum_{k=1}^{4} a_k x_{n-k+1} = a_1 x_n + a_2 x_{n-1} + a_3 x_{n-2} + a_4 x_{n-3}$
- $\Rightarrow x_n = x_{n-2} + x_{n-3}$

We obtain then the following maximal-period 4 bits LFSR:

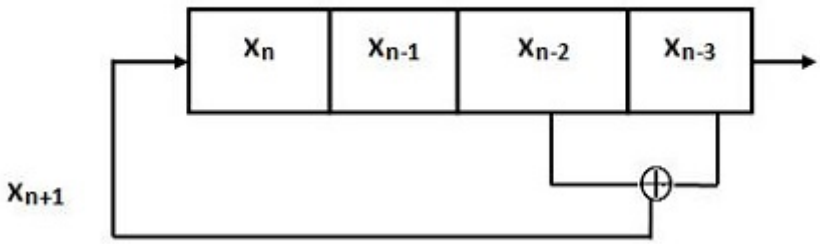


Figure 16: 4-bits LFSR

2. Application to 35-bits LFSR
   Wether: $f(x) = x^{35} + x^2 + 1$

   - maximal period:
     $T = 2^n - 1 = 2^{35} - 1$
   - Identification equation:

$$\begin{aligned} f(x) = x^{35} + x^2 + 1 &= 1 + \sum_{k=1}^{n} a_k x_k \\ &= 1 + a_1 x + a_2 x^2 + ... + a_{34} x^{34} + a_{35} x^{35} \end{aligned} \tag{19}$$

   All $a_k = 0$ except $a_2 = a_{35} = 1$
   - $x_{n+1} = \sum_{k=1}^{35} a_k x_{n-k+1} = a_1 x_n + a_2 x_{n-1} + ... + a_{35} x_{n-34}$
   - By identifying, it comes:

$$x_{n+1} = a_2 x_{n-1} + a_{35} x_{n-34} \tag{20}$$

$$x_{n+1} = x_{n-1} + x_{n-34} \tag{21}$$

We have this maximal-period 35 bits LFSR:

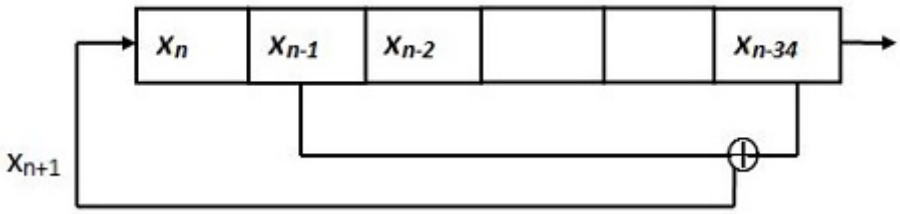


Figure 17: 35 bits LFSR

With the usual change of variable then:
$f(x) = x^{35} + x^2 + 1 \Rightarrow f'(x) = x^{35} + x^{33} + 1$ is also primitive.

- Identification equation:
  $f'(x) = x^{35} + x^{33} + 1 = 1 + \sum_{k=1}^{35} a_k x^k = 1 + a_1 x_+ a_2 x^2 + ... + a_{33} x^{33} + a_{34} x^{34} + a_{35} x^{35}$
  All the $a_k = 0$ except $a_{35} = 1$ and $a_{33} = 1$

$$
\begin{aligned}
x_{n+1} &= \sum_{k=1}^{35} a_k x_{n-k+1} \\
&= a_1 x_n + a_2 x_{n-1} + ... + a_{33} x_{n-32} + a_{34} x_{n-33} + a_{35} x_{n-34} \\
x_{n+1} &= a_{33} x_{n-32} + a_{35} x_{n-34} \\
x_{n+1} &= x_{n-32} + x_{n-34}
\end{aligned}
$$

3. The maximal-period 35 bits LFSR:

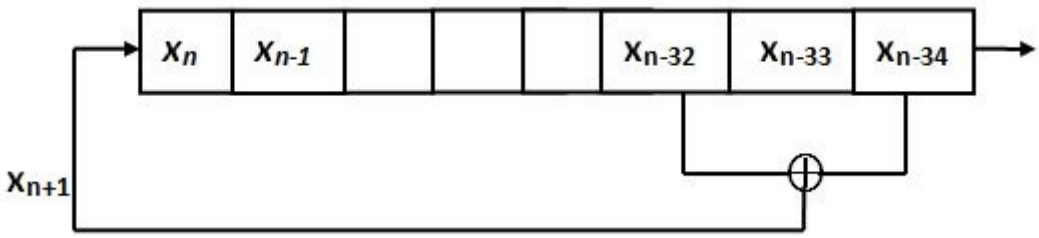


Figure 18: 35 bits LFSR

Finally, it should be noted that[47][65][100][101] recall some works that has been done on binary primitive polynomials and give tables of polynomials available.
However, for the realization of LFSR, it is strongly recommended to use primitive polynomials with non-zero coefficients(dense primitives polynomials) rather than polynomials among which most of the coefficients are zero and which are weak cryptographically.[86]

4 Cryptographic security

4.1 Linear complexity

In terms of cryptographic security, the use of a single LFSR is not sure because this LFSR is predictable:

- And if we know n consecutive bits produced by an n-bits LFSR and the primitive polynomial associated, we can deduce the $(n+1)$-th bit range produced by the register;

- If we also know $(2n)$ consecutive bits produced by an n-bits LFSR without knowing the polynomial primitive associated, we can find this polynomial by the Berlekamp-Massey algorithm[11] [45][64][79][88][90][95].

This algorithm permits us to determine the linear complexity of a random sequence i.e. the length of the smallest LFSR that can generate it(also called linear span)[27][79]. In 1969, James L. Massey [64] proved, in fact, the algorithm proposed in 1967 by Ralph Elwyn Berlekamp for decoding BCH codes[11] also allows the possibility to find the smallest LFSR generating a given sequence.[81] gives a range of results on the linear complexity of random sequences.

4.2 Recommendations-Perspectives

Pseudo-random generator based on FSR used to generate keys must have the following characteristics: [82]

- A long period
- A large linear complexity
- Good statistical properties

As we noted above, the major advantage of LFSR is the ease of hardware and software implementation coupled with their good mathematical conception. However, LFSR, used alone, are not safe an account of their linearity which is exploited to build cryptanalytic attacks foremost among them the Berlekamp-Massey algorithm.

To strengthen cryptographic security of the LFSR generators, we use LFSR generators more complex using nonlinear boolean functions(cryptographic boolean function)[16][17][31][29][46][76][79]which must have certain properties(high algebraic degree, high linear complexity, high non-linearity and high correlation immunity) that can destroy the linearity:

- Non linear combination generators[20][40][42][82][97]: A keystream generator on which the output of several LFSR are combined by a linear function[17][46][82]

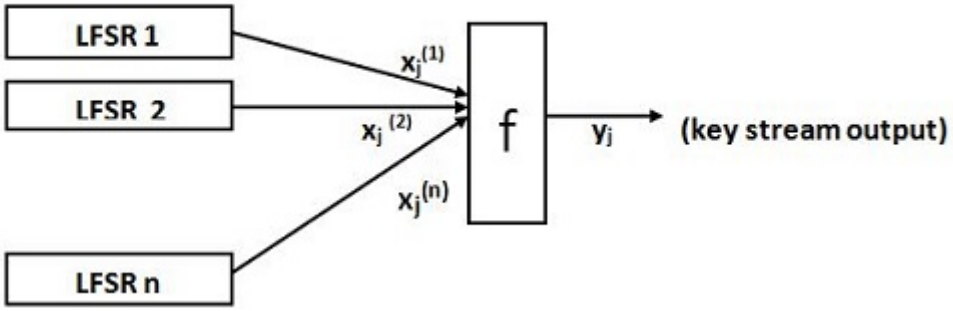


Figure 19: Non linear combined LFSRs

- Non linear filter generator[21][41][82]:A keystream generator consisting of a single LFSR and a nonlinear function (also called Nonlinear filtering function) whose inputs are taken from some shift register stages to produce the output.

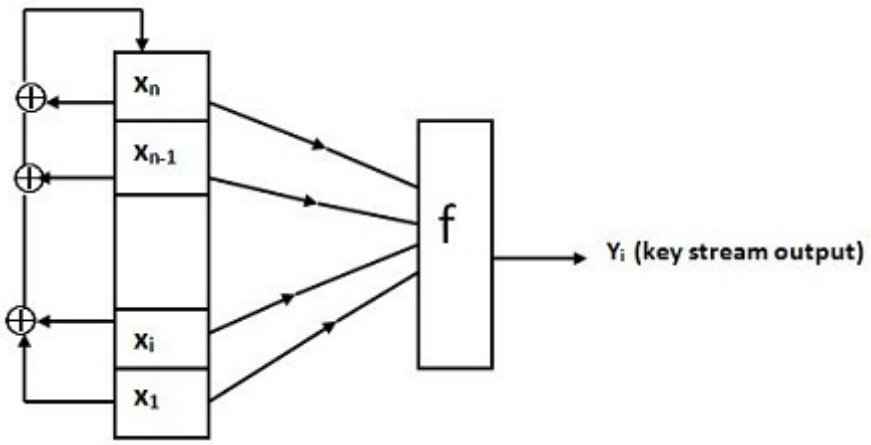


Figure 20: Non linear filter generator

- Clock-controlled generator[12][22][82]: A keystream generator in which an LFSR is used to determine which output symbols of second LFSR are used as the final output.

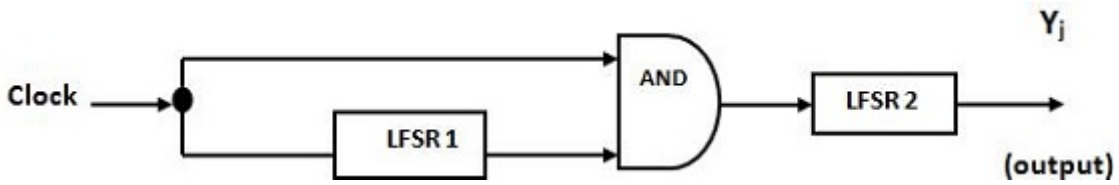


Figure 21: clock control

- It is also more than useful to mention the constructions concerning:
  - The shrinking generated invented in 1993, by D. Coopersmith, H. Krawczys and Y. Mansour[14][24].
  - the self-shrinking generator invented in 1994 by W. Meier and O. Staffelback[69].

But the additional security measures have not so far allowed to shelter attacks:

- Exhaustive attacks;
- Time memory Data Tradeoff attacks[6][13][50];
- Correlation attacks[38][43][50][51][67][68][82][83][91]
- Algebraic attacks[25][26][30][50][51][61][62][67]
- Side channel attacks[52][57][67][92]
- Distinguishing attacks[48][67]

At present, the researches are directed to new classes of feedback Shift registers based notably on the algebraic rings, implying new methods of cryptanalysis and new security measures[27][45][54][55].
In fact, important studies have been conducted in the field of Nonlinear feedback Shift registers(NLFSR), both from the point of view of design as attacks, and which resulted:

- since 1993(by Andrews Klapper and Mark Goresky)to feedback with carry shift registers(FCSR)[45][53][54][55]in Fibonacci mode or Galois mode, with as mathematical basis, the ring of $N-$Adic integers instead of the ring of formal series used for the LFSR.

However, in spite of their large linear complexity, they are susceptible to attacks by the rational approximation algorithm[55]which is similar to that of Berlekamp-Massey.

Therefore they could be coupled with LFSR in the design of pseudo random generators.

- to vectorial feedback with cary shift registers(VFCSR), vector design of the FCSR whose analysis has been extended to finite fields $GF(p^n)$.[2]
- to Filtered feedback with carry shift registers(F-FCSR), design of FCSR to counter the attack by the rational approximation algorithm.[4][5][55]
- to Algebraic feedback shift register(AFSR)when the mathematical basis is $\pi-$Adic ring(not specified as $\pi-$Adic numbers are generalizations of formal series and $N-$Adic integers). LFSR and FCSR are special cases of the AFSR.[45]

It is useful to take a look also on the theory of stability of stream cipher cryptosystem i.e the resistance of such systems to small variations in some of their parameters as regard in particular the linear complexity and nonlinear boolean functions used [27][32][33]:

- For additive synchronous stream cipher, there are already techniques of control of the stability of the linear complexity. But, the problem of the stability of the local linear complexity seems for the moment difficult to solve (this is an open problem).
- For nonlinear combined registers and nonlinear filtered register, partial results were obtained on certain aspects of the theory of stability, but research should be carried further: a promising field of research.

Finally, research is also conducted on the registers with Nonlinear Update(RNLUs) which are generalization of NLFSRs whose study is theoretical and should be further refined.[36][78]

5 Conclusion:

As indicated at the beginning of the article, the proposed method determines mathematically, from the primitive polynomial, linear recurring relation generating the LFSR (and vice versa), and thus facilitate its construction ; it also helps to establish the corresponding reciprocal primitive polynomial which gives the possibility to build another LFSR as good as the first.

We have a design and a careful choice of maximum length LFSR to use, on the basis of the primitive polynomial, the reciprocal polynomial, and associated linear recurring relations, that do not show the methods used so far where the recurrences are established, without further details, from the primitive polynomial to draw the LFSR.

On the other hand, it seemed important to review the LFSR, and to emphasize their cryptographic security with recommendations in the above paragraph and in highlighting research opportunities in this area.

References

[1] Details of the cryptographic principles of the MA4240-Racal Datacom limited, Milford industrial estates Tollgate Road Salisbury,Wiltshire Sp12jg (appendix 1).

[2] Allailou, B. *Conception et Evaluation des générateurs d'Alea.* PhD thesis, Université de Paris 8 (Laboratoire d'Analyse, de Géométrie et Applications-LAGA), 2010.

[3] Anderson, R. On Fibonacci keystream generators. *Lecture Notes in Computer science, Vol.1008, Fast Software Encryption* (December 1994), 346–352.

[4] Arnault, F., and Berger, T., P. F-FCSR,design of new class of stream cipher. *Lecture Notes in Computer science 3557,Fast Software Encryption* (December 2005), 83–97.

[5] Arnault, F., Berger, T., P., and Lauradoux, C. F-FCSR stream cipher. *Lecture Notes in Computer science 4986,New stream cipher designs, the eSTREAM finalists* (2005), 170–178.

[6] Babbage, S. Space Time-off in exhaustive search attacks on stream ciphers. *European Convention on security and detection, IEEE Conference Publication n408* (1995).

[7] Babbage, S., Borghoff, J., and Vosselin, V. The eSTREAM portfolio.

[8] Babbage, S., Canniere, C. D., Canteaut, A., Cid, C., Gilbert, H., Johansson, T., Parker, M., Preneel, B., Rijmen, V., and Robshaw, J., B. The eSTREAM portfolio.

[9] Becker, H., and Piper, F. *Cipher systems,the protection of communications.* Northwood Publications, 1982.

[10] Beckett, B. *Cipher systems,The Protection of Communications.* Masson, 1990.

[11] Berlekamp, Elwyn, R. *Algebraic Coding Theory.* Mc Graw-Hill, 1967.

[12] Beth, T., and Piper, F. The stop-and-go generator. *LNCS 209 Advances in Cryptology-Eurocrypt 84,Springer Verlag* (1985), 88–92.

[13] Birynkov, B., and Shamir, A. Cryptanalytic Time memory Data Trade-off for stream cipher. *Lecture Notes in Computer science,vol n976,Asiacrypt 2000,Springer Verlag* (2000), 1–13.

[14] Blöcher, U., and Dichtl, M. Fish: A Fast Software Stream Cipher,in Fast Software Encryption. *Lecture Notes in Computer science,vol n809,Springer* (2000), 56–63.

[15] Blum, M., and Micali, S. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Computer,Vol 13,N4* (November 1984).

[16] Braeken, A. *Cryptographic Properties of Boolean Functions and S-Boxes.* PhD thesis, Department Electrical Engineering ESTA/COSIC,Universiteit Leuven.Belgium, 2006.

[17] Braeken, A., and Semaev, I. The ANF of the composition of addition and multiplication mod2n with a boolean function. *LNCS,vol n3557,Fast software encryption,12th International Conference Workshop FSE* (February 2005), 121–134.

[18] Brent, Richard, P. On the periods of generalized Fibonacci recurrences. *Lecture Notes in Computer science,vol n3557,Fast software encryption,12th International Conference Workshop FSE* (March 1992).

[19] Canteaut, A. Stream ciphers systems. *Encyclopedia of Cryptology and Security 2005,Springer Verlag* (2005).

[20] Canteaut, A. Combination generator. *Encyclopedia of Cryptology and Security 2011,Springer Verlag* (2011), 222–224.

[21] Canteaut, A. Filter generator. *Encyclopedia of Cryptology and Security 2011,Springer Verlag* (2011), 458–460.

[22] Chambers, W., G., and Gollman, D. Clock-controlled shift registers. *IEEE Journal on selected areas in communications,vol 7,n4* (1989), 525–533.

[23] Chasse, G. Cryptographie mathématique. *Techniques de l'Ingénieur*.

[24] Coppersmith, D., Krawczys, H., and Yishay, M. The shrinking generator. *Lecture Notes in Computer Science,vol n773,Advances in Cryptology,CRYPTO 93,Springer Verlag* (1994), 22–39.

[25] Courtois, Nicolas, T. Fast algebraic attacks on stream cipher with linear feedback. *Lecture Notes in Computer Science,vol n2729,CRYPTO 2003,Springer Verlag* (2003), 176–194.

[26] Courtois, Nicolas, T., and Meier, W. Algebraic attacks on stream ciphers with linear feedback. *Lecture Notes in Computer Science,vol n2656,EUROCRYPT 2003,Springer Verlag* (2003), 345–350.

[27] Cusick, T, W., Ding, C., and Renvall, A. *Stream ciphers and Number theory.* North-Hollow Mathematical Library,Elsevier Sciences B.V, 1998.

[28] Dawson, E. Linear feedback shift registers and stream ciphers.

[29] Dawson, E., and Simpson, L. *Analysis and Design Issues for Synchronous Stream cipher.* Queensland University of Technology,Australia.

[30] Debraize, B., and Goubin, L. Guess-and-determine algebraic attacks on the self-shrinking generation. *Lecture Notes in Computer Science,vol n5086,Fast software encryption,Springer Verlag* (2008), 235–252.

[31] Deepak, Kumar, D., Gupta, Kishan, C., and Maitra, S. Cryptographically significant boolean functions:construction and analysis in terms of algebraic immunity. *Lecture Notes in Computer Science,vol n3557,Fast software encryption,12th International conference workshop,Paris* (February 2005), 107–120.

[32] Ding, C. *The stability theory of stream ciphers.* Department of Computer Science and Engineering,The Hong Kong University of Science and Technology, 2011.

[33] Ding, C., Xiao, G., and Shan, W. *The stability theory of stream ciphers.* Department of Applied mathematics and Institute for Information security,Xidian,China,Springer Verlag, 1991.

[34] Downham, D., Y., and Roberts, F, D. K. Multiplicative Congruential Pseudo-random Generators. *Department of Computational and Statistical Science,The University,Liverpool3*.

[35] Dubrovna, E. Generation of cycles by a composition of NLFSRs. *Designs, Codes and Cryptography, Springer Series, Business Media* (March 2014), 469–486.

[36] Dubrovna, E., and Li, N. An algorithm for constructing a smallest register with non linear update generating a given sequence. *Proceeding of IEEE International Symposium on Multiple-valued Logic(ISMVL 2014)* (May 2014), 254–259.

[37] Dumas, Jean, G., Roch, J.-L., and Varrette, S. *Théorie des Codes.* Dunod, 2007.

[38] Ebrahimi, T., Leprévost, F., and Warusfel, B. *Cryptographie et Sécurité des Systèmes et Réseaux.* Lavoisier, 2007.

[39] Foursov, M. Chiffrements de flux/flot(stream ciphers),Université de Rennes.

[40] Geffe, P., R. How to protect data with ciphers are really hard to break. *Electronics* (January 1973), 99–101.

[41] Golic, Jovan, D. On the security of nonlinear filter generator. *Fast software encryption 1996, LNCS vol.1039,Springer Verlag* (1996), 173–188.

[42] Golic, Jovan, D. Cryptanalysis of alledged A5 stream cipher. *Advances in Cryptology , EUROCRYPT 97,LNCS Vol.1233* (2011), 239–255.

[43] Golic, Jovan, D., Salmasizadeh, M., Simpson, L., and Dawson, E. Fast correlation attacks on nonlinear filter generator. *Information Processing Letters,vol 64,issue 1* (1997), 37–42.

[44] Golomb, Solomon, W. Shift register sequence.

[45] Goresky, M., and Klapper, A. *Algebraic Shift Register Sequences.* Cambridge University Press, 2012.

[46] Gupta, Kishan, C. *Cryptographic and Combinatorial Properties of Boolean Functions and S-Boxes.* PhD thesis, Applied Statistics Unit,Indian Statistical Institute, 2004.

[47] Hansen, T., and Mullen, Gary, L. Primitive polynomials over finite fields. *Mathematics of Computation,vol.59,n200* (October 1992), 639–643.

[48] Hell, M., Johansson, T., and Brynielson, L. An overview of distinguishing attacks on stream ciphers. *Cryptography and Communication,Vol 1-Issue 1,Springer Verlag* (2008), 71–94.

[49] Hersehey, John, E. *Cryptography demystified.* Mc Graw-Hill Telecom.

[50] Joux, A. *Cryptography and Network Security,Algorithmic Cryptanalysis.* CRC Press Taylor and Francis Group, 2009.

[51] Joux, A., Berbain, C., and Gilbert, H. Algebraic correlation attacks against linearly filtered nonlinear feedback shift registers. *Lecture Notes in Computer Science,vol 5381,Springer Verlag* (2009), 184–198.

[52] Joux, A., and Delauney, P. Galois LFSR embedded devices and side channel attacks weaknesses. *Progress in Cryptology-INDOCRYPT 2006,LNCS 4329,Springer Verlag* (2006), 436–451.

[53] Klapper, A. A survey of feedback with cary shift register. *SETA,LNCS 3486,Springer Verlag* (2005), 56–71.

[54] Klapper, A., and Goresky, M. 2-adic shift register. *Fast software Encryption,LNCS 809*, 174–178.

[55] Klapper, A., and Goresky, M. Feedback shift registers,2-adic span and combiners with memory. *Journal of Cryptology,Vol 10,issue 2.*

[56] Klove, T. Linear recurring sequences in boolean rings. *Math. Scand 83* (1973), 5–12.

[57] Kocher, Paul, C. Timing attacks on implementation of Diffie Hellman,RSA,DSS and other systems. *Crypto96,LNCS 1109,Springer Verlag* (1996), 104–113.

[58] Konheim, Alan, G. *Computer security and Cryptography.* John Wiley and Sons,Inc, 2007.

[59] Kurita, Y. Une méthode pour choisir les paramètres d'un générateur de nombres aléatoires congrunetiels. *Laboratoire National de Recherches en Météorologie* (1996), 104,113.

[60] L'Ecuyer, P. Uniform random number generator, Annals of operations research.

[61] Lee, Dong, H. Algebraic attacks on stream ciphers (a survey). *Trends in Mathematics,Information Center for Mathematical Science,Vol 8* (2005), 133–143.

[62] Limniotis, K. Algebraic attacks on stream ciphers-recent development and new results. *Journal of Applied Mathematics and BioInformatics,Vol 3,n1,2013,57-5-81,Scienpress Ltd* (2013).

[63] Mao, W. *Modern Cryptography,Theory and Practice.* Hewlett Packard Company-Prentice Hall PTR.

[64] Massey, James, L. Shift register synthesis and BCH decoding. *IEEE Transaction on Information Theory,Vol IT-15,N1* (January 1969).

[65] Maurin, J. *Simulation déterministe du hasard.* Masson et Cie,Editeurs, 1975.

[66] Mehreen, A., Kansar, F., and Massood, A. Comparative analysis of the structures of estream submitted stream cipher. *ICET 06*, 245–250.

[67] Meier, W. Stream ciphers,a perspective. *LNCS 7374,Africacrypt 2012,Springer Verlag* (2012).

[68] Meier, W., and Staffelback, O. Fast correlation attacks on certain stream ciphers. *Journal of Cryptology N3* (1992), 67–86.

[69] Meier, W., and Staffelback, O. The self shrinking generator. *Lecture Notes in Computer Science,Advances in Cryptology,EUROCRYPT 94,Springer Verlag* (1994), 205–214.

[70] Menezes, Alfred, J., Oorschot, Paul, C. V., and Vanstone, A. *Handbook of Applied Cryptography.* CRC Press,Inc, 1997.

[71] Meyer, Carl, H., and Matyas, Stephen, M. *Cryptography:A new dimension in Computer data security,A guide for design and implementation.* John Wiley and Sons Inc, 1982.

[72] Mogollon, M. *Cryptography and Security services:Mechanisms and Applications.* University of Dallas,USA, 2008.

[73] Oppliger, R. *Contemporary cryptography.* Artech House, 2005.

[74] Paar, C. *Applied cryptography and Data security.* 2005.

[75] Paar, C., and Pelzl, J. *Understanding Cryptography:A Textbook for Students and Practitioners.* Springer Verlag, 2010.

[76] Preneel, B., and Logachev, Oleg, A. Boolean functions in cryptology and information security. *Proceedings of the NATO Advanced Study Institute on Boolean Functions vol 18,IOS Press* (September 2007).

[77] Qu, H. *Stream cipher and Linear complexity.* University of Maryland, 2007.

[78] Rijmen, V. Stream cipher and the estream project. *The ISC International journal of Information security* (January 2010), 3–11.

[79] Robshaw, J., B. Stream ciphers. *RSA Laboratories Technical Report TR-701* (July 1995).

[80] Rothe, J. *Complexity Theory and Cryptography.* Springer Verlag, 2005.

[81] Rueppel, Rainer, A. *Analysis and Design of stream ciphers.* Springer Verlag, 1986.

[82] Rueppel, Rainer, A. Stream ciphers. *Contemporary Cryptology,The Science of Information Integrity,Sindia National Laboratories,IEEE Press* (1992), 65–134.

[83] Salmasizadeh, M., Simpson, L., Golic, Jovan, D., and Dawson, E. Fast correlation attacks and multiple linear approximation. *Information Security and Privacy,LNCS Vol 1270,Springer Verlag* (1997), 228–239.

[84] Saluja, Kewal, K. Linear feedback shift register:theory and applications. *University of Winconsin* (1987,1988,1991).

[85] Schmeh, K. *Cryptography and Public Key Infrastructure on Internet.* Wiley, 2001.

[86] Schneier, B. *Cryptographie appliquée,Protocoles et Codes sources en C.* Thompson Publishing , Vuibert, 1997.

[87] Smart, N. *Cryptography:An Introduction.* McGraw Hill.

[88] Song, H.-Y. Feedback shift register sequences. *Yonsei University,Seoul.*

[89] Stallings, W. *Cryptography and Network security:Principles and Practices.* Prentice Hall, 2011.

[90] Stamp, M., and Low, Richard, M. *Applied cryptanalysis,breaking cipher in the real world.* John Wiley and Sons, 2007.

[91] Stengenthaler, T. Decrypting a class of stream cipher using ciphertext only. *IEEE Transactions on Computers C-34* (1985), 81–85.

[92] Strobel, D. *Side Channel Analysis attacks on Stream ciphers.* PhD thesis, Universitat Bochum, 2009.

[93] Talbot, J., and Welsh, D. *Complexity and Cryptography:An introduction.* Cambridge University Press, 2006.

[94] Trappe, W. *Introduction to Cryptography with Coding theory.* Prentice Hall, 2001.

[95] Van Tilborg, Henk, C. A. *Fundamentals of Cryptology,A professional reference and Interactive Tutorial.* Kluwee Academic Publishers, 2000.

[96] Vergnaux, D. *Exercices et Problèmes de cryptographie.* Dunod, 2012.

[97] Wei, S. On generalization of Geffe's generator. *International Journal of Computer Science and Network Security,Vol 6 n8A* (August 2006).

[98] Yuen, P., K. *Practical cryptology and Web security.* Pearson Education, 2006.

[99] Zemor, G. *Cours de cryptographie.* Université de Bordeaux, 2000.

[100] Zierler, N. Primitive trinomials whose degree is a mersenne exponent. *Information and Control 15* (1969), 67–69.

[101] Zivkovic, M. A table of primitive binary polynomials. *Mathematics of Computation,Vol 69* (1994).