A Proposed Keystream Generator Based on LFSRs.

Adel M. Salman

Baghdad College for Economics Sciences

Abstract

A stream cipher is a system in which we fed a finite key in order to produce infinite key stream to encrypting texts. A stream cipher is used widely because of its security, speed, accuracy, and high flexibility in use. Most of these systems are based on Linear Feedback Shift Registers in order to produce what is called a key stream which must be a pseudo random numbers because of its boolean functions which is used as shift registers. In this paper we proposed a key stream generator algorithm based on feedback shift registers.

المستخلص

التشفير الانسيابي هو نظام يتم تغذيته بمفتاح محدود الطول من اجل انتاج مفتاح انسيابي غير محدود الطول لغرض تشفير النصوص. التشفير الانسيابي يستخدم بشكل واسع بسبب سريته، سرعته، دقته والمرونة في استخدامه. اغلب هذه الانظمة تستند الى المسجل الزاحف الخطي لغرض انتاج المفتاح الانسيابي والذي يجب ان يكون على شكل ارقام شبه عشوائية بسبب الدوال البوليانية المستخدمة على شكل مسجلات زاحفة. في هذا البحث نقدم مقترح لخوارزمية مولد مفتاح عشوائي يعتمد على المسجل الزاحف ذو التغذية المرتدة.

Key words: Linear feed back shift registers, Boolean function, Complexity for solving linear equations.

1- Introduction:

Stream cipher is an important method for information encryption. “A stream cipher is a symmetric cipher which operates with a time-varying transformation on individual plaintext digits”. Stream ciphers typically encrypt data efficiently and have very low memory requirements and therefore cheaper to implement in limited scenarios. Stream cipher techniques are usually best for the cases where the amount of data is either unknown, or continuous such as network streams. A stream cipher generates what is called *keystream* (a sequence of bits used as a key). Encryption is accomplished by combining the *keystream* with the plaintext, usually with the bitwise XOR operation. The generation of the *keystream* can be independent of the plaintext and ciphertext.[1]

Stream ciphers have several advantages which make them suitable for some applications. Most notably, they are usually faster and have a lower hardware complexity than block ciphers. They are also appropriate when buffering is limited, since the digits are individually encrypted and decrypted.[2]

A synchronous stream cipher is one in which the keystream is generated independently of the plaintext message and of the ciphertext. The encryption process of a synchronous stream cipher can be described by the equations

$$\sigma_i + 1 = f(\sigma_i, k),$$
$$z_i = g(\sigma_i, k),$$
$$c_i = h(z_i, m_i)$$

where σ_0 is the initial state and may be determined from the key *k*, *f* is the next-state function, *g* is the function which produces the keystream z_i, and *h* is the output function which combines the keystream and plaintext m_i to produce ciphertext c_i. The encryption and decryption processes are depicted in Figure 1. [3]

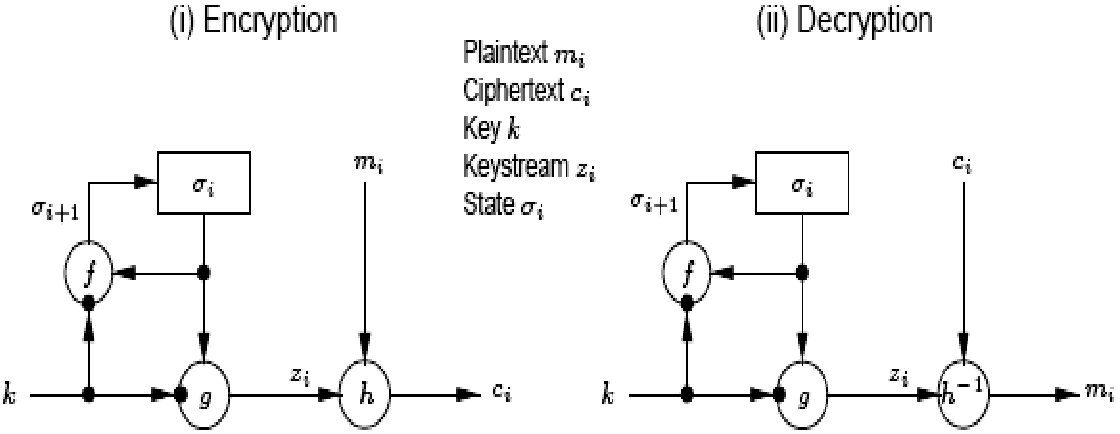


Figure 1: General model of a synchronous stream cipher.

2- Fundamental Concepts :

2-1- Linear Feedback Shift Registers (LFSRs):

Are mostly used in many keystream generators due to their simplicity but inherent linearity of LFSRs not sufficient to provide security to stream ciphers. [2]

Linear feedback shift registers (LFSRs) are used in many of the keystream generators that have been proposed in the literature. There are several reasons for this: [3]

1. LFSRs are well-suited to hardware implementation.
2. They can produce sequences of large period.
3. They can produce sequences with good statistical properties.
4. Because of their structure, they can be readily analyzed using algebraic techniques.

A good start is to use a Linear Feedback Shift Register (LFSR) for achieving a good distribution. The direct output of an LFSR is not a good keystream generator since each symbol produced is simply a linear combination of the previous symbols, and thus very easy to predict. Nevertheless, LFSRs are widely used components inside stream ciphers.[5]

An LFSR is a device made up by registers, able to hold one symbol at a time. The symbols are elements from a field Fq, over which we have chosen to define the LFSR. In stream cipher applications we often have q=2 (binary field) or some extension field of the binary field $q=2^W$, where W is the symbol size of the stream cipher. Initially we can think of an LFSR as a hardware construction, though it is very easy to implement in software as well. Thus we assume a system clock which is responsible for the timing of all events. Figure 2 shows a general LFSR, where the circles denote multiplication with the constant c_i and $\oplus$ is the field addition operation. At each clocking of the LFSR, the registers read a new symbol from their respective input, and as the registers are coupled in series, the symbols move forward at each clocking. The first register receives a new symbol which is a linear combination of the symbols found in the registers after the previous clocking. The exact linear combination used for producing the feedback symbol is determined by the feedback coefficients $c_0, c_1, \ldots, c_l$ shown in Figure 2. Since we need the actual feedback connection c_0 to get any symbols into the register, one normally assumes $c_0 = 1$. As we do not need more registers than necessary to make the feedback connection work, we also assume $c_l \neq 0$ and define the *length* of the LFSR to be *l*. At each time $t \geq 0$ the device is clocked,

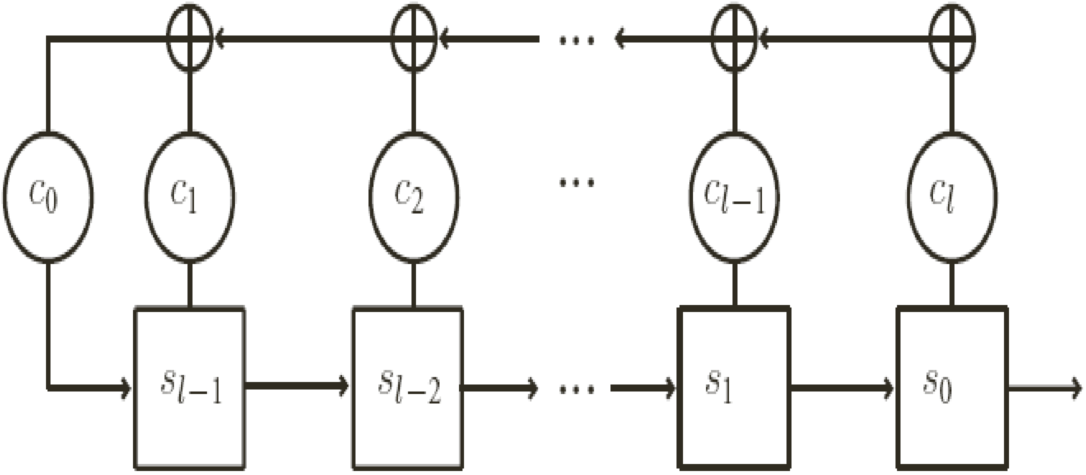


Figure 2: General form of a Linear Feedback Shift Register (LFSR) of length L

and we obtain a new symbol $s_t \in$ Fq at the output of the device. Due to the linear feedback, the symbols st will always fulfill the linear recurrence equation[5]

$$c_0^{-1} s_{t+l} - c_1 s_{t+l-1} - c_2 s_{t+l-2} - \ldots - c_l s_t = 0, \quad t \geq 0.$$

In figure 3 below it shown how the registers could be shifted

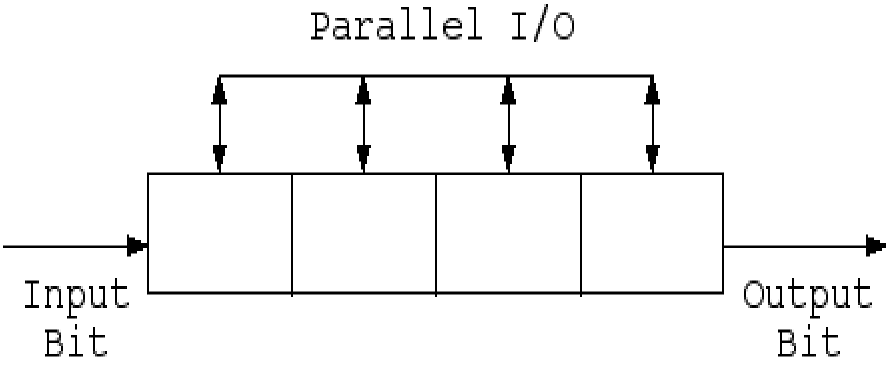


Figure 3: Shift Register

<u>2-1-1- Polynomials:</u> [6]

A polynomial *f(x)* is called ***irreducible*** if *f(x)* cannot be factored as a product of polynomials of smaller degree. Otherwise it is called ***reducible***. (note that the definition of irreducible is closely related to the definition of prime numbers).

Whether a polynomial is irreducible or not, strongly depends on the ground field. For example the polynomial x^2+1 is irreducible over the rationales but is reducible over the complex numbers ($x^2+1=(x+i)(x-i)$).

Let *f(x)* and *g(x)* be two polynomials, *f(x)*=0. Then there exists a unique representation of the form *g(x) = q(x) f(x) + r(x)* with degree (*r(x)*) degree (*f(x)*).

<u>Theorem</u>: Let *f(x)* be an irreducible polynomial over GF(2) of degree L. Then there exists a smallest positive integer P such that the residue of x^P modulo *f(x)* is 1 (i.e. that *f(x)* divides x^P-1), moreover P divides 2^L-1. P is called the period of *f(x)*. An irreducible polynomial with maximal period $P=2^L-1$ is called ***Primitive***.

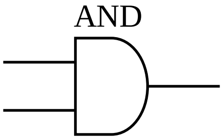
A LFSR sequence with primitive feedback polynomial is called ***maximal length shift register sequence*** (in short m-Sequence). (Note that this definition is justified by the fact that a LFSR of length L cannot produce a sequence of period greater than 2^L-1).

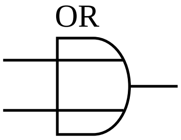
2-2- Boolean Functions: [7]

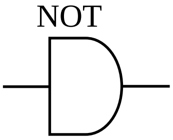
A Boolean variable can only take two values
"False" = 0
"True" = 1

A Boolean function (=function with Boolean arguments and Boolean values) can be described in two ways :

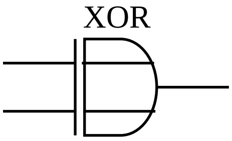
1) The Boolean description uses the operations

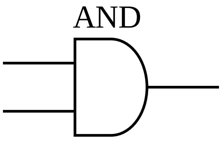






2) The algebraic description uses the operations





"XOR" corresponds to addition modulo 2,
"AND" corresponds to multiplication modulo 2.

Theorem: Every linear autonomous finite state machine A=(S, GF(2), f, g) is equivalent to a linear feedback shift register. Moreover the length of the LFSR cannot be greater than dimension of S.

This description as a linear feedback shift register can be considered as the ***normal form*** of a linear autonomous finite state machine.

2-3- Complexity for solving linear equations: [7]

The number of operation for testing a solution is O(L^2). Thus the number of operations for solving a system of linear equations cannot be smaller than O(L^2) in general.

The system of linear equations for doing the recursion analysis is of a special type. In fact, the matrix of the system is given by

$$\begin{bmatrix} a_{L-1} & a_{L-2} & . & . & . & a_0 \\ a_L & a_{L-1} & . & . & . & a_1 \\ a_{L+1} & a_L & . & . & . & a_2 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ a_{2L-2} & a_{2L-3} & . & . & . & a_{L-1} \end{bmatrix}.$$

Thus the complexity for doing the recursion analysis could be smaller than the complexity for solving a general type system of linear equations.

An efficient algorithm for doing the recursion analysis is the Berlekamp-Massey-Algorithm.

If requires O(L^2) operations for doing the recursion analysis of a sequence with complexity L.

3- The proposed key stream generator:

The proposed generator contains two parts as shown in figure 4 below:

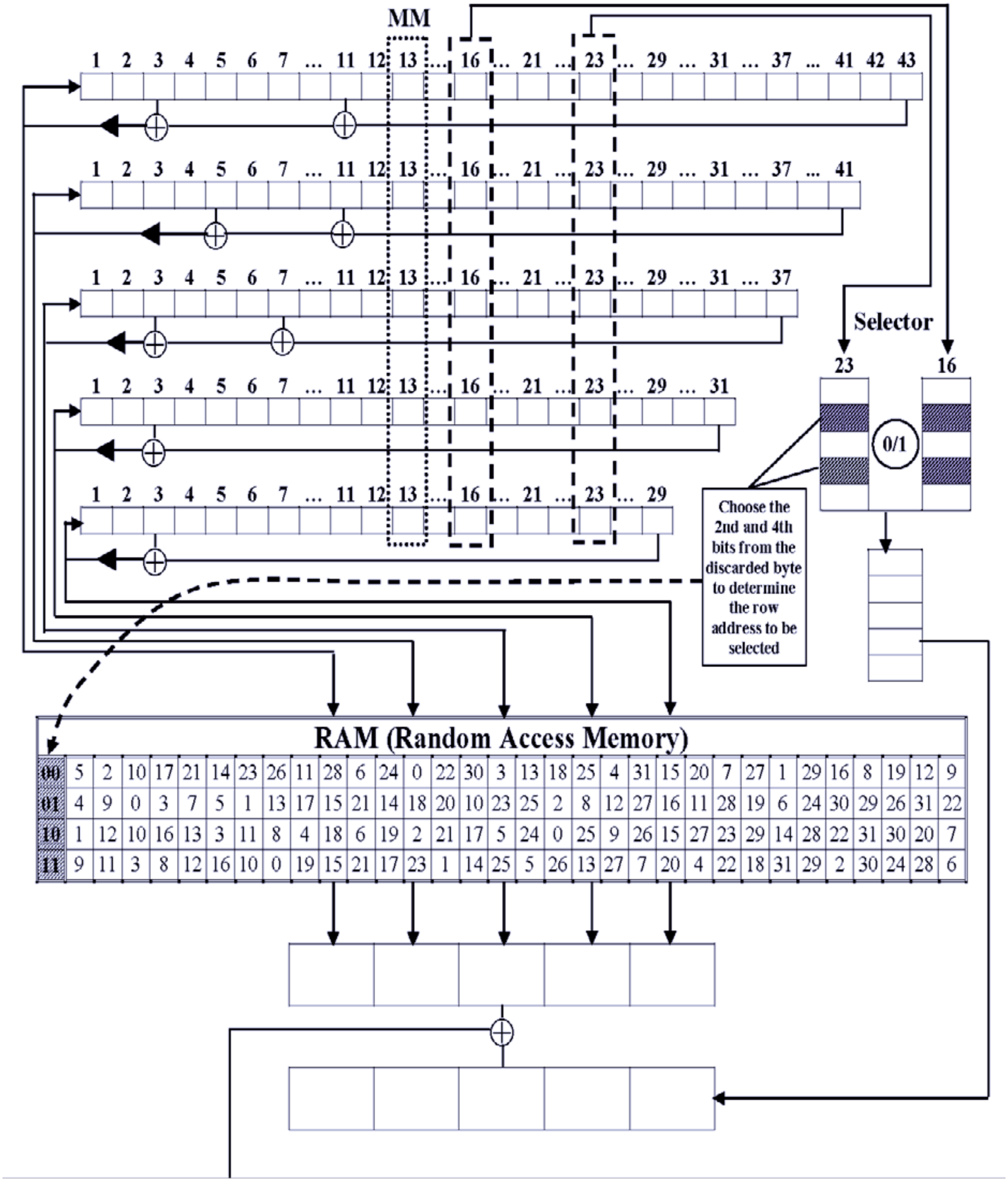
1- Five LFSRs with length (29,31,37,41,43) and the taps (29,3), (31,3), (37,7,3), (41,11,5) and (43,11,3) these polynomials are irreducible and primitive which generates maximal period. The initialization of this part is the secret key (25 characters, 5bit per character) from position 1 to 25 for each register, then by repeating the secret key i.e. the first character in position 26 and the second one in position 27 and so on. The final position in each register contains 1.

Choosing the content of position 13 for movement if it is (0) then the movement of the register will be two clocks and if it is (1) then the movement will be one clock.

Choose two bytes from position 16 and 23 to select one of them depends on the summation of the first register (with length 29 module 2) if (0) then choose position 16 otherwise select position 23.

2- Random Access Memory: random numbers from 0 to 31 with 32 columns and 4 rows and get 5 bits as an address to determine the column from the results of the feedback for the LFSRs of part one.

And two bits (position 2 and 4) from the discarded byte to determine the row.



Choose the 2nd and 4th bits from the discarded byte to determine the row address to be selected

RAM (Random Access Memory)

00	5	2	10	17	21	14	23	26	11	28	6	24	0	22	30	3	13	18	25	4	31	15	20	7	27	1	29	16	8	19	12	9
01	4	9	0	3	7	5	1	13	17	15	21	14	18	20	10	23	25	2	8	12	27	16	11	28	19	6	24	30	29	26	31	22
10	1	12	10	16	13	3	11	8	4	18	6	19	2	21	17	5	24	0	25	9	26	15	27	23	29	14	28	22	31	30	20	7
11	9	11	3	8	12	16	10	0	19	15	21	17	23	1	14	25	5	26	13	27	7	20	4	22	18	31	29	2	30	24	28	6

Figure 4: The Proposed Key Stream Generator

Finally get the two bytes each with 5 bits the first one from the selector of part 1, and the second one from the RAM and by xoring the two bytes we get 5 bits as a key stream.

4- System Complexity :

The complexity of this generator algorithm is 2^{25} for each register so the complexity of part 1 is $2^5 \times 2^5 \times 2^5 \times 2^5 \times 2^5 = 2^{125}$ and in part 2 each location 2^5 and each row contain 32 positions so $(2^5)^{32}$ and 4 rows i.e. 2^2 then the complexity is $2^{160} \times 2^2 = 2^{162}$.

So the whole complexity of this algorithm is $2^{125} \times 2^{162} = 2^{287}$.

5- Statistical Test :

The output is pseudo random sequence because we applied the five basic standard statistical tests: Frequency test, serial test, poker test, run test and autocorrelation test to the samples of the generated sequences. 100 sample sequences were used, each sample has 50 Kbits for level of significance α=0.01the tested samples passed as follows:

100 samples passed the frequency test, and 0 samples failed.
100 samples passed the serial test, and 0 samples failed.
100 samples passed the poker test, and 0 samples failed.
100 samples passed the runs test, and 0 samples failed.
100 samples passed the autocorrelation test, and 0 samples failed.

6- Conclusions:

In this paper a proposed stream cipher algorithm based on the LFSRs architecture has been proposed. It provided the detailed description of the model design with the necessary considerations for the model components. The proposed stream cipher model consists of LFSRs with different lengths as well as different initial states and Random Access Memory (Random numbers from 0 to 31 with 32 columns and 4 rows).

References

[1] R. A. Rueppel. "*Analysis and design of stream ciphers*". Springer-Verlag, 1986.

[2] Arnault, F., Berger, T.” F-FCSR: design of a new class of stream ciphers,” Lecture notes in computer sciences, vol. 3557, pp. 83–97. Springer, Heidelberg 2005.

[3] A. Menezes, P. van Oorschot, and S. Vanstone, "*Handbook of Applied Cryptography*", CRC Press, 1996.

[4] A.Kenso, “Modified self-shrinking generator,” Journal of Computers and Electrical Engineering vol 36, pp. 993–1001, 2010.

[5] P. Ekdahl, "On LFSR Based Stream Ciphers Analysis and Design", LUND Univercity, Ph. D. Thesis, 2003.

[6] E.J. Barbeau, "Polynomials", 1st ed. 1989, 3rd printing, 1989.

[7] Y. Crama and P. L. Hammer, "*Boolean Functions Theory, Algorithms, and Applications*", 2008.