

# A New Approach to Feedback Shift Registers

Myat Su Mon Win<sup>1</sup>

**Abstract**—The pseudorandom number generators based on linear feedback shift registers (LFSRs), are very quick, easy and secure in the implementation of hardware and software. Thus they are very popular and widely used. But LFSRs lead to fairly easy cryptanalysis due to their completely linearity properties. In this paper, we propose a stochastic generator, which is called Random Feedback Shift Register (RFSR), using stochastic transformation (Random block) with one-way and non-linearity properties.

**Keywords**—Linear Feedback Shift Register, Non Linearity, R\_Block, Random Feedback Shift Register

## I. INTRODUCTION

We can control the access of users to the resources by means of smart cards, passwords, fingerprints, etc..

Mutual authentication is also required for communications over the network, between clients and servers. We have to follow an authentication protocol that uses some encryption mechanisms in order to achieve strong authentication. Some applications can also negotiate session keys during or after the authentication that can later be used to cipher the communication, providing integrity and/or confidentiality.

The services of integrity and confidentiality must be applied for the bulk of the information. So, we need fast ciphering mechanisms for high-speed networks. With today's technology, hardware implementations of stream ciphers seem to be the best choice to encrypt at a rate of hundreds of M bits/sec, and thus be compatible with high-speed networks.

Stream ciphering devices seem to be one of the best alternatives in order to provide confidentiality to high-speed transmissions [5]. Feedback Shift Registers (FSRs) are widely used in keystream generators of stream ciphers because they are well suited for hardware implementation, produce sequences having large periods and good statistical properties and readily analyzed using algebraic techniques. Unfortunately, the output sequences of LFSRs are also easily predictable because of their linearity properties. Thus LFSR should never be used by itself as a keystream generator [1]. It should be used by destroying the linearity properties. One approach to destroy the linearity property is to use stochastic transformation based on Random block (R-block) as a feedback function.

The organization of this paper is as follows. Section 2 is devoted to the general structure of LFSR and the properties of maximal length sequence (m-sequence). In section 3, the properties of NLFSRs and the construction of non linear keystream generators are described. A new kind of FSR based on R-block (RFSR) is proposed in section 4. Finally, we conclude our discussion in section 5.

1) Author is the Ph.D candidate of Department of Engineering Physics, Mandalay Technical University, Myanmar, Email: [myatsumon9@gmail.com](mailto:myatsumon9@gmail.com)

## II. LINEAR FEEDBACK SHIFT REGISTERS

Linear Feedback Shift Register (LFSR) is a feedback shift registers which consists of  $n$  input bit shift registers, where the input bit is driven by a linear feedback function of the overall shift register value. The initial value of the LFSR is called the seed, and the sequence of values is completely determined by it [8]. An  $n$  stage LFSR consists of a shift register  $R = (r_n, r_{n-1}, \dots, r_1)$  and a "tap" sequence  $T = (t_n, t_{n-1}, \dots, t_1)$ , where each  $r_i$  and  $t_i$  is one binary digit. At each step, bit  $r_1$  is appended to the key stream, bits  $r_n, \dots, r_2$  are shifted right, and a new bit derived from  $T$  and  $R$  is inserted into the left end of the register (see figure 1).

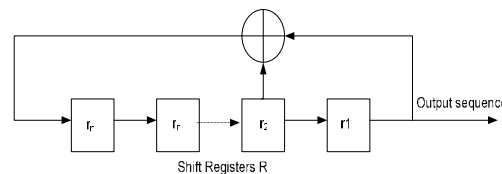


Fig 1 Linear feedback shift register

Letting  $R' = (r'_n, r'_{n-1}, \dots, r'_1)$  denotes the next state of  $R$ .

Thus  $r'_i = r_{i+1} \quad (i = 1, \dots, n-1)$

$$r'_n = TR = \sum_{i=1}^n t_i r_i \bmod 2 = t_1 r_1 \oplus t_2 r_2 \oplus \dots \oplus t_n r_n$$

An  $n$  bit LFSR can be in one of  $2^n - 1$  internal states. This mean that it can generate a  $2^n - 1$  bit long pseudorandom sequence before repeating. It is  $2^n - 1$  not  $2^n$  because a shift register filled with zeros will cause the LFSR to output a never ending stream of zeros whereas this is not particularly useful. Only LFSRs with certain tap sequences will cycle through all  $2^n - 1$  internal states; these are the maximal period LFSRs. The resulting output sequence is called an m-sequence.

In order for a particular LFSR to be a maximal-period LFSR, the polynomial formed from a tap sequence plus the constant 1 must be a primitive polynomial mod 2. The degree of the polynomial is the length of the shift register. A primitive polynomial  $p(x)$  of degree  $n$  over GF (2) is an irreducible polynomial that divides  $x^{2^n-1} + 1$ , but not  $x^d + 1$  for any  $d$  that divides  $2^n - 1$ ,  $d = 2^{n-1}$  [2], so it can not be factored into non-trivial polynomials. An irreducible polynomial  $p(x)$  of degree  $n$  is not said to be primitive if it does not has order  $2^n - 1$ .

An important aspect of irreducible and primitive polynomial is that all the irreducible polynomials are primitive but the reverse is not true. For example, the

polynomial  $x^4 + x + 1$  is irreducible as well as primitive where as the polynomial  $x^4 + x^3 + x^2 + x + 1$  is irreducible but not primitive [6].

LFSRs can have multiple maximal length tap sequences. A maximal length tap sequence also describes the exponents in what is known as a primitive polynomial mod 2. Let's see the above example, a tap sequence of 4,1 describes the primitive polynomial  $x^4 + x + 1$ . Finding a primitive polynomial mod 2 of degree  $n$  (the largest exponent in the polynomial) will yield a maximal length tap sequence for an LFSR that is  $n$  bits long.

There is no quick way to determine if a tap sequence is maximal length. However, there are some ways to tell if one is not maximal length:

- The polynomial is primitive.
- Maximal length tap sequences always have an even number of taps.
- The tap values in a maximal length tap sequence are all relatively prime. A tap sequence like 12, 9, 6, 3 will not be maximal length because the tap values are all divisible by 3.

Discovering one maximal length tap sequence leads automatically to another. If a maximal length tap sequence is described by  $[n, A, B, C]$ , another maximal length tap sequence will be described by  $[n, n - C, n - B, n - A]$ . Thus, if  $[32, 31, 30, 29]$  will also be a maximal length tap sequence. An interesting behavior of two such tap sequences is that the output bit streams are mirror images in time [7].

### III. NON-LINEAR FEEDBACK SHIFT REGISTERS

LFSR should not be used in cryptographic work because the outputs are completely linear, leading to fairly easy cryptanalysis and a better approach is to use a non-linear transformation.

Three general methods are employed to reduce this problem in LFSR-based stream cipher.

- Non-linear combination of several bits from the LFSR state;
- Non-linear combination of the outputs of two or more LFSRs; or
- Irregular clocking of the LFSR

For essentially all possible secret keys, the output sequence of non linear non singular keystream generator should have the following properties [1]:

- Large period;
- Large linear complexity; and
- Good statistical properties.

#### A. Nonlinear combination generators

A nonlinear combination generator uses several maximum-length LFSRs. The key stream is generated as a nonlinear Boolean function  $f$  of the outputs of these LFSRs. The function  $f$  is called the combining function. If  $n$  maximum-length LFSRs with lengths  $l_1, l_2, \dots, l_n$  are used together with the Boolean function  $f$ , the linear complexity of the keystream is

$$f(l_1, l_2, \dots, l_n) = a_0 + a_1 l_1 + \dots + a_n l_n + \dots + a_{l_1 l_2 \dots l_n} l_1 l_2 \dots l_n$$

where  $a_0, a_1, \dots$  are the coefficients in the algebraic normal form of  $f$ , and the expression is evaluated over the ordinary integers instead of the finite field  $F_2$ . Thus, it is desirable to use a combining function with a high nonlinear order. Several feedback shift registers work in parallel and their output is combined using a suitable function  $f$  (see figure 2).

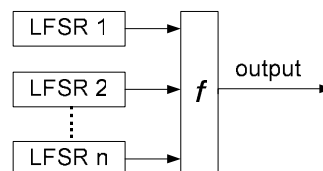


Fig. 2 Structure of combining generator

#### B. Nonlinear filter generators

Another way to destroy the linearity is to use a non-linear filtering function  $f$  with one output and  $n$  inputs, which uses a single maximum-length LFSR (see figure 3.) where  $n$  is the number of stages in the LFSR. A nonlinear filter generator uses a single maximum-length LFSR, and the keystream is generated as a nonlinear function  $f$  of the state of the LFSR. The function  $f$  is called the filtering function. Practical implementation of non-linear filters use S-boxes which have essentially the same requirements than those used in block ciphers. If the LFSR has length  $n$  and  $f$  has nonlinear order  $m$ , the linear complexity is at most  $L_m = \sum_{i=1}^m \binom{n}{i}$  [4].

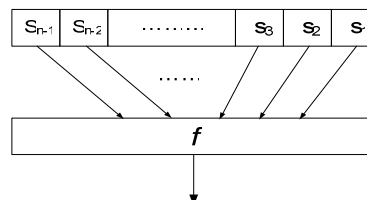


Fig. 3 Non linear filter generator

#### C. Clock controlled generators

There are many variants, but the basic idea is the following: one or more LFSRs control the clock of one or more others LFSRs. A simple variant is the Stop-and-Go generator. It consists of two registers, whereas one register clocks the other if the output is 1, otherwise the previous output is repeated (see figure 4).

Here, the first LFSR R1 decides whether LFSR R2 or LFSR R3 should be clocked. The output is eventually generated via an XOR using R2 and R3 as inputs [1].

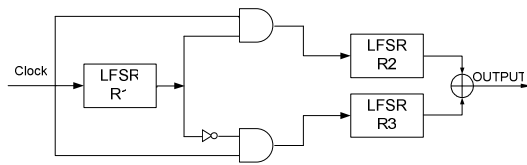


Fig. 4 Clock control generator

#### IV. RANDOM FEEDBACK SHIFT REGISTERS

Another approach to destroy the linearity in LFSR is to use stochastic transformation based on R-block as a feedback function. These registers are called Random Feedback Shift Registers (RFSRs).

##### A. Stochastic transformation

General model of stochastic transformation is as shown in figure 5.

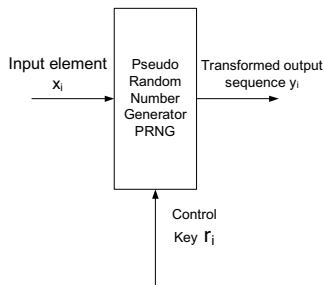


Fig. 5 Stochastic transformation

For each input element  $x_i, (i = \overline{1, m})$  repeat the following steps.

- the first input element of  $x_i$  put into the stochastic generator
- the first control key  $r_i$  is ordered to work these generator
- after working the first control key of  $r_i$ , the generator generates the first resulting sequence of  $y_i$ .

After converting all elements of the original sequence will be obtained resulting sequence of length  $m$ ,

$$y = y_1 y_2 y_3 \dots y_i \dots y_m$$

$$\text{For each element, } y_i = R(x_i, r_i).$$

This transformation can be used effectively to the various tasks related to the protection of information [3].

##### B. Principle of R-block

For the construction of R-block stochastic transformation and its conditional graphic symbol are shown in figure 6 and 7.

H table is filled with key information as the following:

$$H = \{H(m)\}, m = \overline{0, (2^n - 1)}.$$

H table contains elements over  $GF(2^n)$  with 2-dimension  $n \times n$ , i.e. the random variable  $H(m) \in GF(2^n)$ . The resulting equation of stochastic transformation using R-block is:

$$R_H(A, B) = H((m_A + B) \bmod 2^n)$$

where,  $A, B$  are input elements of registers,  $m_A$  is an address of cells containing code  $A$  in  $H$  table, i.e.,  $H(m_A) = A$ . The output of R-block is the essence of reading the contents of cells in  $H$  table. The result  $R_H(A, B)$  of the stochastic transformation depends on the key information in  $H$  table and input parameter  $B$  [3].

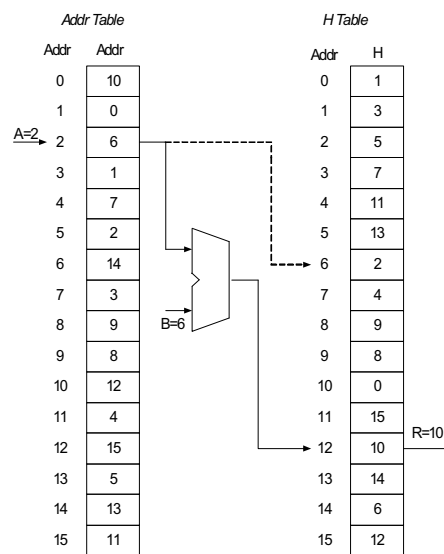


Fig. 6 The logic work of R-block

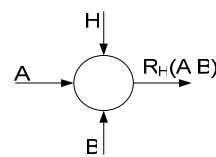


Fig. 7 Conditional graphic symbol for R-block

##### C. Stochastic generators based on R-block

LFSR is constructed by using a linear function as a feedback while NLFSR is built by using a nonlinear feedback function. A stochastic generator RFSR may be constructed by using stochastic transformation R-block as a feedback function (see figure 8) [3].

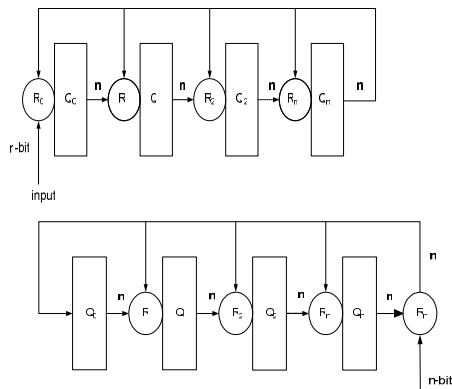


Fig. 8 The two general scheme of n bit RFSRs

We can choose the scheme with one R-block, which can be presented in two identical options (see figure 9-a and figure 9-b).

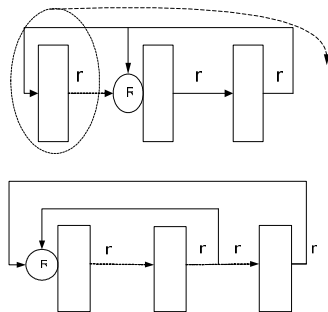


Fig. 9 Stochastic Generator with one R-block (a) RFSR 1 (b) RFSR 2

If we make an excellent choice of stochastic conversion tables, i.e. H tables, non-linear maximum length sequence (m-sequence) can be obtained. The statistical properties of these sequences are superior to the classical m-sequence released from the LFSR with the same period. Figure 10 shows sequences generated by a stochastic generator based on RFSR with reference to various H tables and we found m-sequences among them.

We can try to get the sequence with length  $2^n$ , where n is number of bits, for m-sequences mentioned above by using Boolean functions in given design principle. Boolean function is used to connect the discrete sequences (see figure 11).

Figure 12(a) is an example of the transformation of a stochastic generator, which consists of three cycles in length 22, 25, 16 and one trivial cycle consisting of the state of 000. The transformation may produce a sequence with maximum length 64. In order to get it, the transformation has required the inclusion of the device and the element summator, and element NOR, output of which is connected with the entrance of transfer summator. Conversions of original generator in figure 12 (a) shows a dotted line.

Figure 12(b) is an example of the transformation of a stochastic generator, which consists of three cycles in length 33, 7, 23 and one trivial cycle consisting of the state of 000. The transformation may produce a sequence with maximum length 64. In order to get it, the transformation has required the inclusion of the device and the element XOR, and element NOR, output of which is connected with the LSB bit of register B. Conversions of original generator in figure 12(b) shows a dotted line.

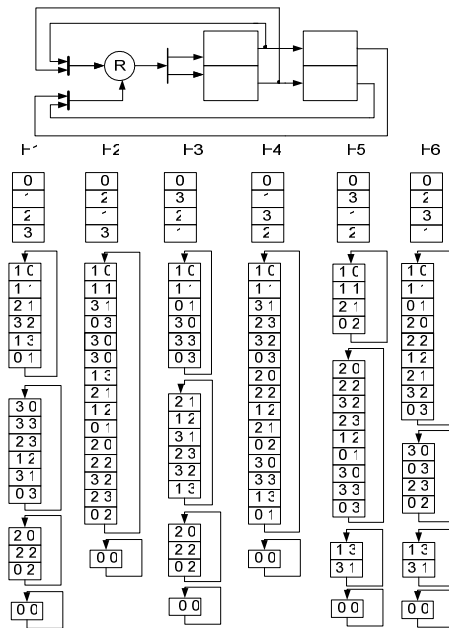
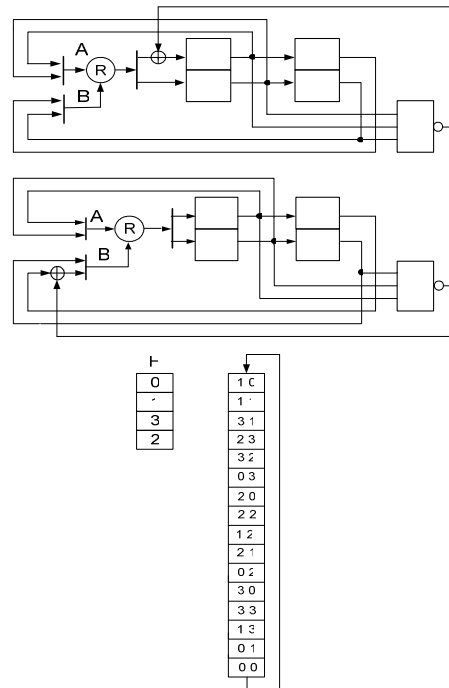


Fig. 10 Sequences generated by RFSR with reference to various H tables

Fig. 11 Designs of RFSR with length  $2^n$

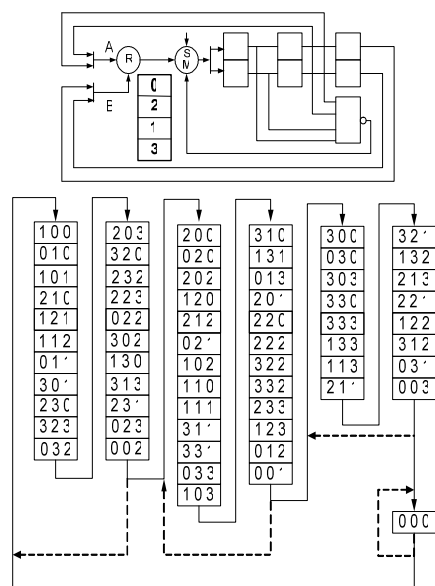


Figure 12. Designs of RFSR with length 64 by using (a) summator

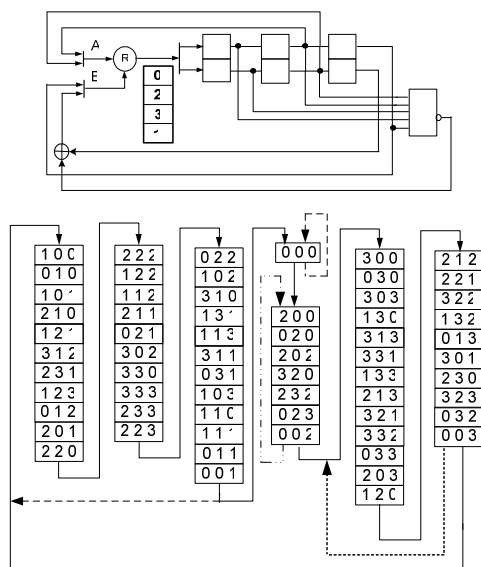


Figure 12. Designs of RFSR with length 64 by using (b) NOR-XOR functions

## REFERENCES

- [1] A.Menezes, P.van Oorschot and S. Vanstone, Handbook of Applied Cryptography, CRC press (1996), pp. 203--209.
- [2] A statistical Test Suit for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST Special publication 800-22, May 15, 2001.
- [3] Bruce Schneier, Applied Cryptography 2<sup>nd</sup> edition: Protocols, Algorithms, and Source Code in C, John Wiley & Sons, Inc., (1996), pp. 374.
- [4] I.V. Chugunkov, M.A. Ivanov, Theory, the use and evaluation of the quality of random sequences generators, Russia, December 27, (2007).
- [5] John Mattson, Stream Cipher Design, Master of Science Thesis, Stockholm, Sweden, (2006), pp. 11-12.
- [6] Rau'l Gonzalo, Daniela Ferrero, Miguel Soriano, Non-Linear Feedback Shift Registers with Maximum Period, May 15, (1997).
- [7] S.Chattopadhyay, S.K.Sanyal, R.Nandi, Development of algorithm for the generation and correlation study of maximal length sequences for applicabilities in CDMA mobile communication systems, India.
- [8] <http://homepage.mac.com/afj/lfsr.html>

## V. CONCLUSION

In this paper, we have implemented RFSR that is based on stochastic transformation which is called R-block. We gave the various designs of stochastic generators based on R-block which release a sequence with full period. It is more efficient and secure than LFSR to generate sequences with large complexity, for use in hardware and software implementation.