Chapter 61
Design and Analysis of LFSR-Based Stream Cipher

Subhrajyoti Deb, Bubu Bhuyan and Navin Ch. Gupta

Abstract Stream ciphers are increasingly used for lightweight applications like RFID, Bluetooth, and Wi-Fi communications where the length of the plaintext is initially unknown. Generally, the stream ciphers are characterized by fast encryption and decryption speed. The LFSR-based stream cipher can generate pseudorandom binary strings with good cryptographic properties. Hardware implementation cost is also minimum for it. In this paper, we have discussed the architecture and properties of the LFSR. We have also discussed the properties of secured Boolean function as one of the important components of stream cipher. Here, we have implemented a generalized nonlinear combination of generator-based model comprising LFSR/NLFSR and Boolean function for designing a pRNG. The bitstream properties of pRNG are tabulated and compared with their best attainable parameters.

Keywords LFSR · Stream · Cryptography · Sagemath

Introduction

Cryptography is the study of mathematical techniques used for achieving secured communication over the unsecured channel. Cryptographic primitives are designed to deal with the basic security issues like confidentiality, integrity, authentication, and non-repudiation. It can be classified into two categories, namely symmetric key and asymmetric key primitives. Symmetric key cryptographic primitives have the advantage of higher throughput over asymmetric ones, and therefore symmetric key cryptographic primitives are widely used for bulk data encryption and decryption.

S. Deb (✉) · B. Bhuyan · N. C. Gupta
Department of Information Technology, North-Eastern Hill University, Shillong, India
e-mail: subhrajyotideb1@gmail.com

B. Bhuyan
e-mail: b.bhuyan@gmail.com

N. C. Gupta
e-mail: rainynavin@gmail.com

J. K. Mandal et al. (eds.), *Proceedings of the International Conference on Computing and Communication Systems*, Lecture Notes in Networks and Systems 24, https://doi.org/10.1007/978-981-10-6890-4_61

Symmetric key ciphers are again subdivided into two categories, namely block cipher and stream cipher. The stream ciphers have the advantages of higher throughput, low latency, and lesser error propagation effect than that of block cipher. The basic working principle of the stream cipher is to generate an arbitrarily long pseudorandom bit stream from a given random string and that pseudorandom bitstream is used to encrypt the message stream. Further, stream cipher based on LFSR (Linear Feedback Shift Register) is characterized by its lightweight property and ease of implementation in hardware. A few examples of popular stream ciphers are A5/1 used in GSM security, E0 used in Bluetooth, RC4 used in SSL, etc. A few prominent lightweight stream ciphers are Grain, WG, Trivium, SNOW, Salsa 20, Sprout, SOBER, etc. [1–3].

The outline of the paper is as follows: section "Literature Survey" provides the literature survey. Section "Background and Preliminaries" provides the background and preliminaries related to stream cipher. Section "Analysis of Feedback Shift Registers for Stream Cipher" provides the FSR analysis part. Section "Proposed Design" describes the architecture of implemented nonlinear generator model and its result analysis. Finally, section "Conclusion and Future Work" concludes the paper.

Literature Survey

Here, we have discussed the different types of LFSR-based stream ciphers. A renewed interest has grown among the research communities for analysis and design of stream ciphers due to the launch of eSTREAM project [1]. This research project was maintained by European Network of Excellence for Cryptology from 2004 to 2008. Only seven candidates are chosen from long-term research project in Europe known as ECRYPT [4]. In Table 61.1, we have listed a few LFSR-based stream ciphers and their building process [2, 3].

Background and Preliminaries

In this section, we have included mathematical preliminaries related to LFSR-based stream cipher design.

Boolean Function

A Boolean function is a mapping from $F_2^n \rightarrow F_2$, over the finite field with two elements$\{0, 1\}$. If the number of combination mapping consists of an equal number of $1's$ and $0's$, then the Boolean function is called as balanced.

Table 61.1 LFSR-based stream cipher list

Cipher name	Platform used	Building process
A5/1	Hardware	Combination of three LFSRs with irregular clocking
E0	Hardware	LFSR of length 4
Sosemanuk	Software	LFSR of length 10
HC-256	Software	Nonlinear Filter 32-bit-to-32-bit mapping Linear masking
Trivium	Hardware	NLFSR (Nonlinear Feedback Shift Register) of lengths 93, 84, and 111
Grain	Hardware	NLFSR of length 128 and LFSR of length 128
Sprout	Hardware	LFSR and NLFSR of the length 40
WG	Hardware and software	LFSR of length 11
Espresso	Hardware	256-bit NLFSR are in Fibonacci configuration
Decim v2	Hardware	LFSR of length 192
Decim-128	Hardware	LFSR of length 288

For example, let us consider $n = 3$ variable Boolean function $f(x_1, x_2, x_3) = x_1x_2 + x_2x_3 + x_3$. The input sequences of (x_1, x_2, x_3) are (000, 100, 010, 110, 001, 101, 011, 111) and the final output of the Boolean function is depicted as (0, 0, 0, 1, 1, 1, 0, 1).

A cryptographically secured Boolean function should satisfy the following properties [5, 6]:

- Boolean function should be balanced.
- The nonlinearity and correlation immunity of the function should be high so that it can resist correlation attack.
- The algebraic degree and algebraic immunity of the function should be high so that it can resist algebraic attack.

Algebraic Normal Form

Usually, every Boolean function has a distinct representation as a multivariate over F_2 which is known as algebraic normal form (ANF). This function can be represented as

$$f(x_1, x_2, \dots, x_n) = c_0 \oplus \sum_{1 \le i \le n} c_i x_i \oplus \sum_{1 \le i \le j \le n} c_{i,j} x_i x_j \oplus \dots \oplus c_{(1,2,\dots,n)} x_1 \dots x_n$$

where the coefficients $c_0, c_i, c_{i,j}, \dots, c_{(1,2,\dots,n)} \in F_2$. In this function, the number of variables in the highest order product term (with coefficient non zero) is known as the algebraic degree. In general, when the degree of the function f is at most one, it can be described as an affine function. The affine functions with ($c_0 = 0$) are known as linear functions [6, 7].

Walsh Transform

This transformation function is an n variable Boolean function. In that case, $c = \{c_1 \dots c_n\} \in F_2^n$ and a n variable linear function can be represented as $l_c(x) = c_1 x_1 \oplus \dots \oplus c_n x_n$. So, this transformation function can be described as

$$W_f(c) = \sum_{x \in F_2^n} (-1)^{f(x) \oplus l_c(x)}$$

From the above definition of $W_f(c)$, it can be observed that, when $f(x) \oplus l_c(x)$ value is 0, then sum is increased by 1, and when this value is 1, sum is decreased by 1 [6, 7].

Nonlinearity

Nonlinearity of a Boolean function f of n variables can be described as the distance between the function and the set of all possible affine functions. Nonlinearity can be defined in terms of Walsh transform as given below [7, 8]:

$$\mathrm{nl(f)} = 2^{n-1} - \frac{1}{2} max\, |W_f(c)|$$

Correlation Immunity

A Boolean function f on F_2^n is said to be correlation of order m, where $1 \le m \le n$, if the output of f and any m input variables are statistically independent. A Boolean function f on F_2^n is correlation immune of order m iff $W_f(c) = 0$ for all vectors $c \in F_2^n$ such that $0 \le |c| \le m$ [6, 8].

Analysis of Feedback Shift Registers for Stream Cipher

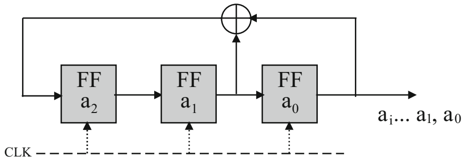
The normal way of designing Feedback Shift Register (FSR) through binary sequences, $\{c_0, c_1, \dots c_n\} \in GF(2)$, fulfills the recurrence relation of order n. FSR is created by *D Flip-flops* which are connected serially and each *D Flip-flop* constructed in such a way that each gate is simulating the Boolean logic for feedback function. Moreover, if FSR runs with linear recurrence, feedback function is known as LFSR and if it runs with nonlinear recurrence, then feedback function is known as NLFSR [9]. eStream selected Grain, Trivium, and MICKEY stream cipher are designed by NLFSR [1].

LFSR

Linear Feedback Shift Register (LFSR) is a shift register with a feedback path. Here, the output sequence of each *D Flip-flop* is joined to the input of the adjacent *D Flip-flops*. Feedback path is defined as the tap position of *D Flip-flop* which takes part in the XOR (modulo 2) operation and provides input to the last *D Flip-flop*. Initial value of LFSR is known as seed value of LFSR. The feedback path is also known as feedback function or connection polynomial [9, 10].

For example, let us consider a LFSR degree is $m = 3$. The LFSR structure and feedback path are shown in Fig. 61.1. Here, this feedback path can be represented in polynomial form as $(x^3 + x^2 + 1)$. The internal state bits are expressed as a_i and it has been shifted by one to the right at each clock. In that case, rightmost state bit is considered as present output bit and the leftmost state bit is calculated by feedback path. Let us consider the output bit is a_i and assuming the initial state bits are (a_0, a_1, a_2).

Now, the output sequence of the LFSR can be calculated as $a_{i+3} = a_{i+1} + a_i$ mod 2, where $i = 0, 1, 2, 3, \dots,$.



$$\begin{aligned} a_3 &= a_1 + a_0 \bmod 2 \\ a_4 &= a_2 + a_1 \bmod 2 \\ a_5 &= a_3 + a_2 \bmod 2 \\ &\vdots \end{aligned}$$

Fig. 61.1 Schematic diagram of LFSR

Properties of LFSR

- In l-stage LFSR, if l number of registers are available in the LFSR, then the number of states is equal to 2^{l-1}.
- However, every feedback path or connection polynomial will not give maximum length. The LFSR will yield maximum length if and only if the corresponding feedback path is primitive polynomial.

Klapper and **Goresky** developed similar type of LFSR design, known as Feedback with Carry Shift Register (FCSR). There are two different types of LFSR, namely Fibonacci and Galois [11]. In FCSR and LFSR, linear sequences are possible to employ through Berlekamp–Massey algorithm [10, 11].

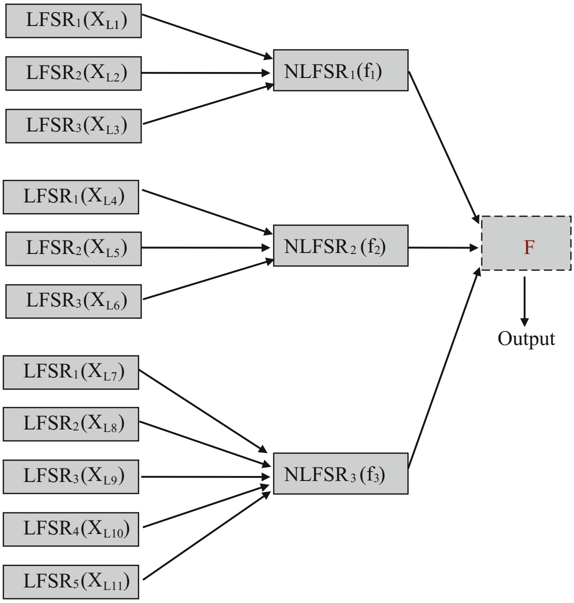
LFSR-Based Stream Ciphers

The main use of LFSR in stream cipher is to produce pseudorandom sequence. We know, LFSR can generate an infinite bitstream. In the most common form, multiple LFSRs are used to build a stream cipher. But LFSR exhibits linear property. Thus the nonlinearity concept has been introduced to overcome the drawback of linear property [12] by irregularly changing the clock of the LFSR. LFSR-based stream cipher uses mainly three classes of pRNGs (pseudorandom number generators), namely *nonlinear combination, nonlinear filter, and clock-controlled generators* [13]. Almost every LFSR-based stream ciphers follow any of these nonlinear techniques or use a combination of these techniques with some extra efforts like adding counter or a combination of different LFSRs with NLFSR [1]. Usually, nonlinear combiner design employs n number of LFSRs of different lengths. In that case, all are initially assigned with nonzero seed values. During each clock pulse, n number of results from the LFSRs are taken and filled as n data inputs to an n variable Boolean function. In case of nonlinear filter generator, n numbers of outputs from different positions of the LFSR are filled as n inputs to an n variable Boolean function. Moreover, the Boolean function and memory collectively construct an FSM [7].

Proposed Design

Our proposed model follows a simple implementation of nonlinear combination generators shown in Fig. 61.2. The model comprises cryptographically secure Boolean function and number of LFSR can be added in a customized fashion. Here, for simplicity purpose, we use less number of the LFSR.

Fig. 61.2 Proposed design structure



Design Specifications

This design consists of three main blocks. In the first block, three LFSRs are initially loaded and passed through $NLFSR_1(f_1)$ function. The second block is also loaded with three LFSR and passed through $NLFSR_2(f_2)$ function and in the third block, five LFSRs are loaded and passed through $NLFSR_3(f_3)$ function. In LFSR, the feedback polynomial values are listed in the next subsection. Finally, three blocks are passed through one nonlinear function (F).

Initialization

Before the output sequence generation, the structure must be initialized with nonzero seed values. Usually, LFSR connection polynomial over *GF(2)* is the primitive polynomial or it can be called as the update function. Now, LFSR filled with a sequence of bits or it can be loaded like a fixed sized bit of hex values, or a string. Table 61.2 shows the LFSR tap polynomials. Specifically, results of the structure, i.e., binary sequences of the functions, are listed in the matrix order.[1]

[1] LF is represented as LFSR.

Table 61.2 Parameters of the proposed model

Tab Polynomials / Feedback Path $LFSR_1 = x_6 + x_5 + 1$ $LFSR_2 = x_5 + x_3 + 1$ $LFSR_3 = x_4 + x_3 + 1$	$NLFSR_1$ Combining function: $f_1(x_1, x_2, x_3) = x_2 + x_1x_3 + x_1x_2$	Output function: $F(x_1, x_2, x_3) = x_3 + x_1x_2 + x_3x_2$
Tab Polynomials/ Feedback Path $LFSR_4 = x_7 + x_6 + 1$ $LFSR_5 = x_5 + x_3 + 1$ $LFSR_6 = x_{11} + x_9 + 1$	$NLFSR_2$ Combining function: $f_2(x_1, x_2, x_3) = x_3x_2 + x_3x_1 + x_1$	
Tab Polynomials/ Feedback Path $LFSR_7 = x_5 + x_3 + 1$ $LFSR_8 = x_7 + x_6 + 1$ $LFSR_9 = x_8 + x_6 + x_5 + x_4 + 1$ $LFSR_{10} = x_9 + x_5 + 1$ $LFSR_{11} = x_{10} + x_7 + 1$	$NLFSR_3$ Combining function: $f_3(x_1, x_2, x_3, x_4, x_5) = x_1x_2 + x_2x_3 + x_3x_4 + x_4x_5 + x_5$	

$$\left(\begin{array}{cccccccccccccccc}
LF_1 & 1&0&0&1&0&0&1&0&1&1&0&1&1&1&0 \\
LF_2 & 1&0&1&0&1&0&0&0&0&1&0&0&1&0&1 \\
LF_3 & 0&0&1&1&0&1&0&1&1&1&1&0&0&0&1 \\
LF_4 & 1&0&0&1&0&0&1&1&0&1&1&0&1&0&1 \\
LF_5 & 0&1&0&1&0&0&0&0&1&0&0&1&0&1&1 \\
LF_6 & 1&0&0&0&1&0&0&1&1&0&1&1&0&1&0 \\
LF_7 & 1&1&0&0&0&1&1&0&1&1&1&0&1&0&1 \\
LF_8 & 0&0&1&1&0&0&0&0&1&0&1&0&0&0&1 \\
LF_9 & 1&1&1&1&0&0&0&0&1&0&1&1&1&1&0 \\
LF_{10} & 1&0&0&0&1&0&0&0&1&0&0&0&0&0&0 \\
LF_{11} & 1&1&1&1&1&0&0&0&0&0&0&0&1&1&1 \\
f_1 & 0&0&1&1&1&0&0&0&1&1&0&0&0&0&1 \\
f_2 & 0&0&0&1&0&0&1&0&1&1&0&1&1&1&1 \\
f_3 & 1&1&0&0&0&0&0&1&0&0&0&1&1&1&0 \\
F & 1&1&0&1&0&0&0&1&1&1&1&0&0&0&1
\end{array}\right)$$

Results and Performance Analysis

We have done all our experiments in SageMath tool. Here, the final bits obtained from the nonlinear filter, i.e., F, will be considered as the final output bit which is used as a nonlinear sequence. We have taken only 15 bits of output; however, the number of bits can be increased or decreased as per requirement and also can be further converted to fixed bit hex values or string. In this paper, we have analyzed the nonlinearity property of the proposed schemes which are numerically depicted. Table 61.3 shows the typical values of parameters such as balancedness, nonlinearity,

Table 61.3 Cryptographic properties obtained after using the Boolean functions

Function	Balancedness	Nonlinearity	Maximum nonlinearity	Algebraic immunity	Correlation immunity	Walsh transform
f_1	YES	2	2.59	2	0	$(0, 4, 4, 0, 0, 4, -4, 0)$
f_2	YES	2	2.59	2	0	$(0, 0, -4, 4, -4, -4, 0, 0)$
f_3	YES	12	13.18	2	0	$(0, 8, 0, -8, 8, 0, 8, 0, 0, -8, 0, 8, -8, 0, -8, 0, 8, 0, 8, 0, 0, 8, 0, -8, 8, 0, 8, 0, 0, 8, 0, -8)$
F	YES	2	2.59	2	0	$(0, 4, 0, -4, 4, 0, 4, 0)$

maximum nonlinearity, algebraic immunity correlation immunity, and Walsh transform. More specifically, proposed model resultant bits are shown in Table 61.3. It shows the maximum possible nonlinearity and proposed design nonlinearity.

Conclusion and Future Work

In this paper, we have surveyed LFSR-/NLFSR-based stream ciphers. We have also implemented one nonlinear-based generator model to generate cryptographically secured bitstream. The various properties of randomness like algebraic immunity, correlation immunity, Walsh transformation, nonlinearity, etc. are listed in the tabulated form. The nonlinearity of the bitstream is compared with maximum nonlinearity achievable for a particular Boolean function. Research problems on NLFSR are still not well understood like patterns and its behaviors. Development of an intelligent algorithm for designing customized LFSR-based stream cipher using the generalized model shall be our future research work.

References

1. eSTREAM: the ECRYPT Stream Cipher Project. http://www.ecrypt.eu.org/stream/
2. Kocheta, M., Sujatha, N., Sivakanya, K., Srikanth, R., Shetty, S., Mohan, P.A.: A review of some recent stream ciphers. In: 2013 International conference on Circuits, Controls and Communications (CCUBE). (2013)
3. Eisenbarth, T., Kumar, S., Paar, C., Poschmann, A., Uhsadel, L.: A survey of lightweight-cryptography implementations. IEEE Design & Test of Computers **24**(6) (2007) 522–533
4. Videau, M. In: eSTREAM. Springer US, Boston, MA (2011) 426–427
5. Shehhi, M.A.A., Baek, J., Yeun, C.Y.: The use of boolean functions in stream ciphers. In: Internet Technology and Secured Transactions (ICITST), 2011 International Conference for. (Dec 2011) 29–33
6. Maitra, S., Sarkar, P.: Cryptographically significant boolean functions with five valued walsh spectra. Theoretical Computer Science **276**(1) (2002) 133–146
7. Maitra, S.: Autocorrelation properties of correlation immune boolean functions. In: International Conference on Cryptology in India, Springer (2001) 242–253
8. Nawaz, Y., Gong, G., Gupta, K.C. In: Upper Bounds on Algebraic Immunity of Boolean Power Functions. Springer Berlin Heidelberg, Berlin, Heidelberg (2006) 375–389
9. Deb, S., Biswas, B., Kar, N. In: Study of NLFSR and Reasonable Security Improvement on Trivium Cipher. Springer India, New Delhi (2015) 731–739
10. Klapper, A., Goresky, M. In: Cryptanalysis Based on 2-Adic Rational Approximation. Springer Berlin Heidelberg, Berlin, Heidelberg (1995) 262–273
11. Klapper, A., Goresky, M.: Feedback shift registers, 2-adic span, and combiners with memory. Journal of Cryptology **10**(2) (1997) 111–147
12. El Hennawy, H.M., Omar, A.E., Kholaif, S.M.: Lea: link encryption algorithm proposed stream cipher algorithm. Ain Shams Engineering Journal **6**(1) (2015) 57–65
13. Khan, M.A., Khan, A.A., Mirza, F.: Transform domain analysis of sequences. CoRR arXiv:1503.00943 (2015)