Journal of Advances and Scholarly Researches in Allied Education
Vol. 16, Issue No. 9, June-2019, ISSN 2230-7540

Analysis and Design of LFSR Based Cryptographic Algorithm

Shabbir Hassan[1]* Mohammad Ubaidullah Bokhari[2]

[1,2] Department of Computer Science, Aligarh Muslim University, Aligarh, 202002, India

Abstract – Nowadays security has become a great concern in the field of computer science and information technology. In order to protect data from unintended users and to achieve a desirable level of security, several cryptographic algorithms based on various technology have been proposed. Linear Feedback Shift Register (LFSR) may play an important role in the design of such cryptographic algorithms. LFSR based cryptographic algorithms are often lightweight in nature and are more suitable for resource constrain devices. In this paper we present a detailed analysis of LFSR and design of m - sequence LFSR to implement cryptographic algorithms.

Keywords: Galois Field GF(p^m), Primitive Polynomial p(x), Primitive Polynomial p(x) over GF(p^m), LFSR, m - sequence, Run Length, Linear Recurrence, NIST.

----------X----------

1. INTRODUCTION

This paper represents a detail analysis of LFSR based techniques to design cryptographic algorithms. To secure data in a communication channel, several encryption algorithms and mathematical tools are used. All such algorithms can be easily implemented using mathematical tools that has been dealt in this paper. Modular arithmetic and equivalence classes are widely used to implement key exchange algorithms to establish a secure connection in the network. This paper gives an insight to all these mathematical tools with suitable examples. Due to the simplest structure and implementation of LFSR, it has been widely used in network communication and industries for generating Pseudo Random Sequences. Generally LFSRs are constructed by D Flip Flop and two input XOR gates. It can be implemented in two ways, first is Fibonacci implementation and other is D Flip Flop implementation. An LFSR with maximal length sequence output is called $m-sequence$ LFSR and is widely used in experimental design to get magnetic resonance imaging experiments, a pioneering non-invasive technology for studying the behaviour of human brain and to generate Pseudo Noise (PN) or a Pseudorandom Sequences (PS). The functional magnetic resonance imaging experiment has shown that the brain stimuli have correlated with the magnetic resonance scanner of the brain to collect functional MRI data for statistical analysis. An $m-sequence$ can also be used to justify the order appearance and timing of the brain stimuli. Due to this hopeful work, primitive polynomial and $m-sequence$ LFSRs have gained more acceptance in practice. In this paper we present Galois Field, Primitive Polynomial and Primitive Polynomial over Galois Field, LFSR and statistical inference of $m-sequence$ LFSR along with their related terminology.

2. GALOIS FIELD

A Galois Field generally denoted by GF is a set and is particularly useful in translating computer data as they are represented in the binary vectors. Since the elements of vector are the member of the finite set $S=\{0,1\}$, that are the element of Galois Field having 2 elements, also called Prime Field as shown in the Figure II. The Advance Encryption Standard (AES) utilizes the ideas of Galois Field. A Galois Field exists if and only if, it has p^m elements, where $p \in \mathbb{P}\ and\ m \in \mathbb{Z}^+, p$ is the *characteristic* of the Field, however order of the Field p^m represents the number of elements it has. Concept of Galois Field is used in our web browser to establish a secure connection on HTTPS [1]. LFSR perform its multiplication on Galois Field. The elements of the Galois Field $GF(p^m)$ is defined as:

$$GF(p^m) = (0, 1, 2, \dots, (p-1)) \cup$$
$$p, (p+1), (p+2), \dots, (p+p-1) \cup$$
$$p^2, (p^2+1), (p^2+2), \dots, (p^2+p-1) \cup$$
$$\cdot$$
$$\cdot$$
$$\cdot$$
$$p^{m-1}, (p^{m-1}+1), (p^{m-1}+2), \dots, (p^{m-1}+p-1)$$

The order of the Field is given by p^m while p is called the characteristic of the Field. From the above generalization, we can say that a Galois Field $GF(5)$ must have $(0,1,2,3,4)$ elements in it, where each element represents a polynomial of degree zero [2]. Concept of Galois Field is widely used in the Field of Cryptography. Since each byte is represented as a vector of a Finite Field, Encryption and Decryption using mathematical arithmetic are very simple [3].

3. PRIMITIVE POLYNOMIAL

The criterion for an irreducible polynomial to be a primitive is that "*a polynomial* $p(x)$ *over* $GF(2)[x]$ *of degree* m *is an irreducible if it has no factor of degree less than* m*, moreover is a factor of other polynomial* $P(x) = x^d - 1$*, where* d *is equivalent to* $2^m - 1$*"*. For example the polynomial $x^3 + x + 1$ is irreducible and have no any factor or factor polynomial $f(x)$ of degree less than 3, and also $x^3 + x + 1$ is a factor of polynomial $P(x) = x^7 - 1$ hence it is a *primitive polynomial* [3]. For any degree there must be a primitive polynomial.

A. Primitive Polynomial Over $GF(p^m)$

Consider polynomial $r(x) = x^2 + 1$ define over the domain real number $\mathbb{R}$, but its root does not lie in the domain of $\mathbb{R}$. However its root lies on the domain of complex number. Similarly a polynomial $p(x) \in GF(p)[x]$ doesn't have its roots in their Characteristic Field $GF(p)$ however it has its root in the Field $GF(p^n)$, this $GF(p^n)$ is called an "*Extension Field*" of $GF(p)$ [4]. Sometimes $GF(2)$ also referred as binary Field. Binary addition & multiplication are done by bit wise XOR and AND operation under $modulo(2)$ operation, and they satisfy commutative, associative and distributive law [3, 2]. Since XOR operation returns zero if both the operands are similar and returns one otherwise, this ensures that the addition & subtraction are same in Galois Field having Characteristic Field $GF(2)$. A polynomial $p(x)$ of degree n over Galois Field $GF(2)$ is symbolized as $p(x) \in GF(2)[x]$ and is defined as.

$$p(x) = a_n x^n, a_{n-1}x^{n-1}, \dots\dots, a_{n-r}x^{n-r}, \dots\dots\dots a_2 x^2, a_1 x, a_0$$

Where $a_i \in \{0, 1\}$. For any positive integer m, there are 2^m polynomials are possible each of degree m. For example the polynomials of degree 1 and 2 are depicted in Table II.

Table II.

Primitive Polynomial of Degree ≤ 2	
n	*Possible polynomials*
1	$x,\ x+1$
2	$x^2+x+1,\ x^2+x,\ x^2+1,\ x^2$

For any prime power q and any positive integer n, there exists a primitive polynomial of degree $n$over Galois Field$GF(q)$.

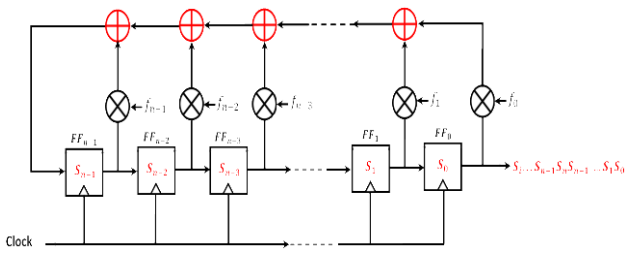
$$a_q(n) = \frac{\varphi(q^n - 1)}{n}$$

Primitive polynomials are widely used in Field element representation, Pseudo-Random bit generation and CRC codes. Primitive polynomials over $GF(2)$ are used for pseudorandom bit generation. In fact, every linear feedback shift register with maximum cycle length (which is $2^n - 1$, where n is the length of the linear feedback shift register) may be built from a primitive polynomial. For example, given the primitive polynomial $x^{10} + x^3 + 1$, we start with a user specified 10 bit seed occupying bit positions 1 through 10, starting from the least significant bit.

We then take the 10^{th} and 3^{rd} bits, and create a new 0^{th} bit, so that the XOR of the three bits is 0. The seed is then shifted left one position so that the 0^{th} bit moves to position 1 in each clock pulse [3, 7]. This process can be repeated to generate $2^{10} - 1 = 1023$ pseudo-random bits. In general, for a primitive polynomial of degree n over $GF(2)$, this process will generate a maximum of $2^n - 1$ pseudo random bits before repeating the same sequence, while non – primitive polynomial produces sequence less than $2^n - 1.$ One important property is to note that their reciprocal also form primitive polynomial (i.e. they come in pair). Example $1 + x^3 + x^4$ is degree 4, its reciprocal $1 + x + x^4$ i.e. 10011 and 11001, both are primitive. Technically, one can define primitive polynomial using concepts better from Finite Field Theory [5].

4. LINEAR FEEDBACK SHIFT REGISTERS

Linear Feedback Shift Registers (LFSR) is a complex combination of one bit memory element that comprises two mainly parts: (i) a clock storage element (Flip-Flop or *FF*) and (ii) a feedback path. The number of storage elements gives us the degree of the LFSR. The internal value of LFSR is called initial fill, initial vector or a seed and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current or previous states. *Output of the LFSR is one bit at each clock*. Likewise the register has a finite number of possible states, it must eventually enter a repeating cycle excluding all zeroes pattern, it will traverse the sequence exactly as before. The internal state bits are denoted by s_i and are shifted by one unit right with each clock pulse. The rightmost bit gives the current output whereas the leftmost bit is to be computed by a feedback function *f(x)* which is a XORsum of some *FF* values in previous state. Since XOR is a linear operation hence the circuit is called a *Linear Recurrence [11]*. Application of LFSR includes cycling through the addresses for refreshing a Dynamic Random Access Memory (DRAM), implementing CRC and many more. However it has some Problems for TPGs [9, 11].



A. Terminology Related with LFSR

Taps, period, internal state, Initialization Vector IV, lockup state etc. are the essential terminology associated with LFSR that are very important to discuss.

a. *Tap:* Lines that run from the output of one register within the LFSR into the XOR gates that determine input to the register within the LFSR.

b. *Period:* An m degree LFSR can produced a maximum of $2^m - 1$ distinct sequence of random number, and then it repeated the same sequence [10].

c. *Internal State:* At each clock pulse, all the bit are shifted towards MSB from LSB, and then XOR bit is feed into the LSB register of LFSR [10, 12].

d. *Seed:* The initial value of the LFSR is called a seed [13].

e. *Lockup State:* All zero initial values or seed stuck up the LFSR into an unrecoverable state, and would never leave this state [9, 11].

B. Types of LFSR

Finally for every primitive polynomial there are in fact 4 linear feedback shift register may be implemented either by using XOR gated in series with each FF output, or with the XOR gate external to the shift register in the feedback path. The external XOR LFSR is called Standard LFSR or Type-I LFSR or External LFSR as shown in the Figure I. The internal XOR LFSR is called Modular LFSR or Type-II LFSR or Internal [13].

a. *Standard LFSR*

Figure shows ^{n} stage standard LFSR. It consists of ^{n} FF and a number of XOR gates. Since XOR gate are placed on the external feedback path, so it is also referred as *external XOR LFSR as* shown in Figure III [12, 15].

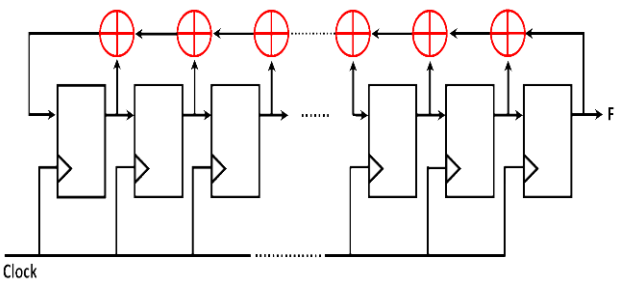


Figure. III

b. Modular LFSR

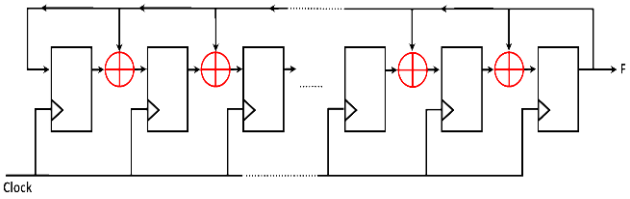


Figure. IV

Similarly an n stage modular LFSR with each XOR gate placed between two adjacent *FF* is shown in Figure, and is called an *internal XOR LFSR*, because each stages are introduced at most one XOR gate delay [6]. It has higher clock frequency than Standard LFSR as shown in Figure. IV [14]. The sequence generated by the Type-1 and Type-2 LFSR are totally different even they seeded with the same initialization vector as shown in the Figure below.

Following Figure 5.(1,3) External and Figure 5.(1,3) Internal represents the design of 3 stage Standard and Modular LFSR as *m-sequence* PNRG that gives output of all $2^3 - 1$ deterministic states.

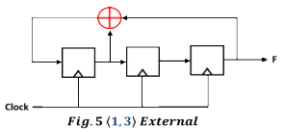


Fig. 5 (1, 3) External

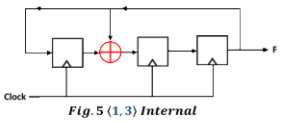


Fig. 5 (1, 3) Internal

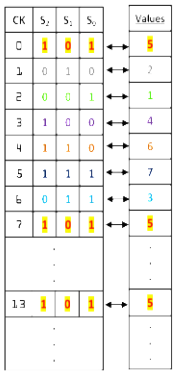


Fig. 5 (1, 3) External

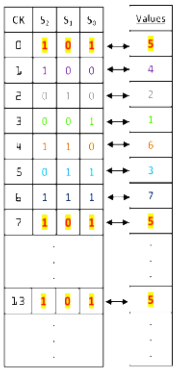


Fig. 5 (1, 3)Internal

5. DESIGN OF *m-sequence* LFSR

An LFSR of size m can result in producing all feasible states throughout the period p which is equal to $2^m - 1$ shift, but it will achieve this period only when appropriate feedback paths have been chosen. For example, an 8 stage LFSR would probably possess a widest possible combination of 1s and 0s after reaching at 255 shifts. Each sequence produced in this shift is a maximal sequence, in general a maximum length sequence. These sequences usually are referred as $m - sequence$ or Pseudo Noise (PN) or Pseudorandom Sequences (PS) [7, 8]. Maximal length generators can in fact produce two sequences. The first has a length of one, and occurs when the initial state of the generator is set to all zeros. The other one has a length of 2^m–1. Together, both of these two sequences keep track of all 2^m states of a m¯ bit state register. Once the feedback taps of an LFSR are non-maximal, the length of the generated sequence relies on the initial state of the LFSR. Each of these sequences is called a State Space of the LFSR. Since an *m-sequence* LFSR gives all possible runs of 0s and 1s with unique combinations of bits, this deterministic sequence holds *Balance, Run Length and Phase Shift* properties and grows *Asymptotically Large [11, 12]*. Thus we can implement a key exchange algorithms or LFSR based cryptographic algorithms that follow all the NIST standard [9].

6. CONCLUSION

The paper presented a brief analysis and design of LFSR based *m-sequence* PRNG that are very useful to implement cryptographic algorithms. Some important mathematical tools such as Field, Galois Field, Primitive Polynomial and Primitive Polynomial over Galois Field and LFSR have been discussed. The paper describes a design of *m-sequence* PRNG that follows the security measures as described by NIST standard. In future work, it is proposed to implement LFSR based cryptographic algorithms that are used in Key Exchange and Data Encryption by using the said mathematical tools.

7. REFERENCES

[1] National Institute of Standards and Technology, Advanced Encryption Standard, FIPS 197 (2011).

[2] Fernandes, Rebecca Angela, and Niju Rajan (2008). "Power Optimization of Linear Feedback Shift Register (LFSR) using Power Gating." Power 5.05.

[3] D. A. Cox (2012). Galois Theory, 2nd ed., Wiley, Hoboken.

[4] D. A. Cox, Evariste Galois and Solvable Permutation Groups, http://www.cs.amherst.edu/~dac/ lectures/bilbao.pdf.

[5] G. Frei (2007). The Unpublished Section Eight: On the Way to Function Fields over a Finite Field, pp. 159–198 in "The Shaping of Arithmetic after C. F. Gauss's Disquisitiones Arithmeticae," ed. C. Goldstein, N. Schappacher, J. Schwermer, Springer-Verlag, Berlin, 2007.

[6] E. H. Moore (1896). A Doubly-Infinite System of Simple Groups, pp. 208–242 in "Mathematical papers read at the International Mathematical Congress held in connection with the World's Columbian Exposition, Chicago, 1893," Macmillan & Co., New York, 1896.

[7] Mashhadi, Samaneh, and Massoud Hadian Dehkordi (2015). Two verifiable multi secret sharing schemes based on nonhomogeneous linear recursion and LFSR public-key cryptosystem." Information Sciences 294: pp. 31-40.

[8] Tan, Zuxiong, et. al. (2018). A New Pseudo-Random Number Generator Based On The Leap-Ahead LFSR Architecture." 2018 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA). IEEE, 2018.

[9] Mo, Hongjia, and Michael Peter Kennedy (2016). "Influence of Initial Conditions on the Fundamental Periods of LFSR-Dithered MASH Digital Delta–Sigma Modulators with Constant Inputs." IEEE Transactions on Circuits and Systems II: Express Briefs64.4: pp. 372-376.

[10] Tzanakis, Georgios, et. al. (2016). "Constructing new covering arrays from LFSR sequences over Finite Fields." Discrete Mathematics 339.3 (2016): pp. 1158-1171.

[11] Panda, Amit Kumar, Praveena Rajput, and Bhawna Shukla (2012). FPGA implementation of 8, 16 and 32 bit LFSR with maximum length feedback polynomial using VHDL." 2012 International Conference on Communication Systems and Network Technologies. IEEE, 2012.

[12] Ahmad, Afaq, Sayyid Samir Al-Busaidi, and Mufeed Juma Al-Musharafi (2013). "On Properties of PN Sequences generated by LFSR a Generalized Study and Simulation Modeling." Indian Journal of Science and Technology 6.10 (2013): pp. 5351-8.

Corresponding Author

Shabbir Hassan*

Department of Computer Science, Aligarh Muslim University, Aligarh, 202002, India

hassan.analyst@gmail.com