Cryptographically Secure Linear feedback shift Register

Deeksha Sharma, Abdul Khalid, Shradha Parashar

Abstract— **True random bit generator requires a naturally occurring source of randomness. Designing a hardware device or software program to exploit this randomness and produce a bit sequence that is free of biases and correlation is difficult task. For most cryptographic applications, the generator must not be subject to observation or manipulation by an adversary. So pseudorandom bit generator (PRNG) is used to create a sequence of bits that appear to random but not exactly random. As the word 'pseudo' suggest, these are not random some algorithms that use mathematical formulae or simply pre-calculated tables are used to produce sequence of number that appear random. A cryptographically secure pseudo random number generator (CSPRNG) is a PRNG with properties that make it suitable for use in cryptography. In this paper a modified version of linear feedback shift register (LFSR) is generated which will satisfy the requirement of cryptographically secure PRNG.**

***Index Terms*—About four key words or phrases in alphabetical order, separated by commas.**

I. INTRODUCTION

Generating random number is an essential task in cryptography. A PRNG is a program written for, and used when large quantities are needed. Random numbers are necessary not only for generating cryptographic keys but also are needed in steps of cryptographic algorithms or protocols (e.g. initialization vectors for symmetric encryption, password generation, nonce generation. One time pads, salts in certain signature schemes etc). These pseudo-random sequence can be used in a large variety of applications including multiple access and polling techniques, secure and private communications, error detecting and correcting codes, and cryptographic system [1]. The requirement of an ordinary PRNG is also satisfied by a CSPRNG, but the reverse is not true [2]. CSPRNG requirements fall into two categories [3]-

a) They must pass statistical randomness tests; and
b) They hold up under serious attack.

The security of the scheme depends on the quality of the PRNG. If a user has access to a string, he can use a deterministic or cryptographic PRNG to expand the seed into a longer sequence which is distributed in the application. However, in many situations, it is unrealistic to assume that user have access to secret randomness. An LFSR is a mechanism for generating a sequence of binary bits. The register consists of a series of cells that are set by an initialization vector that is, most often, the secret key. The behavior of the register is regulated by a clock and at each clocking instant, the contents of the cells of the register are shifted left by one position, and the exclusive-or of a subset of the cell contents is placed in the rightmost cell. A basic LFSR consists of 3 components: the input sequence (initialization vector), the feedback (tap sequence) and the output. As the name suggests, the feedback gives a linear relationship between the input and the output.

A LFSR is a register of bits that performs discrete step operation that

- Shift all the bits one position to the left and
- Replace the vacated bit by the modulo two addition of the bit shifted off and the bit at a given tap position in the register

A general linear feedback shift register is shown in figure 1 with tap position 12,11,10,2.

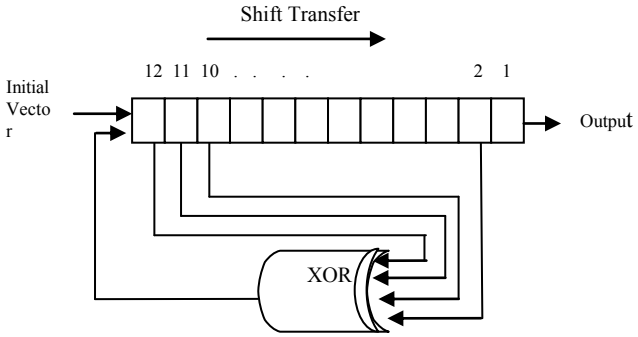

Figure 1: General 12 bit linear feedback shift register.

Let the input sequence of length n be (s0, s1, …, sn-1). The feedback is thus a linear function f (s0, s1, …, sn-1) defined by

$$f(s) = \sum_{i=0}^{n-1} c_i s_i$$

where co, c1, …, cn are constant coefficients. The output of the LFSR is determined by the initial values s0, s1, …, sn-1 and the linear recursion relationship:

$$s_{k+n} = \left(\sum_{i=0}^{n-1} c_i s_{i+k} \right), k \geq 0$$

or equivalently,

***Deeksha Sharma**, Computer Science Department, Noida institute of engineering and technology Gr.Noida, India, +91 9650908333.*
***Abdul Khalid**, Computer Science Department, Noida institute of engineering and technology Gr.Noida, India, +919911666034.*
***Shradha Parashar**, Computer Science Department, Institute of Technology & Management, Aligarh, India, +919548629352.*

$$\sum_{i=0}^{n} c_i s_{i+k} = 0, \; k \geq 0$$

where cn = 1 by definition [5]

This LFSR has some disadvantages. In practice, the output exhibit artifacts which cause them to fail statistical pattern-detection tests [3]. These include:

- Shorter than expected periods for some seed states (such seed states may be called 'weak' in this context);
- Lack of uniformity of distribution for large amounts of generated numbers;
- Correlation of successive values;
- Poor dimensional distribution of the output sequence;
- The distances between where certain values occur are distributed differently from those in a random sequence distribution.

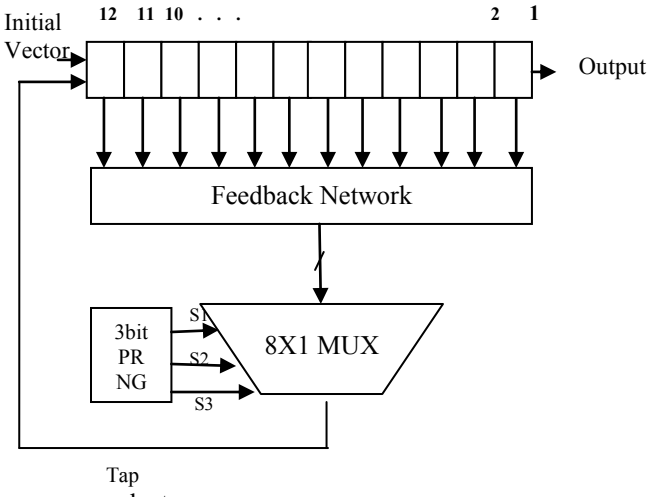

Figure 1: General 12 bit linear feedback shift register.

II. PROPOSED METHODOLOGY

In an LFSR, the bits contained in selected positions in the shift register are combined in some sort of function and the result is fed back into the register's input bit [3]. An Initial Vector (IV) is inserted as a seed to generate the pseudo random numbers. A sight change in IV will change the entire sequence of numbers. By definition, the selected bit values are collected before the register is clocked and the result of the feedback function is inserted into the shift register during the shift, filling the position that is emptied as a result of the shift.

The feedback function in an LFSR has several names: XOR, odd parity, sum modulo 2. Whatever the name, the function is simple: 1) Add the selected bit values, 2) If the sum is odd, the output of the function is one; otherwise the output is zero.

The bit positions selected for use in the feedback function are called "taps". The list of the taps is known as the "tap sequence". By convention, the output bit of an LFSR that is 12 bits long, the feedback tapping are kept changing. This changing is done by a 3bit LFSR which works as a selector of 8X1 mux which make the generated code quite complex.
The following tables contain m-sequence feedback sets for LFSR,

Selection	LFSR Tapping
1	[12, 11, 10, 4]
2	[12, 11, 10, 2]
3	[12, 11, 8, 6]
4	[12, 11, 7, 4]
5	[12, 10, 9, 3]
6	[12, 10, 5, 4]
7	[12, 9, 8, 5]

Table1 : Tapping Sequences

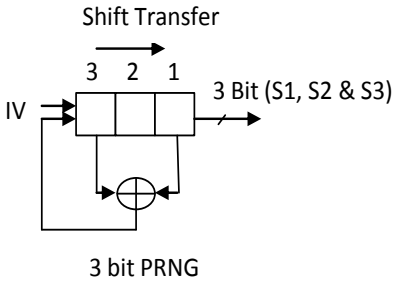

Figure 1: General 12 bit linear feedback shift register.

Simple Example
Step 1: Let we assume a register 'S' with IV 111000101011

12	11	10	09	08	07	06	05	04	03	02	01
1	1	1	0	0	0	1	0	1	0	1	1

Step 2: Let Assume 3bit PRNG generate 101 bit sequence. Then S1=1, S2=0 and S3=1 works as selector to 8X1 MUX.

Step 3: S1,S2 and S3 will select 5th (as (101)2=5) tap position from the tap selection table. 12, 10, 9, 3.

12	11	10	09	08	07	06	05	04	03	02	01
1	1	1	0	0	0	1	0	1	0	1	1

Step 4: Bit XOR the selected positions (1 XOR (1 XOR (0 XOR 0))) = 0.

Step 5: Selected bit is feed back at 12th position and whole LFSR will shift right.

Step 6: The current position of LFSR will generate the output.

Step 7: Repeat the Process until desired numbers are generated.

ISSN: 2278 – 1323

III. EXPERIMENTAL RESULTS

Some tests are performing on the 10000 sample of random numbers. For this purpose a NIST test suit is used. Some of the successful tests are shown as follows.

a) Test for the Longest Run of Ones in a Block

Description: The focus of the test is the longest run of ones within M-bit blocks. The purpose of this test is to determine whether the length of the longest run of ones within the tested sequence is consistent with the length of the longest run of ones that would be expected in a random sequence [6]. Note that an irregularity in the expected length of the longest run of ones implies that there is also an irregularity in the expected length of the longest run of zeroes. Long runs of zeroes were not evaluated separately due to a concern about statistical independence among the tests.

b) Non Overlapping Template Matching Test

Description: The focus of this test is the number of occurrences of pre-defined target substrings. The purpose of this test is to reject sequences that exhibit too many occurrences of a given non-periodic (aperiodic) pattern [7]. For this test and for the Overlapping Template Matching test, an m-bit window is used to search for a specific m-bit pattern. If the pattern is not found, the window slides one bit position. For this test, when the pattern is found, the window is reset to the bit after the found pattern, and the search resumes.

c) Lemel Ziv Complexity Test

Description: The focus of this test is the number of cumulatively distinct patterns (words) in the sequence. The purpose of the test is to determine how far the tested sequence can be compressed [8]. The sequence is considered to be nonrandom if it can be significantly compressed. A random sequence will have a characteristic number of distinct patterns.

d) Test for frequency within a Block

Description: The focus of the test is the proportion of zeroes and ones within M bit blocks. The purpose of this test is to determine whether the frequency of ones is an M-bit block is approximately M/2.

e) Random Excursion variant test

Description: The focus of this test is the number of times that a particular state occurs in a cumulative sum random walk. The purpose of this test is to detect deviations from the expected number of occurrences of various states in the random walk.

f) Randomness

Description: Randomness means lack of pattern or predictability in events [6] Randomness suggests a non-order or non-coherence in a sequence of symbols or steps, such that there is no intelligible pattern or combination. The LFSR generates 2ℓ (where ℓ is the bit length of the shift register) in this case ℓ=12 states always occur in a 12 bit LFSR. Here 7 different taps are used, so 212 X 7= 28672 times after the 1st number will generate.

IV. CONCLUSION

In modern cryptography, a number of elementary building blocks like block ciphers, stream ciphers, or hash functions are used. Stream ciphers are often based on pseudorandom generators (PRGs) that are used to transform a small initial value into a long sequence of seemingly random bits. Many PRG designs are in turn based on linear feedback shift registers (LFSRs), which can be constructed in such a way as to have optimal statistical and periodical properties. In order to understand the security needs of a cryptographic building block, it is unavoidable to delve into cryptanalysis, which is the activity of searching for security weaknesses of cryptographic algorithms. The underlying goal of cryptanalysis is not destructive, but constructive: Only by improving the understanding of possible problems, it is possible to propose new design criteria for cryptographic systems. Thus, this paper discussed both construction principles and cryptographically secure LFSR-based PRNGs. In Introduction part we understand a general LFSR, its functioning and disadvantage. Next we introduced an extension to a general 12 bit LFSR. Random use of different taps makes LFSR more unpredictable and cryptographically secure.

REFERENCES

[1] Solomon W. Golomb, shift Register Sequences. Aegean Park Press Laguna Hills, CA, USA 1981.
[2] Huang, Andrew (2003). Hacking the Xbox: An Introduction to Reverse Engineering. No Starch Press Series. No Starch Press. p. 111.ISBN 9781593270292.
[3] http://www.random.org/randomness/
[4] Manikandan at el. " Design & verification of prbs for maximal length using vhdl "International Journal of Advances in Electrical & Electronics Engineering (IJAEEE) ISSN 2278-8948, Volume-2, Issue-5 page 28, 2013
[5] http://www-math.cudenver.edu/~wcherowi/courses/m5410/m5410fsr.html pp 1 – 8, 09 January 2002.
[6] Atsushi Uchida "Optical Communication with Chaotic Lasers: Applications of Nonlinear Dynamics and Synchronization" Wiley publishers, page 507.
[7] Juan Soto and Lawrence Bassham "Randomness Testing of the Advanced Encryption Standard Finalist Candidates", Computer Security Division National Institute of Standards and Technology 100 Bureau Drive, Stop 8930 Gaithersburg, MD 20899-8930.
[8] Song-Ju Kim, Ken Umeno, and Akio Hasegawa, "Corrections of the NIST Statistical Test Suite for Randomness", Chaos-based Cipher Chip Project, Presidential Research Fund, Communications Research Laboratory, Incorporated Administrative Agency.

Deeksha Sharma was born in India in August 1990. She has completed graduation in Information Technology from Aligarh college of Engineering and Technology, Aligarh India in 2011 and pursuing post graduate degree in Software Engineering from Noida institute of engineering and technology Gr.Noida.

Her interest areas are Advanced Digital Signal Processing, Security Application, and Digital Image Processing.

Abdul Khalid was born in India in March 1975. He has completed graduation in Computer Engineering from Galgotia College of engineering and technology Gr.Noida, India in 2006 and post graduate degree in Computer Engineering from Galgotia College of engineering and technology Gr.Noida in 2008. He joined the Department of computer Science & Engineering at Noida institute of engineering and technology Gr.Noida, as assistant Professor and has more than 8 years of experience in teaching. He has published 2 papers in National and International Conferences and Journals.

His interest areas are Software Engineering, Genetic Algorithm, and Software Reliability.

Shradha Parashar was born in India in June 1988. She has completed graduation in Computer Engineering from Aligarh college of Engineering and Technology, Aligarh India in 2009 and post graduate degree in Computer Engineering from Zakhir Hussain college of Engineering and Technology, AMU, Aligarh. She joined the Department of computer Science & Engineering at ITM , Aligarh, as a assistant Professor and has more than 3 years of experience in teaching. She has published 5 papers in National and International Conferences and Journals.

Her interest areas are Advanced Digital Signal Processing, Security Application, and Digital Image Processing.

ISSN: 2278 – 1323