A New Approach to LFSR based Pseudorandom Numbers Generator

Aye Myat Nyo
Ph.D Candidate
University of Computer Studies, Mandalay
Mandalay, Myanmar
ayemyatnyo81@gmail.com

Prof. Dr. Than Naing Soe
Computer University, Myitkyina
Myitkyina, Myanmar
Konaing2006@gmail.com

***Abstract*— Creating the common security basis is one of the most important tasks for all security system.Many cryptographic protocols require random or pseudorandom inputs at various points.According to their security level, pseudorandom number generators (PRNGs) can be used in many other applications such as Monte-Carlo simulation, spread spectrum communication, radar ranging, randomized algorithms and cryptography. The strength of many cryptosystems relies on the qualities of PRNGs. Generation high-quality randomness PRNG is an important part of many cryptographic operations. PRNG uses one or more inputs and generates multiple “pseudorandom” sequences. In this paper, PRNG is designed by applying linear feedback shift register (LFSR) and Random-box (R-box).The seed value directly affect the quality of PRNG’s output sequences, so reseeding by using R-box is proposed.Randomness properties of PRNG depend on tag sequence.To improve the randomness properties in proposed design of PRNG, selection of optimal primitive polynomial is developed. To measure the quality of proposed PRNG, some statistical tests provided by NIST (National Institute of Standards and Technology) Test Suite were used.**

Keywords*— *PRNGs, R-box, NIST

I. INTRODUCTION

Random numbers play an important role in the use of encryption for various network security algorithms and protocols.Many cryptographic primitives require random numbers, to be used as keys, challenges, unique identifiers, etc [3]. One possible method for such generation is to use a pseudo-random numbers generator.

The need for random and pseudorandom numbers arises in many cryptographic applications. Many cryptosystems were vulnerable to particular attack due to the weakness of applied PRNGs. Moreover, there are too many test suits that are intended to test the randomness of the PRNGs [1].

A pseudo-random number generator (PRNG) is a function that, once initialized with some random value (called the *seed*), outputs a sequence that appears random. Generation high-quality randomness PRNG is a vital part of many cryptographic operations. Inputs to PRNGs are called ***seeds***, in contexts in which unpredictability is needed.The outputs of a PRNG are typically deterministic functions of the seed.The resulting sequences will pass many reasonable tests of randomness.

A linear feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state. Mostly used linear function of single bits is XOR, thus normally it is a shift register whose input bit is driven by the exclusive-or (XOR) of some bits of the overall shift register value. The initial value of the LFSR is called the 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) state.

II. RELATED WORK

There are many pseudorandom number generators. Among them, Linear Feedback Shift Register (LFSR), Lagged Fibonacci generator, Linear congruential generator, RC4 PRNG, Modular Exponentiation (MODEXP), Mother-of-All, RANROT, Mersenne Twister and Blum-Blum-Shub (BBS) are well-known generators.

Blum-Blum-Shub generator is computed via $S_{n+1}= (S_n^2)$ mod m. Then the output is some function on S_{n+1}, which often is taken as bit parity or some particular bits of $S_{n+1.}$

A fuzzified approach to generation of cryptographically secure Pseudo-random numbers [6] is a fuzzy based adaptive algorithm for the reseeding operation of Fortuna.A5/1 is based around a combination of three linear feedback shift registers (LFSRs) with irregular clocking.

"A New Reseeding Technique for LFSR-based Test Pattern Generation" [2] developed the algorithm into two parts: the first part describes the determination of primitive polynomials. The second part describes the generation of m-sequences based upon the primitive generator polynomial.Development of algorithm for the generation and correlation study of maximal length sequences for applicabilities in CDMA mobile communication systems [5]developed an algorithm for the generation of m-sequence of length 7 to 255 and also studied the correlation properties of the generated m-sequences using MATLAB.

III. PSEUDORANDOM NUMBER GENERATOR (PRNG)

A pseudorandom number generator is a cryptographic algorithm used to generate sequence of numbers that must appear randomly. In other words, a pseudorandom number generator (PRNG) is an algorithm for generating a sequence of numbers that approximates the properties of random numbers.Typically a pseudorandom number generator uses one or more inputs and generates multiple pseudorandom sequences that are approximately independent of each other. Pseudorandom number generators are widely used in such applications as computer modeling (e.g., Markov chains), statistics, experimental design, cryptographic approach, etc.

A. Linear Feedback Shift Registers

A feedback shift register is made up of two parts: a shift register and a feedback function. The shift register is initialized by loading the key into it. The outputs that influence the input are called taps. The tap sequence of an LFSR can be represented as a polynomial called the feedback polynomial. The new left-most bit is computed as a function of the other bits in the register and then loop forever output the lowest bit of the register and shift the register right one bit .

An n-bit LFSR can be $2^n - 1$ internal state. In order for a particular LFSR to be a maximal-period LFSR, the polynomial formed from a tap sequence should be chosen carefully.

B. Taps Sequence grant Maximal Period

A maximal period tap sequence also describes the exponents that are known as a primitive polynomial.

Primitive polynomial tap sequence (4, 1) = x^4+x^1+1 Properties of Primitive polynomial is to have maximal length tap sequences always have an even number of taps. The tap values in a maximal length tap sequence are all relatively prime. Example, A tap sequence like 12,9,6,3 will not be maximal length because the tap values are all divisible by 3.

IV. PROPERTIES OF PRNG

A PRNG is its own kind of cryptographic primitive and the better understanding of these primitives will make it easier to design and use PRNGs securely [2]. A PRNG is a single point of failure for many real-world cryptosystems. Many systems use badly-designed PRNGs, or use them in ways that make various attacks easier [6]. So random and pseudorandom numbers generated for cryptographic applications should be unpredictable.

Criteria for good PRNG are:

- The numbers generated are uniformly distributed.
- The numbers generated are statistically independent.
- The numbers generated are easily reproducible (so one can replicate a study).
- The sequence is fast to compute.
- Output sequences of PRNGs should have maximal period.
- They should have backward and forward unpredictability.

A. Random Numbers Testing

Random and pseudorandom numbers generated for cryptographic applications should be unpredictable. Randomness is a probabilistic property. The properties of a random sequence can be characterized and described in terms of probability. Because there are so many tests for judging whether a sequence is random or not .No specific finite set of tests is deemed "complete".

V. PRINCIPLE OF R-BOX CONSTRUCTION

Inputs of Random-box are A, B, H (Hash table) and Addr (Address table).

$$A=Addr(A) \qquad (1)$$

$$B=(A+B)\text{mod } 256 \qquad (2)$$

$$Output=H(B) \qquad (3)$$

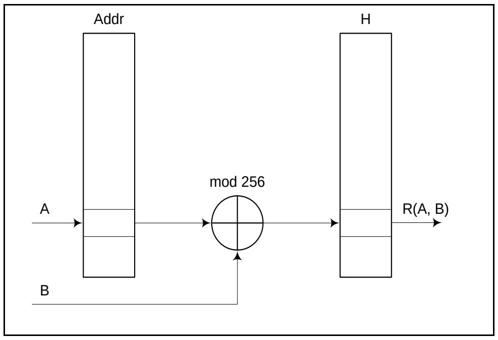
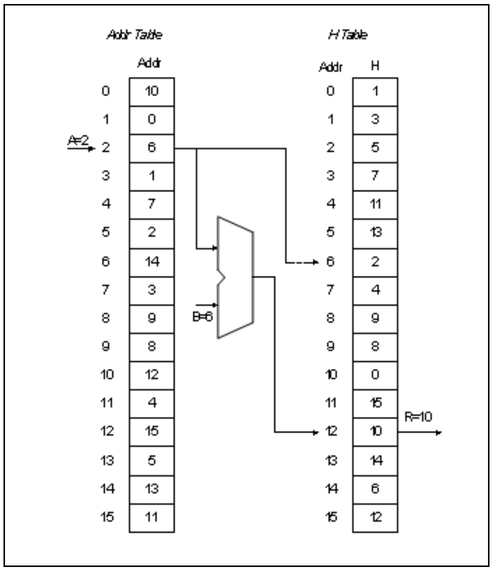


Fig. 1. Principle of R-box construction



	Addr
0	10
1	0
2	6
3	1
4	7
5	2
6	14
7	3
8	9
9	8
10	12
11	4
12	15
13	5
14	13
15	11

Addr	H
0	1
1	3
2	5
3	7
4	11
5	13
6	2
7	4
8	9
9	8
10	0
11	15
12	10
13	14
14	6
15	12

Fig. 2. Example of R-box construction

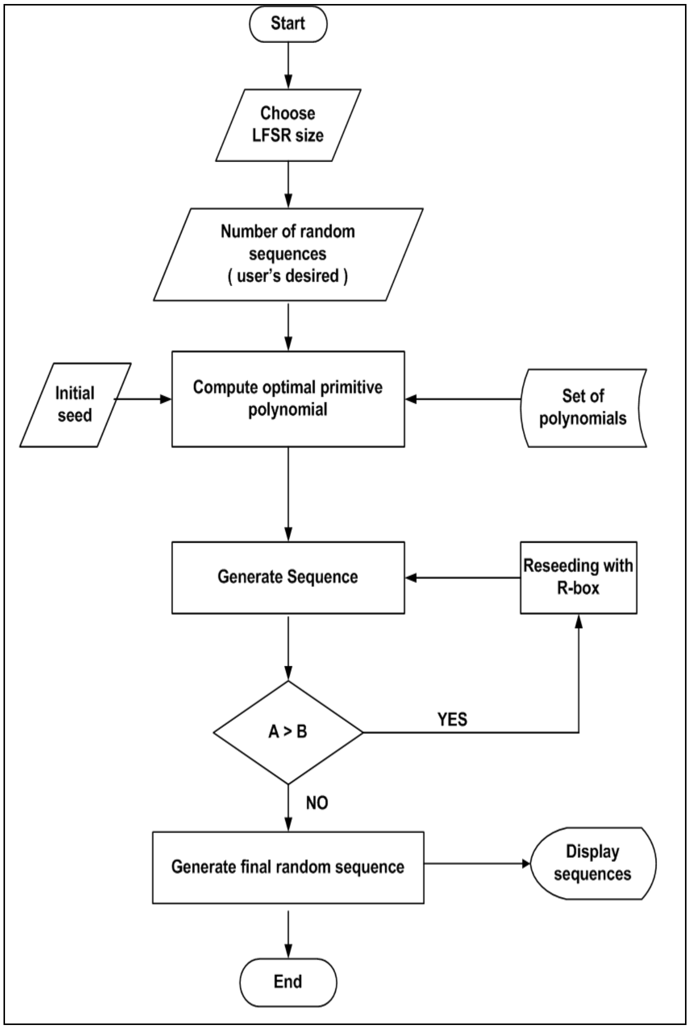
H table are initialized by initialization vector therefore $R_H(A, B) = H((m_A + B) \text{ mod } 2)$. A, B are input elements of registers. m_A is the address of a cell of the A in H-Table therefore $H(m_A) = A$.

VI. PROPOSE PRNG

The objectives of proposed PRNG are to develop an efficient PRNG to provide maximal period and randomness properties and to analyze the quality of random sequences of proposed PRNG by using some well known tests of NIST.

Proposed Effective LSFR based PRNG reseeding operation by applying Random-box is considered in PRNG. We find optimal primitive polynomial which is from set of primitive polynomials. Effort to evaluate the quality of random sequence, we will use some tests from the statistical package of NIST Test Suite.

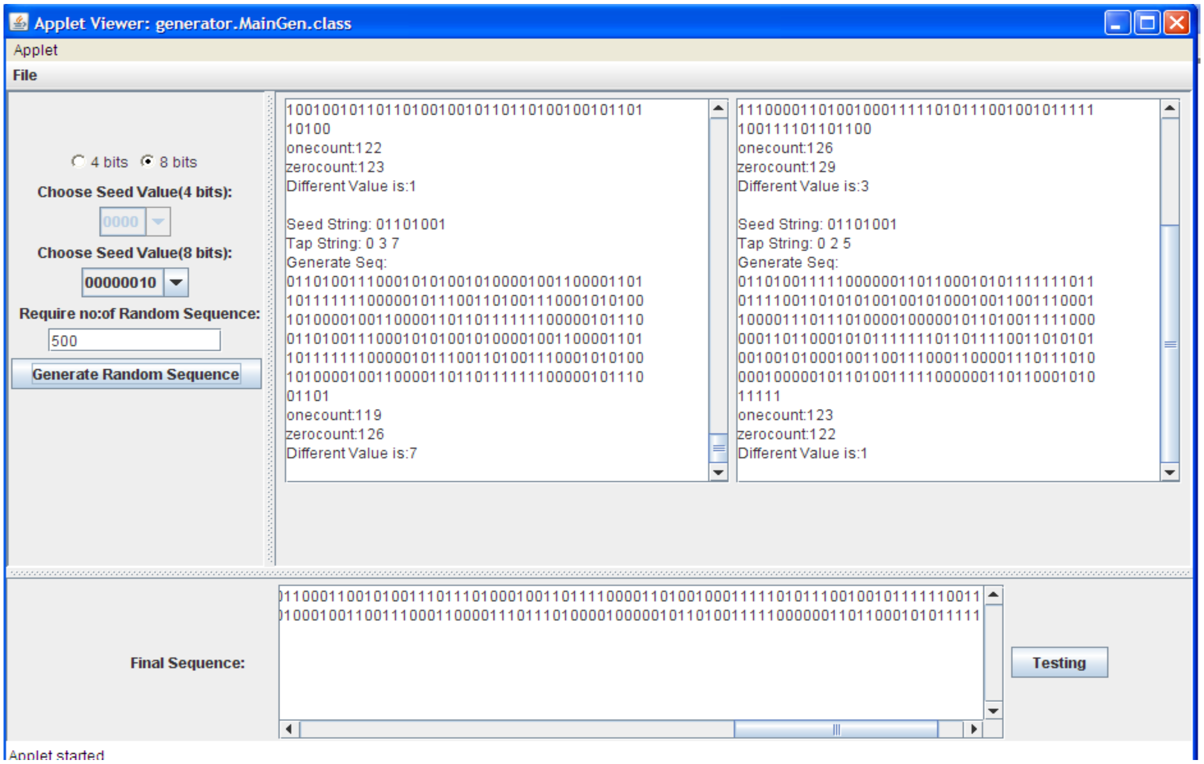
For using my PRNG, user can choose length of register (4 bit or 8 bit) and then user can pay desired length of random sequence. By using user selected seed value, the system can compute which primitive polynomial is appropirate for balance of number of occurance of 1and 0 in this random sequence. When the number of user required sequence is greater than LFSR's maximal period we must use R-box to reseed the LFSR so we can get unrepeated random sequence from previous sequence.



A=user's required random sequences

B=LFSR's maximal period

Fig. 3. Flowdiagram of proposed PRNG



○ 4 bits ● 8 bits

Choose Seed Value(4 bits): 0000

Choose Seed Value(8 bits): 00000010

Require no:of Random Sequence: 500

```
1001001011011010010010110110100100101101
10100
onecount:122
zerocount:123
Different Value is:1

Seed String: 01101001
Tap String: 0 3 7
Generate Seq:
0110100111000101010010100001001100001101
1011111110000010111001101001110001010100
1010000100110000110110111111100000101110
0110100111000101010010100001001100001101
1011111110000010111001101001110001010100
1010000100110000110110111111100000101110
01101
onecount:119
zerocount:126
Different Value is:7
```

```
1110000110100100011111010111001001011111
10011110110110 0
onecount:126
zerocount:129
Different Value is:3

Seed String: 01101001
Tap String: 0 2 5
Generate Seq:
0110100111110000001101100010101111111011
0111100110101010010010100010011001110001
1000011101110100001000001011010011111000
0001101100010101111111011011110011010101
0010010100010011001110001100001110111010
0001000001011010011111000000110110001010
11111
onecount:123
zerocount:122
Different Value is:1
```

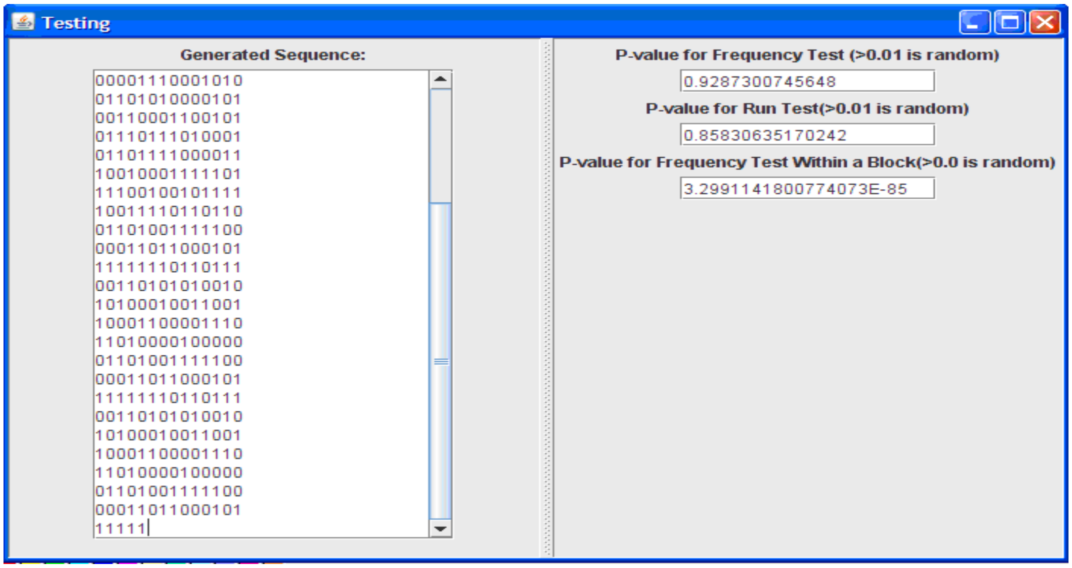
Final Sequence:

```
0110001100101001110111010001001101111000011010010001111101011100100101111110011
0100010011001110001100001110111010000100000101101001111100000011011000101011111
```

Fig. 4. Frame of proposed PRNG

In above example, 8 bits register is used to generate 500 number of pseudorandom numbers. User can get 255 number of random sequence because maximal length of 8 bits register is 255.And then need to change seed value to get another random sequence that is non repeat to previous random sequences.

After genetating user required number of pseudorandom numbers, some statistical tests provided by NIST Test Suite such as Frequency Test, Run Test and Frequency Test within a Block Test were used to measure the quality of proposed PRNG.



Generated Sequence:

```
00001110001010
01101010000101
00110001100101
01110111010001
01101111000011
10010001111101
11100100101111
10011110110110
01101001111100
00011011000101
11111110110111
00110101010010
10100010011001
10001100001110
11010000100000
01101001111100
00011011000101
11111110110111
00110101010010
10100010011001
10001100001110
11010000100000
01101001111100
00011011000101
11111
```

P-value for Frequency Test (>0.01 is random)

0.9287300745648

P-value for Run Test(>0.01 is random)

0.85830635170242

P-value for Frequency Test Within a Block(>0.0 is random)

3.2991141800774073E-85

Fig. 5. Testing of proposed PRNG

VII. CONCLUSION

Developing a good, efficient algorithm for generating pseudorandom numbers on a computer is a very difficult problem. A way to destroy the linearity in LFSR and construction of effective pseudorandom number generator is by using dynamic s. Non linear generators have less regular structure which is much harder to analyze. Proposed LFSR's reseeding operation that is based on stochastic transformation which is called R-box to get maximal period of each register and seed value. It is more efficient and secure than LFSR to generate random sequences because it can get nonrepeat random sequences.

References

[1] A statistical Test Suit for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST Special publication 800-22,May 15,2001.

[2] A. Srinivasan and D. M. Ceperley,Random Number Generators for Parallel Application.

[3] E. Kalligeros, X. Kavousianos , D. Bakalis and D. Nikolos,A New Reseeding Technique for LFSR-based Test Pattern Generation.

[4] P. Ekdahl,On LFSR based Stream Ciphers

[5] 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.

[6] S. Wang, K. J. Balakrishnan. S. T. Chakradhar,Efficient Unknown Blocking Using LFSR Reseeding.

[7] M. A.Akbar and M. Z. Khalid, Fuzz-Fortuna: A fuzzified approach to generation of cryptographically secure Pseudo-random numbers.