Statistical Evaluation of LFSR-Based Stream Ciphers

Nwe Ni Than, Tin Win Maw
Computer University, Taung-Ngu, Myanmar
nweni.than02@gmail.com,tinwinmaw2009 @gmail.com

Abstract

The need for random and pseudorandom numbers arises in many cryptographic applications. Generating high-quality randomness is a vital part of many cryptographic operations. This paper presents some aspects of selecting and testing random and pseudorandom number generators. The outputs of such generators may be used in many cryptographic applications, such as the generation of key material. In this paper, the key generators of A5/1 and Grain-80 stream cipher algorithms are tested by using statistical tests. The strength of A5/1 and Grain-80 stream cipher are evaluated and compared using statistical tests which are Frequency (Monobit) Test, Runs Test and Serial Test to test the randomness of these ciphers.

Keywords: pseudorandom, A5/1, Grain, NIST.

1. Introduction

In today's world of modern computers and ubiquitous networks most security solutions rely on the use of cryptography. Generating high-quality randomness is a vital (and maybe most difficult) part of many cryptographic operations and the importance of a careful design of cryptographic (pseudo)random number generators cannot be underestimated. Following the second Kerckhoff's Principle – the security of a cryptosystem shall not be based on keeping the algorithm secret but solely on keeping the key secret – the quality and unpredictability of secret data is critical [5].

Many classes of pseudorandom number generators exist, but the goal of a pseudorandom number generator in cryptography is production of pseudorandom data that are computationally indistinguishable from statistically ideal random data.

Pseudo-random number generators are often based on cryptographic functions like block ciphers or stream ciphers. Communication system should be supported with stream cipher to keep information safe. This is because stream cipher for bulk transmission of data such as text due to faster implementation and not introduction error propagation [1].

A5/1 and Grain, well-known stream ciphers are based on the linear feedback shift register (LFSR) for wireless communication. LFSR is the heart of any digit system that relies on pseudorandom bit sequence (PRBS). It is one of the most useful techniques for generating pseudorandom generators. A pseudo-random number generator produces a sequence of bits that has a random looking distribution.

The system is implemented to test whether the given input sequence is a random or not. The Section 2 is related works and the proposed system overview is presented in Section 3. Proposed System Implementation is illustrated in Section 4. In Section 5, experimental results are shown and conclusion and further extensions are included in Section 6. Finally, Section 7 describes references of this paper.

2. Related Works

Michalis Galanis et al., [6] in this paper, the hardware implementations of five representative stream ciphers are compared in terms of performance and consumed area in an FPGA device. The ciphers used for the comparison are the A5/1, W7, E0, RC4 and Helix. ABD RAHIM MAT SIDEK et al., [1] this paper presents results for the analysis of various stream cipher algorithms such as Linear Generator, Shrinking, Multiplexing and Variable-Memory Binary Generator (Memory Generator). All of these are based on the linear feedback shift registers with combining functions. For a common key length of 64 bits, the objective is to find the best algorithm using the statistical tests, linear complexity profile, correlation attack and guess and determine attack. In this paper the stream of A5/1 and Grain generators are evaluated using statistical tests to test whether the output key stream is random or not.

3. Linear Feedback Shift Register Based-Stream Ciphers

In modern cryptography, stream ciphers are most useful in applications where information needs to be processed at high speed and when less computational resources are available. A stream cipher can be viewed as a finite state machine with internal state and update function. Commonly, a stream cipher receives a key K and an initialization vector IV, and generates a long key stream.

There are many different pseudorandom sequence generators applied to stream ciphering. This paper only discusses about stream cipher based on the structure of LFSR. Linear feedback shift registers (LFSRs) are popular components in stream [9]. The LFSR provides a simple way to obtain sequences of vary high periods together with excellent statistics properties.

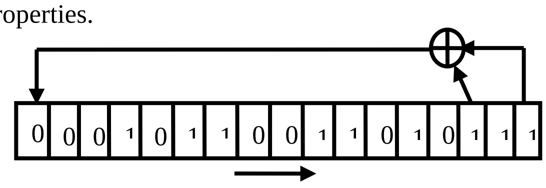
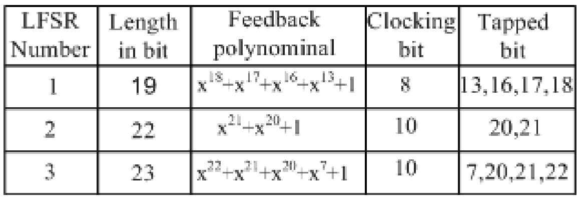


Figure 1: Linear Feedback Shift Register

3.1 A5/1 Stream Cipher

A5/1 is a generic synchronous stream cipher. Three linear feedback shift register (LFSRs) R1, R2 and R3 of lengths 19, 22 and 23 bits, respectively, are used. Figure 2 shows the primitive feedback polynomials of the registers.



LFSR Number	Length in bit	Feedback polynominal	Clocking bit	Tapped bit
1	19	$x^{18}+x^{17}+x^{16}+x^{13}+1$	8	13,16,17,18
2	22	$x^{21}+x^{20}+1$	10	20,21
3	23	$x^{22}+x^{21}+x^{20}+x^{7}+1$	10	7,20,21,22

Figure: 2 Feedback Polynomial of LFSRs in A5/1

A description of the A5/1 cipher is given in Figure 3. When a register is clocked, the bit at its tap positions are XORed together and stored in the least significant bit (lsb) of the register, after its bits have been shifted one position to the left. The most significant bit of each register is its output bit. The output of the A5/1 generator is at each clocking cycle obtained by XORing the output bits of three registers.

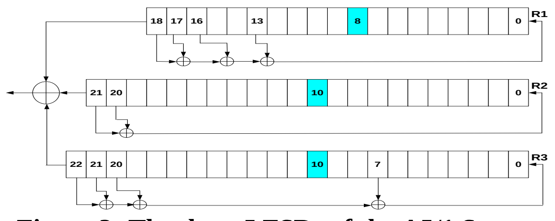


Figure 3: The three LFSRs of the A5/1 Stream Cipher

Initially the registers are set to zero. The algorithm takes as input two parameters: a 64-bit secret session key K_c and a 22-bit counter Fn. The contents of the three registers are re-initialized.

In the initialization phase, all registers are clocked regularly at each cycle. During the first 64 clock cycles, the 64 bits of the session key K_c are XORed into the lsb of each of the registers in parallel. During the next 22 cycles, the frame number F_n is mixed into the registers. The state of the three registers after key setup is called the initial state of frame n.

After initialization, an irregular clocking rule is introduced. The bits of the three register – bits 8, 10 and 10, are used to determine the stop/go clocking. At each step, the majority of the three bits is calculated and only those registers for which the clocking agree are clocked.

The registers are first clocked for 100 clock cycles without producing any output. Finally, the output bit is produced after each clock cycle [3, 4].

3.2 Grain Stream Cipher

The algorithm takes as input two parameters: an 80-bit secret session key and a 64-bit initialization vector.

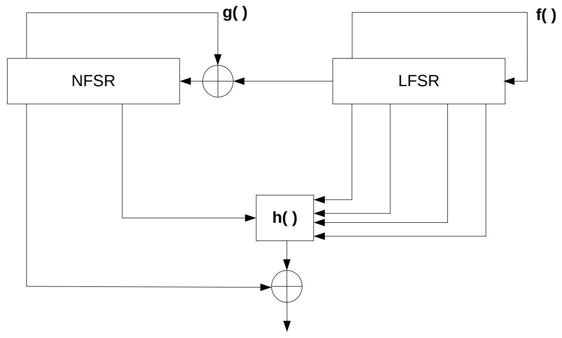


Figure 4: Key Generator of Grain Cipher

This cipher consists of three main building blocks-LFSR, NFSR and a filter function as shown in figure 4. The initialization of the key is done as follows.

First, load the NFSR with the key bits, and then load the first 64 bits of the LFSR with the IV. The remaining bits of the LFSR are filled with 1. Because of this the LFSR cannot be initialized to the all zero state.

The feedback polynomial of the LFSR, f(x) is defined as

$$f(x) = 1 + x^{18} + x^{29} + x^{42} + x^{57} + x^{67} + x^{80}.$$

The feedback polynomial of the NFSR, g(x), is defined as

$$g(x) = 1 + x^{17} + x^{20} + x^{28} + x^{35} + x^{43} + x^{47} + x^{52} + x^{59} + x^{65} + x^{71} + x^{80}+x^{17} x^{20} + x^{43} x^{47} + x^{65} x^{71} + x^{20} x^{28} x^{35} + x^{47} x^{52} x^{59} + x^{17} x^{35} x^{52} x^{71}+ x^{20} x^{28} x^{43} x^{47} + x^{17} x^{20} x^{59} x^{65} + x^{17} x^{20} x^{28} x^{35} x^{43} + x^{47} x^{52} x^{59} x^{65} x^{71} +x^{28} x^{35} x^{43} x^{47} x^{52} x^{59}.$$

From this state, 5-variables are taken as input to a function, h(x). The input is taken both from the LFSR and NFSR. The function is defined as

$h(x) = x_1 + x_4 + x_0 x_3 + x_2 x_3 + x_3 x_4 + x_0 x_1 x_2 + x_0 x_2 x_3 + x_0 x_2 x_4 + x_1 x_2 x_4 + x_2 x_3 x_4$

Then the cipher is clocked 160 times without producing any running key. Instead the output of the filter function, h(x), is fed back and Xored with the input, both to the LFSR and to the NFSR [7].

3.3 Tests for Stream Cipher

There are several tests that can be used to quantify the strength of stream ciphers. Standard tests that are independent of the algorithm are statistical tests, correlation attack and linear complexity profile. This paper restricts only statistical tests.

4. Proposed System Implementation

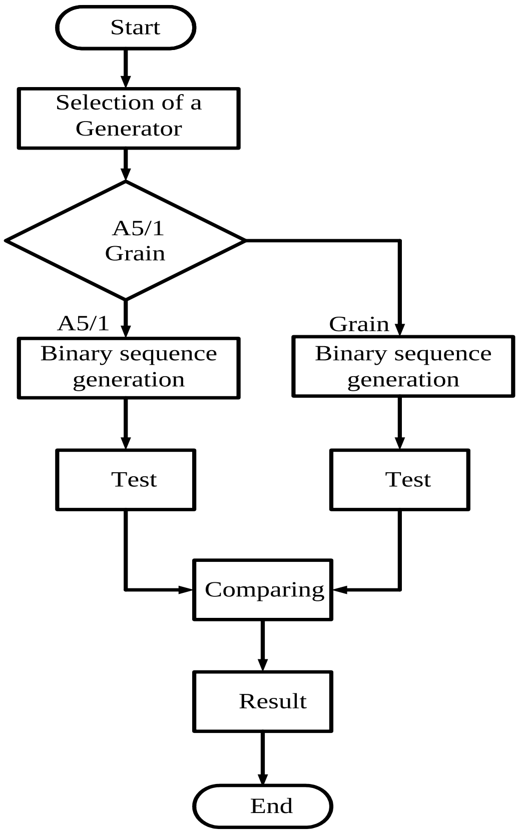


Figure 5: System Flow Chart

We propose A5/1 and Grain generators need to be used for evaluation. The generator produces a binary sequence of 0's and 1's of a given length n. For each generator, we initialize the key 64 bits for A5/1 and the key 80 bits for Grain. These generators generate a set of m binary sequences and save the sequences. Finally, the randomness of output sequences of these generators is tested by using statistical tests and compared the tests result shown in Figure 5.

4.1 Strategies for the Statistical Analysis

There are many distinct strategies employed in the statistical analysis of a random number generator. NIST has adopted the strategies outlined as follow:
Stage 1: Selection of a Generator.
Stage 2: Binary Sequence Generation
Stage 3: Execute the Statistical Test Suite
Stage 4: Examine the P-values
Stage 5: Assessment: Pass/Fail Assignment

In this paper, the generators of A5/1 and Grain stream ciphers are selected and used to test randomness of output sequence of these generators by using statistical tests.

4.2 Statistical Tests

Various statistical tests can be applied to a sequence to attempt to compare and evaluate the sequence to a truly random sequence. The National Institute of Standard and Technology (NIST) developed 16 statistical tests to test the randomness of (arbitrarily long) binary sequences produced by cryptographic random or pseudorandom number generators. Among 16 tests, this paper applies and uses three tests: Frequency (Monobit) Test, Runs Test, and Serial Test.

4.2.1 Frequency (Monobit) Test

The purpose of this test is to determine whether the number of ones and zeros in a sequence are approximately the same as would be expected for a truly random sequence.

This test calculates the number of ones and zeroes of the binary sequence and then calculates the probability value and checks the results. If there are no large differences between the results, this binary sequence is random with the following equations.

the test statistic $S_{obs} = \frac{|S_n|}{\sqrt{n}}$

P-value = erfc $\left(\frac{S_{obs}}{\sqrt{2}}\right)$

where n = the length of the bit string ;
erfc = complementary error function.

4.2.2 Runs Test

The purpose of the runs test is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. In particular, this test determines whether the oscillation between such zeros and ones is too fast or too slow with the following equations.

the test statistic $V_n(\text{obs}) = \sum_{k=1}^{n-1} r(k) + 1$

r(k)=0 if $\varepsilon_k = \varepsilon_{k+1}$, and r(k)=1

$$\text{P-value} = \text{erfc}\left(\frac{|V_n(\text{obs}) - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)}\right)$$

where n = the length of the bit string ;
r(k) = the number of runs of ones and zeros of various lengths ;
erfc = complementary error function.

4.2.3 Serial Test

The purpose of this test is to determine whether the number of occurrences of the m-bit overlapping patterns is approximately the same as would be expected for a random sequence. Random sequences have uniformity; that is, every m-bit pattern has the same chance of appearing as every other m-bit pattern with the following equations [2, 8].

$$\psi_m^2 = \frac{2^m}{n}\sum_{i_1 \ldots i_m}(v_{i_1 \cdots i_m} - \frac{n}{2^m})^2 = \frac{2^m}{n}\sum v^2_{i_1 \ldots i} - n$$

the test statistic $\nabla \psi_m^2 = \Psi_m^2 - \psi_{m-1}^2$

the test statistic $\nabla^2 \psi_m^2 = \psi_m^2 - 2\psi^2_{m-1} + \psi^2_{m-2}$

$$P-value1 = igamc(2^{m-2}, \nabla \psi_m^2 / 2)$$

$$P-value2 = igamc(2^{m-3}, \nabla^2 \psi_m^2 / 2)$$

where n = the length of the bit string ;
m = the length in bits of each block ;
igamc = Incomplete Gamma Function

5. Experimental Results

With the standard key length which is 64 bits for A5/1 and 80 bits for Grain, and 1000 bits sequence, the strength comparison of stream ciphers was made. For statistical test, test conditions which are 1000 sequences chosen randomly and all the results are shown in Table 1.

For a particular statistical test, compute the proportion of sequences that pass. For example, if 1000 binary sequences were tested (i.e., m = 1000), α = 0.01 (the significance level), and 996 binary sequences had P-values $\geq$ 0.01, then the proportion is 996/1000 = 0.9960. The range of acceptable proportions is determined using the confidence interval defined as, $\hat{p} \pm \sqrt{\frac{\hat{p}(1-\hat{p})}{m}}$, where $\hat{p}$ = 1 –α; m is sample size. If the proportion falls outside of this interval, then there is evidence that the data is nonrandom. The confidence interval is

$$0.99 \pm 3\sqrt{\frac{0.99(0.01)}{1000}} = 0.99 \pm 0.0094392$$

(i.e., the proportion should lie above 0.9805607).

The outputs (binary sequence) or various test conditions of two generators are tested using statistical tests: Frequency test, Runs test and Serial Test as shown in table1. For example, when 1000 output sequences test conditions are generated randomly by A5/1 generators and they are tested by using statistical tests, the 1000, 530 and 551 output sequences are truly random sequences by testing Frequency Test, Runs Test and Serial Test respectively.

After the analyses, it is found that the bit sequence of A5/1 generator is more random than Grain.

Table 1: Comparison results for Statistical tests of A5/1 and Grain Generators

Generator	Tests	Proportion	
A5/1	Frequency (Monobit) Test	1000/1000	1.0000
A5/1	Runs Test	530/1000	0.5300
A5/1	Serial Test	551/1000	0.5510
Grain	Frequency (Monobit) Test	999/1000	0.9990
Grain	Runs Test	586/1000	0.5860
Grain	Serial Test	123/1000	0.1230

6. Conclusion and Further Extension

Random numbers play an important role in the use of encryption for various security applications. The stream cipher is a type of cipher system where the process of enciphering and deciphering is performed one bit at a time. The comparison of A5/1 and Grain generator are performed.

We found that the bit sequence of A5/1 generator is more random than Grain. With the present performance of computer technology in terms of speed and memory space, A5/1 generator can be implemented easily on application such as messaging system, e-mail or mobile communication system. Due to its characteristic, this generator will guarantee the digital information exchange in mobile communication is confidential.

This is implemented as a tool. So, it is fully extensible. Some random number generators and other NIST tests can be added to this system.

7. References

[1] ABD RAHIM MAT SIDEK1 & AHMAD ZURI SHA'AMERI2, "Comparison Analysis of Stream Cipher Algorithm For Digital Communication, Jurnal Teknologi,46(D) Jun 2007:1-16, Universiti Teknologi Malaysia.

[2] Orr Dunkelman , "Cryptanalysis'of the A5/1 GS] Andrew Rukhin et al., "Statistical Test Suite For Random And Pseudorandom Number Generators For Cryptographic Applicatios", NIST Special Publication 800-22, 2010.

[3] Eli Biham, M Stream Cipher".

[4] Imran Erguler [1,2] , and Emin Anarim[2] , "A Modified Stream Generator For The GSM Encryption Algorithms A5/1 And A5/2".

[5] Jan Krhovjak, "Analysis, demands, and properties of pseudorandom number generators".

[6] Michalis Galanis et al., "Comparison of the Hardware Implementation of Stream Ciphers," The International Arab Journal of Information Technology, Vol.2, No,4, October 2005.

[7] Martin Hell[1], Thomas Johansson[1] and Willi Meier[2], "Grain-A Stream Cipher for Constrained Environments".

[8] U. Maurer, "A Universal Statistical Test for Random Bit Generators," Journal of Cryptology. Vol. 5, No. 2, 1992, pp. 89-105.

[9] http://en-wikipedia.org/wiki/Stream Cipher