IJCSN - International Journal of Computer Science and Network, Volume 7, Issue 2, April 2018
ISSN (Online): 2277-5420
www.IJCSN.org
Impact Factor: 1.5

Testing and Generation of Synchronous Stream Ciphers

[1]Nusrat Mohi Ud Din

[1]M.Tech IT, Central University Of Kashmir
Srinagar, India

Abstract - *Random numbers generators are used in many applications e.g. statistical sampling, cryptography, computer simulation. In this paper our focus will be on random number generators used in cryptography. This paper presents and discusses the analysis methods applied in symmetric cryptography, especially on stream ciphers. In modern digital cryptography Random numbers play a very crucial role. Due to unpredictable, unknown, un-guessable and irreproducible properties of Random numbers, they play a significant role in cryptography in securing the secret information. Many cryptographic systems are compromised due to the lack of thorough analysis of Random number generator and of the quality of sequences produced.*

Keywords - *LFSR, NLFSR, Randomness, Statistical Tests, Pseudorandom numbers*

1. Introduction

Internet of things has revolutionized the digital world by enabling the communication between billions of devices connected over the internet but at the same time has exposed the users to variety of security threats and vulnerabilities. Cryptography plays an important role in security and privacy and one of the important primitives of cryptography are the stream ciphers. Stream ciphers can be classified as being either synchronous or self-synchronous based on the internal state. If the change in the state is independent of plaintext or cipher text message, then it is synchronous stream cipher. Otherwise if the internal state is updated based on the previous cipher text, then it is self-synchronous. Synchronous stream ciphers are in used in applications where low delay and high speed are required. For example A5 stream cipher family is used in GSM standard and E0 cipher in Bluetooth applications to provide security. But unlike block ciphers, we do not have a standard in stream ciphers.

The security of the cryptographic system is strongly related to the random number generators. If an inadequate random number generator is utilized the whole cryptographic system gets compromised. As a result cost and efforts to recover from the security break would be very high. The key properties of random number are unpredictability, independency (lack of correlation), uniform distribution (lack of bias). Random numbers also have certain statistical properties and can be measured using statistical tests [6]. There is a wide range of statistical test suites NIST, TestU01, Diehard, ENT, Crypt-X, FIPS140- 1, FIPS140-2 etc. Although by applying these Statistical tests thoroughly one can get high confidence in the generator but he cannot be sure.

2. Different Design Approach To Stream Ciphers

2.1 Linear Feedback Shift Register Based Stream Ciphers

Linear feedback shift registers (LFSR) are used in stream ciphers for key stream generation. They owe good statistical properties like low implementation cost, large period. They have simple structure and can be easily analyzed mathematically. The desirable properties of LFSR based key stream are linear complexity, large period and good statistical properties. These properties of LFSR are necessary but are not sufficient conditions for stream ciphers to be considered cryptographically secure. However linear feedback shift register do not guarantee security, they are prone to various attacks like known plaintext attack, Algebraic attack, fast correlation attack, cache timing attack. The three general methods to increase the security in LFSRs are filtering function, clock controlling and non-linear

combing function.

2.2 Non-Linear Feedback Based Shift Register Based Stream Ciphers

An alternative to LFSRs is the Non-Linear Feedback based Shift Register (NLFSRs) for producing pseudorandom sequences for stream ciphers. NLFSR based stream ciphers are more resistant to cryptanalytic attacks than LFSRs. LFSR are inherently linear. The general methods to introduce non- linearity in LFSR are Non-linear combination generator, Non- linear filter generator, Clock-controlled generator.

2.2.1 Non-Linear Combination Generator

The non-linear Boolean function F called the combining function is used for manipulating the output of several parallel LFSR to produce the key stream. The Boolean function F must have high algebraic degree, high non-linearity and high correlation immunity.

2.2.2 Non-Linear Filter Generator

The non-linear Boolean function F called the combining function is used for manipulating the output of several parallel LFSR to produce the key stream. The Boolean function F must have high algebraic degree, high non-linearity and high correlation immunity.

2.2.3 Clock Controlled Generator

In this generator two registers are used, generator register (GR) and control register(CR).The register facilitating clocking control is called control register (CR). The register generating the key stream according to the output sequence of CR is called GR. The GR is clocked in an irregular manner. Generators based on this principle are like stop-and-go, alternating step generator and the shrinking generator.

3. Generation of Random Sequences

RNGs can be categorized into three types, True random number generators (TRNGs), pseudo random number generators (PRNGs) and unpredictable random number generators (URNGs).

3.1 True Random Number Generator (TRNGs)

In TRNGs, randomness is extracted by sampling and digitization of natural phenomenon (e.g. thermal noise, radiation, jitter etc.). TRNGs are unpredictable, irreproducible and have high level of non-determinism. They do not have perfect uniform distribution and independence. They require special hardware and some of them are slow and impractical. The characteristics of TRNGs are different from PRNGs. Firstly TRNGs are inefficient than PRNGs, also TRNGs haven no period.

3.2 Pseudo Random Number Generator (PRNGs)

In PRNGs randomness is extracted from the initial value called seed, which then prolonged by means of a formula (usually recursive). PRNG is a deterministic algorithm that produces the sequences which are not truly random. PRNGs must satisfy following conditions:

- The statistical properties of TRNGs sequences must be close to true random numbers.
- The initial value (seed) must be large enough to be secure against the exhaustive search key that uses pseudorandom generator.
- The statistical properties of pseudo random number generator must pass the statistical tests of random numbers

3.3 Unpredictable Random Number Generators (URNGs)

In UPRNGs randomness is extracted from the easily available devices e.g. computer components. They reveal the behavior of both TRNGs and PRNGs.

4. Statistical Tests

The quality of stream ciphers can be measured by performing the various statistical tests. There is a wide range of statistical tests such as NIST Test Suite, ENT, Diehard, FIPS140-1, FIPS140-2, CRYPT-X, TestU01 etc.

4.1 FIPS 140–1/FIPS140–2Test

FIPS statistical tests contain the Monobit Test, the Poker Test, the Runs Test and the Long Run Test.

4.2 DIEHARD Statistical Tests

This set of tests was aimed to identify weaknesses in

many common non–cryptographic pseudorandom number genera- tor (PRNG) algorithms. The output of the generator of 11 megabytes or more is analyzed by these tests. The DIEHARD Statistical tests include: birthday spacing test, overlapping 5–permutation test, binary rank test 31×31, binary rank test 32×32, binary rank test 6×8, bitstream test, Opso, Oqso and DNA tests, count–the–1's test on a stream of bytes, parking lot test, minimum distance test, 3Dspheres test. Most of the DIEHARD tests return a p-value, which should be uniform on [0, 1] if the input file contains truly independent random bits.

4.3 ENT Test Suit

Ent contains five tests. Ent has two operation modes: binary and byte, with the latter one enabled by default. The five tests are as follows:

Entropy-This test computes the entropy of the sequence that is to be examined. A random sequence should have high entropy.

Chi-square tests- The chi-square test computes the frequency of the symbols, and compares it with the frequency expected in a uniform distribution.

Arithmetic mean- this is an arithmetic mean of the symbols in the sequence. The expected statistic value for a true random sequence is 0:5 in binary mode and 127:5 in byte mode.

Monte Carlo Value for Pi

Serial Correlation-This test computes the correlation between two consecutive symbols (bits or bytes) in the sequence. A good random sequence would have low correlation, very close to zero; while a bad random sequence would lead to higher values.

4.4 NIST Test Suite

NIST Test is one of the most popular tools for pseudo random sequence analysis. The NIST Test suite is a Statistical package comprising of 15 tests that were settled to test randomness of binary sequences formed by either hardware or software oriented cryptographic random or pseudorandom number generators. The 15 statistical tests are-The Frequency (Monobit) Test, Frequency Test within a Block, The Runs Test, Tests for the Longest-Run-of-Ones in a Block, The Binary Matrix Rank Test, The Discrete Fourier Transform(Spectral)Test, The Non- overlapping Template Matching Test, The Overlapping Template Matching Test, Maurer's "Universal Statistical" Test, The Linear Complexity Test, The Serial Test, The Approximate Entropy Test, The Cumulative Sums (Cusums) Test, The Random Excursions Test and The Random Excursions Variant Test [10].

5. Literature Survey

5.1 The conventional pseudorandom numbers generators (Linear Congruential Generator, Mersenne Twister method, Marsaglia Ziggurat algorithm) and chaos-based pseudorandom number generators (logistic, chebyshev, saw tooth-like) had been analysed under FIPS140-2 and SP800-22. The tests that had been performed are Monobit test, poker test, long run test, run test. The 20,000 bits of output sequences are provided as input to these tests.
The PRNG based on chaotic method chebyshev did not satisfy three tests except long run test. The results of long run show that there are no long runs [8].

5.2 The specification of Dragon is that it has two versions Dragon-128 and Dragon-256. For Dragon-128 key and initialization vector (IV) are 128-bits and Dragon-256 has key and IV 256-bits.Dragon has NLFSR, an update function F and a memory unit M. The update function manipulates the internal state to generate a pseudo-random number keystream. The throughput of Dragon is in gigabits per second and requires very little more than 4 kilobytes of memory, thus making it suitable for constrained environment [9]. Keystream generated by Dragon had been analysed under the statistical tests of CRYPTX [7]. The 30 bit keystream produced by the Dragon, each having size of 8MB had been analysed under Frequency, Subblock, Change point, Binary Derivative and Run tests. The sequence and linear complexity tests were also executed on 30 keystream each of size 200 kb. Dragon passed all the tests.

5.3 Two Pseudo random sequences generated by two 32-bit LFSRs are used as Test sequences in the NIST Statistical Test suite. Two Feedback primitive polynomials that define the LFSRs are

$$L(x) = x^{32} + x^{31} + x^{29} + x^{1} + 1 \quad (1)$$

$$L(x) = x^{32} + x^{19} + x^{18} + x^{13} + 1 \quad (2)$$

Sequences that had been generated were of size 110MB. It was found that the sequence generated by LFSRs of size 110MB, the NIST test suite was not able to achieve DFT test. And all other tests executed correctly. The pseudo random sequences generated by LFSR that is

being described by polynomial (1) did not pass the Rank test and Overlapping Template matchings. The pseudo random sequences generated by LFSR described by polynomial (2) did not pass the Matrix Rank Test [2].

5.4 The Grain Stream Cipher is a synchronous stream cipher consisting of NLFSR, LFSR and a Boolean function [11] and is suitable for environments with limited resources like gate, power consumption and chip area. Grain-128 uses 128-bit secret key and 96-bit initialization vector (IV). The algorithm had been analysed in the NIST statistical Test Suite and results obtained prove that the algorithm is not random at the 1% significance level [12].

6. Conclusion

Although there are various stream ciphers used in cryptography, because of their speed. But unlike block ciphers there is no standard for stream ciphers. The security of the cryptographic system depends on the keystream generator. The flaws in keystream generator can led to the compromise of the whole cryptographic system. Cryptography requires the high quality and high performance random number generator. Various statistical are available that can be applied on the pseudorandom number generators to increase the confidence in the generator before being used for any cryptographic application. In future Statistical Test suites can be performed on NLFSR based stream ciphers to ensure the security of these Stream Ciphers and to measure the quality of random sequences produced.

References

[1] M. U. Bokhari, Faheem Massodi, "Comparative Analysis of Structures and Attacks on various Stream Ciphers", *proceedings of the 4th National conference, Computing for National Development*, 2010.

[2] Rafal Stepien, Janusz Walczak, "Statistical Analysis of the LFSR generators in the NIST STS Test Suite", *Computer Applications in Electrical Engineering*, 2011.

[3] Robertas Smaliukas, "Block Cipher And Non-Linear Shift Register Based Random Number Generator Quality Analysis", *Technical Report*", October 2015.

[4] Julio Hernandez-Castro, David F. Barrer "Evolutionary Generation and degeneration of Randomness to assess the Independence of the ENT Test Battery*", Evolutionary Computation, IEEE*, 2017.

[5] Faheem Massodi,Shadab Alam ,M U Bokhari ,"An analysis of Linear Feedback Shift registers in Stream ciphers", *International Journal of Computer Applications*,Vol.46, May 2012.

[6] Kinga Marton, Alin SUCI , Christian SACAREA , Octavian CREŢ, **"**Generation And Testing Of Random Numbers For Cryptographic Applications", *Proceedings Of The Romanian Academy, Series A,*vol.13,pp. 368–377, 2012.

[7] E. Dawson, A. Clark, G. Gustafson, and L. May, CRYPT-X'98 User Manual, 1999.

[8] Rashidah Kadir, Mohd Aizaini Maarof, "A Comparative Statistical Analysis of Pseudorandom Bit Sequences", *Fifth International Conference on Information Assurance and Security*, 2009.

[9] Chen, Kevin, et al. "Dragon: A Fast Word Based Stream Cipher", *International conference on Information Security and Cryptology,* Springer, Berlin, 2004.

[10] A. RUKHIN, et al., "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", *NIST Special Publication 800–22, National Institute of Standards and Technology*, revised May 2001.

[11] Hell, Martin, Thomas Johansson, and Willi Meier, "Grain: a Stream Cipher for Constrained Environments", *International Journal of Wireless and Mobile Computing,* 2007.

[12] Zawawi, Seman, Mohd Zaizi," Randomness Analysis of Grain-128", *AIP Conference Proceedings,* 2013.