Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Overview of Randomness Test on Cryptographic Algorithms

To cite this article: Zhang Mengdi *et al* 2021 *J. Phys.: Conf. Ser.* **1861** 012009

Overview of Randomness Test on Cryptographic Algorithms

Zhang Mengdi*, Zhang Xiaojuan, Zhu Yayun and Miao Siwei

China Electric Power Research Institute, Beijing, 100192, China

*Corresponding author's e-mail: zhangmengdi@epri.sgcc.com.cn

Abstract. Randomness is an important research topic in the field of information security, especially in cryptography. Randomness test techniques are used to examine the quality of random numbers so that they meet the requirements of the application. The randomness of cryptographic algorithms is one of the key concerns in the algorithm design. People have put forward different standards and test requirements for the randomness of cryptographic algorithms, as well as developed corresponding randomness test kits. This paper analyzes the randomness test technologies of cryptographic algorithms and the general randomness test methods, and compares them on this basis. At the end, the actual application scenarios to apply these randomness test methods are discussed.

1. Introduction

Randomness is a form of contingency, which is uncertainty of each event in the event set with a certain probability. Combined with the characteristics of statistics and cryptanalysis, the random sequence should satisfy three conditions: 1. The sequence follows a uniform distribution. 2. Each element of the sequence is independent of each other. 3. The rest of the sequence can not be predicted from any sequence. According to the ways of generating random numbers, random numbers can be divided into two categories: true random numbers and pseudo-random numbers. True random number generators (RNGs) are composed of two parts: entropy source and algorithm post-processing. The common entropy sources include thermal noise amplification, oscillator sampling, chaotic circuit, light source noise, chaotic lasers and quantum noise. Pseudo random number generators (PRNGs) take a seed as input and generate an output sequence by function. Pseudo random number generator is widely used because of its convenient and fast characteristics.

At present, the algorithm randomness test is mainly realized by examining the randomness of the output sequence. The randomness test of the output sequence is through the sample output, which is statistically tested to detect whether it has the characteristics of the real random number sequence. The statistical test method used is hypothesis test, which assumes that the sequence to be tested is a real random sequence, and the hypothesis is H_0, corresponding to another hypothesis H_1, that is, the sequence to be tested is not a random sequence. In the test, selected statistics are calculated for the sample sequence, and then compared with threshold value. Because probability distribution is related to threshold value, if statistical value exceeds threshold value, this kind of small probability event should not occur from the perspective of hypothesis testing. If statistical value exceeds threshold value, H_0 is rejected, otherwise H_0 is accepted.

In this article, Chapter 2 gives the background of randomness test methods and standards and state quo of randomness tests. The general test methods on different cryptographic algorithms are also described. Chapter 3 introduces the common randomness test standards, methods and toolkits, while Chapter 4 describes various application scenarios of randomness test.

2. Research Background

In 1949, Shannon published "Information Theory of Secret Systems", which established the theoretical basis for private key cryptography [1], and cryptography has become a science since then. The publication of "New Directions in Cryptography" in 1976 [2] and the promulgation and implementation of the American data encryption standard DES [3] in 1977 marked the birth of modern cryptography. The Randomness test is a testing method based on hypothesis testing. Hypothesis testing [4] is an important type of statistical inference problem, which is based on the principle of small probability. In probability theory, an "impossible event" refers to an event whose probability is less than a specific and small enough number [5]. Specifically, the randomness test is to determine whether a certain pattern exists in the output sequence. Because the probability of this particular pattern appearing multiple times in a longer sequence is very small, it can be considered that the sequence has poor randomness.

In order to evaluate whether pseudo-random numbers are independent and unpredictable, U.S. computer scientist D. Knuth proposed 11 randomness test methods in 1968 such as frequency test, run-length test, poker test, etc., and became the pioneer of systematic randomness testing [6]. The Knuth test suite is one of the statistical randomness suites, but the suite is mainly used for real number sequences, and the test parameters are not explicitly given. American mathematician G. Marsaglia is another originator of the field of random number research. In 1968, he proposed the use of linear congruential method to construct a pseudo-random number generator (Linear Congruential Generator, LCG) [7], and released it as a CD-ROM Diehard Tests in 1995 [8], a random test method set containing 12 methods. Menezes mentioned the 5 basic tests in the book "Handbook of Applied Cryptography" in 1997 [14].

The German Federal Office for Information Security (Bundesamtfiir Sicherheit in der Informationstechnik, BSI) issued the PRNG evaluation standard AIS 20 [9] in 1999 and released it in 2001. The evaluation standard AIS 31 for RNG [10]. In 2011, dozens of standards such as AIS 20 and AIS 31 were integrated into a RNG-specific evaluation program AIS 20/AIS 30 [11], which contains 9 random test methods. In 2001, the National Institute of Standards and Technology (NIST) issued the SP 800-22, a standard for randomness testing, with a total of 16 test methods and the corresponding statistical test suite Statistical Test Suite[8]. As the relevant standards and tools for randomness testing in the world have matured, the State Cryptography Administration (SCA) of China began drafting the "Random Testing Specification" in 2009, with a total of 15 test methods. In 2012, it officially became the national secret standard GM/T 0005-2012[12], and was upgraded to the national standard GB/T 32915-2016[13] in 2016.

In 2010, Sulak F, Doğanaksoy A et al. raised an alternative method for analyzing short sequences without adjustment testing [16]. Turan MS, DoĞanaksoy A et al. [17] in 2008 focused on the independence of randomness test and its impact on test suite coverage. They observed through experiments that the frequency of short sequence, overlapping template, longest template run, random walk height and maximum order complexity test are related. It also proposed the concept of sensitivity and analyzed the influence of simple transformation on the output p value. Regarding the test results of the test package, the independence and dependence of the random tests and their coverage are critical to the test results. In 2018, Onur KOÇAK, Fatih et al. [18] gave the Knuth suite test parameters to make the test suitable for integer sequences and binary sequences, and made suggestions for the selection of these parameters, so as to apply the kit to some widely used encrypted random number sources.

In the perspectives of cryptographic algorithms, Chen Hua summarized the randomness test methods of various cryptographic algorithms in [21], including the randomness test of block cipher algorithm, stream cipher algorithm and pseudo-random number generation algorithms, with the research on the randomness test on hash function added, which includes the randomness test of the digest, message diffusion and key diffusion. He also designed a randomness test method about the key schedule algorithm of block cipher, by which we can calculate the degree of statistical independence between sub keys. For block cipher algorithms, there are some test items based on binomial distributions, including grouping frequency test, followability test, cipher text independence test, clear text avalanche test and key avalanche test. The test on key schedule algorithm is to expand and combine X keys into Y sequences, then perform the local randomness test, so that the degree of statistical independence between all the sub keys is decided. For stream cipher, the output should be indistinguishable with the true random number, and the seed key should be evenly spread in the key stream, which is through transformation and local randomness test.

3. Randomness Test Standards, Methods and Tools

With development of the randomness test, test standards have been formulated in different countries. At present, SP 800-22 of NIST, GB/T 32915-2016 of SCA and AIS 20 / AIS 31 of BSI are the most well-known. To measure the test results, SP800-22 and GB/T 32915-2016 use the p-value approach, which is to compare the p-value and significance level α, and decide whether it indicates failure. However, AIS 20/AIS 31 applies the threshold approach, by comparing the observed value with the threshold corresponding to the significance level.

NIST SP800-22 adopted 2 approaches including (1) the examination of the proportion of sequences that pass a statistical test and (2) the distribution of p-values to check for uniformity. If either of these approaches fails, additional experiments should be conducted to see if the failure was a statistical anomaly or clear evidence of non-randomness.

GB/T 32915-2016 is used to test the binary sequences generated by commercial RNGs. If it passes all the 15 test items, then we can say it conforms with this standard.

AIS 20/ AIS 31 is applied to test the RNGs and contains 9 test methods of T0-T8 with 2 test procedures of A and B. Process A includes T0-T5, and process B includes T6-T8. DRNGs, PTRNGs and NPTRNGs need to take the test procedure A. The goal of test procedure A is to check whether the random numbers are statistically inconspicuously. The Binary valued das-random numbers of PRTNGs need to select test procedure B, which is to ensure that the entropy per das-bit is sufficiently large.

The following table summarizes and compares the three standard test methods, with a total of 24 test methods [20].

Table 1. Randomness Test Methods Comparison in 3 standards

	Randomness Test Methods	**NIST**	**SAC**	**BSI**
1	The Frequency (Monobit) Test	√	√	√
2	Frequency Test within a Block	√	√	×
3	The Runs Test	√	√	√
4	Tests for the Longest-Run-of-Ones in a Block	√	√	×
5	The Binary Matrix Rank Test	√	√	×
6	The Discrete Fourier Transform (Spectral) Test	√	√	×
7	The Non-overlapping Template Matching Test	√	×	×
8	The Overlapping Template Matching Test	√	×	×
9	Maurer's "Universal Statistical" Test	√	√	×
10	The Linear Complexity Test	√	√	×
11	The Serial Test	√	√	×
12	The Approximate Entropy Test	√	√	×
13	The Cumulative Sums (Cusums) Test	√	√	×
14	The Random Excursions Test	√	×	×
15	The Random Excursions Variant Test	√	×	×
16	The Poker Test	×	√	√
17	The Runs Distribution Test	×	√	×

18	The Binary Derivation Test	×	√	×
19	The Autocorrelation Test	×	√	√
20	The Disjointness Test	×	×	√
21	The Long Run Test	×	×	√
22	The Uniform Distribution Test	×	×	√
23	The Comparative Test for Multinomial Distributions	×	×	√
24	The Entropy Estimation Test	×	×	√

Some of the classical methods are explained as follows.

1. Frequency test

The frequency test mainly depends on the proportion of "0" and "1" in the whole sequence. This test is the basis of the randomness test, which should be carried out first, and other tests should be carried out after the frequency test is passed. If the frequency test can not pass, then the sequence is not random without other tests. In a real random sequence, when the length tends to infinity, the number of "0" and "1" should be roughly equal, that is, half of each.

2. Run test

Run length is a subsequence composed of continuous "0" or "1" in a sequence. The purpose of run detection is to determine whether the number of "1" runs of different lengths and the number of "0" runs are consistent with the expected value of the ideal random sequence. Specifically, this test means to determine whether the oscillation between such "0" and "1" sub blocks is too fast or too slow.

3.Rank test of binary matrix

Binary matrix rank test is to divide binary sequence into several equal lengths and non overlapping matrices, and then count the rank distribution of all matrices to detect whether the linear independence of each matrix meets the requirements of random sequence, so as to judge whether the sequence has randomness.

4. Linear complexity test

The Linear complexity test is to divide binary sequence into several m-bit non overlapping bit blocks, and count the shortest length of linear feedback register (LFSR) of each bit block to detect whether the linear complexity distribution of all bit blocks meets the requirements of random sequence, so as to judge whether the sequence has randomness.

5. Discrete Fourier test

This test is mainly to see the peak height of the sequence after split step Fourier transform. The purpose is to detect the periodicity of the signal to be tested, so as to reveal the degree of deviation between it and the corresponding random signal. The method is to observe whether the number of peaks above the 95% threshold is significantly different from that below 5%.

6. Overlapping template matching test

Overlapped subsequence test is to count the occurrence times of all m-bit overlapped subsequence patterns in binary sequence to detect whether 2^m patterns appear with equal probability, so as to judge whether the sequence is random.

7. Accumulation sum test

Accumulation sum test is to construct overlapping increasing subsequences after the binary sequence is standardized, and calculate the absolute value of random walk of each subsequence to detect whether the maximum value conforms to the expected value of the random sequence, so as to judge whether the sequence has randomness.

These tests focus on different characteristics of non randomness which may exist in sequences. A test suite is formed by selecting from test set to test necessary randomness properties as much as possible, where the eliminating ability and mutual information entropy are concerned. The [19] gives the eliminating orders of the 15 test methods of NIST SP800-22. The mutual information entropy is denoted by $I(X,Y)=H(X)+H(Y)-H(X,Y)$, meaning the actual amount of information that two tests can provide. In 2020, Karell Albo Jorge Augusto et al. [22] proposed a method of detecting statistical dependence by using mutual information. The main advantage of using mutual information is that it has the ability to detect non-linear correlations, while the linear correlation coefficients used in previous work cannot be detected. Dependency detection between statistical randomness tests allows one to discriminate statistical randomness tests that measure similar characteristics, and thus minimize the amount of statistical randomness tests that need to be used.

Randomness test toolkit is realized by integrating test methods into a program. Some of the well-know toolkits include the officially released NIST STS, ENT, Diehard, TestU01 and etc. NIST STS is published in NIST SP800-22, which is a personal tool for statistical testing of PRNGs. ENT pseudo-random number sequence test program takes the byte stream of the file to be tested as the input, including five test items: entropy, chi square test, arithmetic mean, Monte Carlo value method to calculate pi and sequence correlation coefficient. The program can be used to evaluate pseudo-random number generators for encryption and statistical sampling applications, compression algorithms, and other applications where file information density is of concern. The Diehard test is a set of statistical tests used to test the quality of random number generator, which contains a total of 15 kinds of tests, such as Birthday spacings, Monkey tests, The squeeze test and so on. TestU01 software library implements several types of random number generators, classical statistical tests of random number generators, some other tests and some original tests proposed in the literature. These tests can be applied to predefined generators in the library, user-defined generators, and random number streams stored in files. TestU01 provides multiple sets of tests, including smallcrush (10 tests), crush (96 tests) and bigcrush (160 tests). These test toolkits can be improved and optimized to adapt to different platforms or embeded systems.

4. Application Scenario Analysis of Randomness Test

Randomness test is mainly used for checking the quality of RNG and PRNG. In 2017, G.S. Karimovich et al. proposed RNG and PRNG based on computer sources, and used NIST STS tools to test their randomness respectively [23]. The results show that the 128 or 256 bit key generated by RNG has at least 88% probability of passing all the randomness tests; the 106 bit sequence generated by PRNG has nearly 100% probability of passing the rest tests except for random offset test and random offset variable test. Therefore, RNG and PRNG have good randomness and reliability.

In the test on cryptographic algorithms, we use the randomness test to decide whether the algorithm is secure. In 2008, J. Nakahara proposed the 3D-AES crypto system, which extends the block length and key length to 512 bits and the number of iterations to 22 rounds. The purpose is to encrypt large capacity data safely and efficiently [24]. However, in 2017, S. Ariffin and N.A.M.Yusof tested the randomness of different data sets generated by the 3D-AES algorithm through two rounds of AES evaluation [25]. At the significance level of 0.001, the results showed that the key avalanche test failed, so it was determined that the 3D-AES crypto system did not have randomness.

In the progress of engineering applicability, it is mainly to apply randomness test to various cryptographic components or structures, such as random number generators, symmetric encryption systems, hash functions and so on. According to the test results, we can determine whether the test object has randomness property. The application scenarios of random number include: verification code generation, lottery activity, UUID generation, session id generation, token generation and CSRF Token, password recovery token, game (generation of random elements), shuffle, sequence with specific shape of Tetris, game explosive equipment and password application scenario, and generating keys (symmetric password, message authentication), generating key pairs (public key, digital signature), generating IV (CBC, CFB and OFB mode for block cipher), generating nonce (for defending replay attack, CTR mode of block cipher), generating salt: PBE for password based cipher, etc.

There are many security vulnerabilities of random numbers, and many random number vulnerabilities are involved in various programming languages, such as CVE2013-6386. The random numbers generated by the rand function are predictable pseudo-random numbers, which can be used by remote attackers to predict the security string and bypass the established access restrictions. In this case, random numbers generated by the application needs to be examined. Due to the lack of basic cryptography knowledge, developers do not know to use random numbers or use non-compliant random number generators, or imperfect interface documents in some application scenarios and frameworks, or developers do not read them carefully, which leads to the existence of random number security vulnerabilities. For example, a pseudo-random number is needed as a token for password recovery, but many businesses generate the token directly according to the user name, as well as in OAuth 2.0 protocol where a third party needs to pass a state parameter as a CSRF token to prevent CSRF attacks. Many developers do not use this parameter, or pass in a fixed or non-random value instead of a real random value. Because the authenticator can't verify the validity of this value at the business level, it leads to the CSRF attack of OAuth. In this case, we need to use the randomness test methods to check the random number generated by this kind of application, so as to prevent the vulnerability of pseudo-random numbers, which leads to system security problems.

5. Summary

Randomness test on cryptographic algorithms is to guarantee the correct operation of secure mechanisms in information security. The misuse of random numbers results in insecure RNGs, weak cryptographic algorithms and vulnerabilities in applications. In this paper, the standards and test methods of randomness test algorithms are analyzed and summarized. The analysis shows that at present, the randomness detection mainly focuses on the randomness evaluation of the algorithm output sequence, and the algorithm itself has an impact on the random number of the output sequence. At present, there is no relevant technology, standards or toolkit to analyze it. Therefore, the next step of this paper is to analyze the impact of the encryption algorithm's own implementation on the random number of the cipher algorithm's output sequence, and design related analysis and evaluation standards and tools for experiments.

Acknowledgments

This work was supported by the Innovation Fund Program of CEPRI (Research on environment planning and key technologies of cryptographic simulation and verification, XX-83-20-002).

References

[1]C.E.Shannon.“Communication Theory of Secrecy Systems" Bell System Technical Journal,1949(vol.28):656-715.

[2]W.Diffie,M.E.Hellman.“New Directions in Cryptograhy” IEEE Transactions on Information Theory,1976(vol.IT-22),No.6:644-654.

[3]NBS."Data Encryption Standard."FIPS PUB 46,National Bureau of Standards,Washington,D.C.(Jan.1977)

[4] Sheng Chu, Xie Shiqian, Pan Chengyi. Probability theory and mathematical statistics. Higher Education Press, 1999.

[5]Juan Soto.Statistical Testing of Random Number Generators[DB/OL].http://www.nist.gov.

[6]D.Knuth.The Art of Computer Programming Volume 2:Seminumerical Algorithms lst edition[M].Addison-Wesley,1969.

[7]G.Marsaglia.Random numbers fall mainly in the planes[Jl.Proceedings of the National Academy of Sciences of the United States of America,1968,61(1):25.

[8]Wikipedia.Diehard tests[EB/OL].https://en.wikipedia.org/wiki/Diehard_tests,2018.

A.Rukhin,J.Soto,J.Nechvatal,et al.A statistical test suite forrandom and pseudorandom number generators for cryptographic applications[S].NIST Special Publication 800-22,2001.

[9]W.Schindler.AIS 20:Functionality classes and evaluation methodology for deterministic random number generators,Version 2.0[Jl.Bundesamt ftir Sicherheit in der Informationstechnik(BSI),1999:5-11.

[10]W.Killmann,W.Schindler.AIS 31:A proposal for:Functionality classes and evaluation methodology for true (physical)random number generators,Version 3.1[J].Bundesamt fiir Sicherheit in der Informationstechnik(BSI),2001.

[11]W.Killmann,W.Schindler.AIS 20/AIS 31:A proposal for:Functionality classes for random number generators Version 2.0[J].Bundesamt fiir Sicherheit in der Informationstechnik (BSI),2011:44-54.

[12] Li Dawei, Feng Dengguo, Chen Hua, etc. Randomness testing specification [S]. National Cryptography Administration GM/T 0005-2012, China Standard Press, 2012.

[13] Li Dawei, Feng Dengguo, Chen Hua, etc. Information security technology binary sequence randomness detection method[S]. China National Standardization Administration GB/T 32915-2016, China Standards Press, 2016.

[14]Menezes A J,van Oorschot P C,Vanstone S A. Handbook of Applied Cryptography[M]. FL:CRC Press, 1997:178-183.

[15]Marsaglia G. The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness [EB/OL]. (1995)[2009-06-06]. http://stat.fsu.edu/ geo/diehard.html.

[16] Sulak F, Doğanaksoy A, Ege B, et al. Evaluation of randomness test results for short sequences[C]//International Conference on Sequences and Their Applications. Springer, Berlin, Heidelberg, 2010: 309-319.

[17]Turan M S, DoĞanaksoy A, Boztaş S. On independence and sensitivity of statistical randomness tests[C]//International Conference on Sequences and Their Applications. Springer, Berlin, Heidelberg, 2008: 18-29.

[18]Onur KOÇAK,Fatih SULAK,Ali DOĞANAKSOY,Muhiddin UĞUZ. MODIFICATIONS OF KNUTH RANDOMNESS TESTS FOR INTEGER AND BINARY SEQUENCES[J]. Communications Faculty of Sciences University of Ankara Series A1 Mathematics and Statistics,2018(2).

[19] HUANG Jia-lin, LAI Xue-jia. Eliminating Ability and Correlation of Random Statistical Tests. Experts' Focus, 2009.10: 43-46

[20]Xu Peifan. Implementation and Application of The Methods of Randomness Test. National Key Laboratory of Science and Technology on Communications, 2016.

[21] Chen Hua. Security Test on Cryptographic Algorithms and Design of Key Cyrptographic Components. Graduate School of the Chinese Academy of Sciences, 2004.

[22]KarellAlbo Jorge Augusto,LegónPérez Carlos Miguel,MadarroCapó Evaristo José,Rojas Omar,SosaGómez Guillermo. Measuring Independence between Statistical Randomness Tests by Mutual Information.[J]. Entropy (Basel, Switzerland),2020,22(7).

[23]G.S.Karimovich, K.Z.Turakulovich, H.I.Ubaydullayevna. Computer's source based (Pseudo) random number generation[C]//Information Science and Communications Technologies (ICISCT), 2017 International Conference on. IEEE, 2017: 1-6.

[24] J.Nakahara. 3D: A three-dimensional block cipher[C]//International Conference on Cryptology and Network Security. Springer, Berlin, Heidelberg, 2008: 252-267.

[25] S.Ariffin, N.A.M.Yusof. Randomness analysis on 3D-AES block cipher[C]//2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD). IEEE, 2017: 331-335.