İstatistikçiler Dergisi: İstatistik & Aktüerya

Journal of Statisticians: Statistics and Actuarial Sciences

IDIA 9, 2016, 1, 1-11

Geliş/Received:23.11.2015, Kabul/Accepted: 01.04.2016

www.istatistikciler.org

Araştırma Makalesi / Research Article

Statistical Testing of Cryptographic Randomness

Haydar Demirhan
Hacettepe University
Department of Statistics
06800-Beytepe, Ankara, Türkiye
haydarde@hacettepe.edu.tr

Nihan Bitirim
Council of Higher Education
06539-Çankaya, Ankara, Türkiye

Abstract

Security of a cryptographic application is highly related to the quality of randomness of the mechanism used to encrypt a message. A ciphering process used to encrypt a message is mainly based on the cryptographic random numbers. There are numerous methods proposed to generate random numbers for cryptographic applications in the literature. To decide whether a cryptographic random number generator is suitable for cryptographic applications or not, various statistical randomness tests are introduced. In practice, test batteries that contain more than one randomness test are constructed and all the tests in a battery are applied to evaluate the quality of random number generator. In this article, we present a review of test batteries and recent statistical randomness tests used to evaluate output of a cryptographic random number generator. We criticize test batteries in the sense of multiple testing problem, highlight some misuses of statistical notions in hypothesis testing of cryptographic randomness, and discuss potential solutions to multiple testing problem seen in the test batteries.

Keywords: *Chi-Square, Kolmogorov-Smirnov, hypothesis testing, multiple testing, power, random number generator, significance level, test battery, test of randomness, Type-I error, Type-II error.*

Öz

Kriptografik Rasgeleliğin İstatistiksel Olarak Test Edilmesi

Bir rasgele sayı üretecinin rasgeleliğinin değerlendirilmesi için bir test kümesindeki tüm testler ilgili üretece uygulanmaktadır. Bu çalışmada, test kümeleri ve güncel olarak önerilmiş olan kriptografik rasgelelik testleri derlenmiş, test kümeleri çoklu test problemi açısından değerlendirilmiş, bazı istatistiksel kavramların hatalı kullanımları ve çoklu test problemine ilişkin olası çözümler üzerinde durulmuştur.

Anahtar sözcükler: Çoklu test problemi, rasgele sayı üreteci, rasgelelik testi, test kümesi.

1. Introduction

Cryptographic applications are based on random numbers that have some special characteristics. Mainly, there are three classes of random numbers: true random numbers, pseudo-random numbers, and quasi-random numbers. True random numbers are based on physical sources and do not require a starting sequence, which is called seed. It is expected that they have neither a correlation pattern nor period. Pseudo-random numbers seem as random, pass through statistical randomization tests, and they are

efficiently generated by computers. However, they require a seed; hence, they are reproducible. Quasi-random numbers are generated by using special algorithms and they are well-distributed within a unit-square or unit-cube. The main disadvantages of quasi-random numbers are that quasi-random number generators lose their performance with increasing dimensionality and there is no statistical test to evaluate the quality of a quasi-random number generator [22]. In cryptographic applications, a special subset of pseudo-random numbers called cryptographic random numbers are employed. Cryptographic random numbers must satisfy very strong statistical requirements to be unpredictable. Unpredictability is directly related with contexts of autocorrelation and independence of a sequence of random variables, realizations of which constitute a cryptographic random number sequence. For a sequence of random variables, no autocorrelation and independence both imply randomness. In cryptography, randomness is the key requirement for suitability of a random number generator (RNG). In addition to context of randomness, local randomness concept is also considered in terms of cryptographic randomness [28].

A randomness analysis of output of an RNG is necessary to confirm that an RNG of interest is suitable for use in encryption processes. This analysis is made by conducting one or more statistical randomness tests. The issue of testing the random number generators has been attracted attention of researchers from 1960's to nowadays. Because the quality of randomness in used random number sequence constitutes the hearth of a ciphering process, it is very important to apprehend mechanism behind the statistical testing of randomness. A randomness test of an RNG is conducted at two stages. At the first stage, empirical distribution of a test statistic is obtained over a random number sequence being tested. At the second stage, goodness-of-fit of the empirical distribution to a theoretical distribution is tested. For instance, let us have m ones and zeros at hand. It is possible to generate 2^m different sequences with this set of zeros and ones. In a randomness test, whether these 2^m sequences occur with equal probability or not is tested. The null hypothesis of this test is "H_0: Sequences generated by the RNG of interest are random." This null hypothesis is tested at a predetermined probability of rejecting the null hypothesis when it is true, which is called significance level. Actual value of this probability is called Type-I error rate.

There are more than a hundred statistical tests that can be used to test randomness of a sequence of random numbers [19]. Because individual use of these tests would not be beneficial under some circumstances, use of their collections as test batteries is proposed in the literature [20, 24]. Each test in a test battery is applied separately to the RNG under consideration at a level of significance of α. If all or a predetermined portion of tests conclude that the RNG of interest generates random numbers, it is deduced that the degree of positive belief on randomness of the RNG is strong [2, 15, 25].

Although this manner of testing seems to be reasonable, it causes a severe problem called multiple testing problem in statistics. However, to the best of our knowledge, there is no article in the cryptography literature focusing on the test of cryptographic randomness under multiplicity. Also, we identify some mistakes in use of some important notions of statistical hypothesis testing in the cryptography literature. For a reliable and scientifically suitable hypothesis testing in such an important and critical field, these issues should be taken into consideration. With this motivation, we focus on statistical randomness tests, test batteries, and use of basic statistical hypothesis testing notions in testing the cryptographic randomness. In this article, our aim is twofold. First, we review the literature on the cryptographic randomness tests, provide basic information on test batteries as a whole, and highlight recent innovations in statistical randomness tests for cryptographic RNGs. Second, we aim at attracting attention to an appropriate and scientific way of testing randomness of an RNG. In this regard, we focus on selection and interpretation of significance level and multiple testing problem in detail. We evaluate each test battery according to the impact of conducting more than one statistical randomness test simultaneously.

In Section 2, test batteries seen in the literature are described. In Section 3, some basic and recently proposed cryptographic randomness tests not included in a test battery are mentioned. In Section 4, some statistical issues that should be handled with care are mentioned and impact of multiplicity are evaluated for each test battery in terms of statistical measures used to evaluate quality of a statistical test procedure. In Section 5, findings are summarized and discussed.

2. Test batteries

Instead of testing the quality of randomness of an RNG by using an individual statistical test, some test batteries have been formulated in the literature. An RNG is expected to pass through (the null hypothesis cannot be rejected) all or a predetermined portion of tests in a test battery of interest to be useful in cryptographic applications. In this section, we review the basic and mostly used test batteries proposed in the literature.

The first test battery for testing cryptographic randomness is the one introduced by Knuth [16, 17, 18]. This battery has survived for a long time and it is referred as a basic test battery [21]. It includes 10 statistical tests which are stated in Table 1 [18, 38]. Nowadays, it is claimed that Type-I error of Knuth test battery tends to be more than a predetermined nominal significance level [15]. This implies that the probability of deciding the randomness of a sequence when it is actually non-random is not as low as desired for the battery of Knuth.

Table 1. Main test batteries and corresponding tests and measures.

Knuth (11)*	Birthday spacings; Collision; Coupon collector's; Frequency; Gap; Maximum-of-t; Permutation; Poker; Run; Serial; Serial correlation
Diehard (16)*	Binary rank (31x31, 32x32, and 6x8 matrices); Birthday spacings; Bitstream; Count the 1s (for streams and for bytes); Craps; DNA; Minimum distance; Overlapping permutations; Overlapping sums; Overlapping-Pairs-Sparse-Occupancy; Overlapping-Quadruples-Sparse-Occupancy; Parking lot; Random spheres; Runs; Squeeze
Dieharder (26)*	All the tests in Diehard plus Marsaglia-Tsang GCD; Generalized minimum distance; Lagged sum; Permutations; Monobit; Runs; Serial (Generalized); Bit distribution; Kolmogorov-Smirnov
NIST (15)*	Approximate entropy; Binary matrix rank; Cumulative sums; Discrete Fourier transform; Frequency; Frequency within a block; Linear complexity; Maurer's "universal statistical"; Non-overlapping template; Overlapping template; Random excursions; Random excursions variant; Serial
Helsinki (4)*	Autocorrelation; Cluster; n-block; Random walk
ENT (2)**	Arithmetic mean; Chi-square test; Enrtopy test; Monte-Carlo Pi estimation; Serial correlation
Crypt-X (6)*	Binary derivative; Change point; Frequency; Linear complexity; Runs; Sequence complexity
SPRNG (13)*	Blocking; Collisions; Coupon collector; Fourier; Frequency Transform; Gap; Ising model; Maximum-of-t; Permutations; Poker; Random walk; Runs; Serial

* Shows number of tests in each battery.

** In addition to tests there are three statistical measures in this battery.

GCD: Greatest Common Divisor; DNA: Deoxyribonucleic Acid.

Marsaglia [23] introduced the Diehard test battery. It includes 12 tests mentioned in Table 1 [1]. Some of the tests are repeated with different parameter settings. Tests and their details are given by Marsaglia and Tsang [24]. It is claimed that there are several disadvantages that decreases suitability of Diehard battery of tests. The most important disadvantage of Diehard is that input parameters are fixed by software; hence, user is not allowed to change the values of parameters. Besides, there are some difficulties in data input [20].

Regarding the disadvantages of Diehard test battery, a new one called Dieharder is introduced [5]. Dieharder includes 26 fully implemented randomness tests mentioned in Table 1 [5]. It can be perceived as a novel improvement of Diehard and provides a user friendly interface and a useful and open source toolset for users of random numbers [5, 40]. Dieharder test battery is beneficial in testing the random numbers rather than bit sequences [40]. Software used to implement Dieharder battery is open source and prepared in R that works under Linux or Unix operating systems.

US National Institute of Standards and Technology (NIST) developed a test battery called NIST battery in 2001 [31, 32]. This battery is composed of 15 tests given in Table 1 [36, 38]. NIST is beneficial in testing

output of binary RNGs used in cryptographic applications. It includes existing and new randomness tests from the literature and applies chi-square goodness-of-fit test with two different degrees of freedom approaches [20, 36]. Because being defined as a standard, NIST battery is still used as a straightforward tool for formal certifications. Sadique et al. [36] review the tests included in NIST test battery. They also provide minimum required lengths of bit sequences for the tests included in NIST battery and criticise some of the tests of NIST battery in terms of CPU time and values of test statistics. In both Diehard and NIST batteries, parameters of the tests are fixed; hence, it is possible to run a test by a simple function call [20]. Although this makes these batteries more user-friendly than their existing counterparts, it decreases their flexibility.

L'Ecuyer and Simard [20] composed a C library called TestU01 that includes most of the available randomness tests and RNGs in the literature. TestU01 is able to run various combinations of these tests. It is possible to test some combinations of RNGs by using combinations of randomness tests via TestU01 suite [20, 21, 29]. It includes six predefined test batteries working for either uniform distributed random numbers or bit sequences [20]. Therefore TestU01 can be perceived as a suite of test batteries. Test batteries in TestU01 are mentioned in Table 2. Some of the tests are applied more than once with different parameter combinations under some of the batteries of TestU01. The batteries Rabbit, Alphabit, and BlockAlphabit, which applies the same tests with Alphabit under different parameter combinations, are used to test bit sequences, and the rest are used for sequences of random numbers. Although some of the NIST tests are also included in TestU01, it does not implement all tests in NIST test battery [39]. The library TestU01 is developed on ANSI C; hence, it is compiled by GNU tools instead of today's C compilers. L'Ecuyer and Simard [20] also present results of applications of test batteries of TestU01 to well-known RNGs.

Table 2. Test batteries in TestU01 suite and names of tests under each battery.

Battery name	Tests
Small Crush (10)*	Birthday Spacings; Collision; Coupon Collector; Gap; Hamming Independence; Maximum of t ; Random Walk One; Rank of a Binary Random Matrix ; Simplified Poker; Weighted Distribution
Crush (96)*	Appearance Spacings; Autocorrelation; Birthday Spacings; Close Pairs Bit Match; Closest Pairs; Collision; Collision-Permutation; Coupon Collector; Fourier 3; Gap; GCD; Hamming Correlation; Hamming Independence; Hamming Weights; Lempel Ziv; Linear Complexity; Longest Head Run; Maximum of t ; Periods in Strings; Permutation; Random Walk One; Rank of a Binary Random Matrix ; Runs; Runs of Bits; Sample Correlation; Sample Mean; Sample Product; Savir2; Serial; Simplified Poker; Sumcollector; Weighted Distribution
Big Crush (106)*	Appearance Spacings; Autocorrelation; Birthday Spacings; Closest Pairs; Collision; Collision-Permutation; Coupon Collector; Fourier 3; Gap; GCD; Hamming Correlation; Hamming Independence; Hamming Weights; Lempel Ziv; Linear Complexity; Longest Head Run; Maximum of t ; Periods in Strings; Permutation; Random Walk One; Rank of a Binary Random Matrix ; Runs; Runs of Bits; Sample Correlation; Sample Mean; Sample Product; Savir2; Serial; Simplified Poker; Sumcollector; Weighted Distribution
Rabbit (38)*	Appearance Spacings; Autocorrelation; Close Pairs Bit Match; Fourier 1; Fourier 3; Hamming Correlation; Hamming Independence; Hamming Weights; Lempel Ziv; Linear Complexity; Longest Head Run; Multinomial Bits; Periods in Strings; Random Walk One; Rank of a Binary Random Matrix ; Runs
Alphabit (17)*	Multinomial Bits; Hamming Correlation; Hamming Independence; Random Walk One
BlockAlphabit (17)*	Multinomial Bits; Hamming Correlation; Hamming Independence; Random Walk One

* Shows number of tests in each battery.

In addition to these test batteries, there are also small scale test batteries that have limited impact in the literature. ENT is another test battery proposed by Walker [42]. ENT has 5 very basic statistics given in Table 1 [14]. Two of these statistics provide statistical goodness-of-fit tests and the rest are statistical measures used to evaluate quality of randomness. The web site ``random.org'' uses ENT battery to test its

random numbers [15, 25]. Vattulainen et al. [40] proposed a test battery based on Ising model and random walks on lattices. This battery is called Helsinki by Rutti [33]. Tests included in Helsinki battery are mentioned in Table 1 [41]. Information Security Research Center at Queensland University of Technology introduced another test battery called Crypt-X, which was for commercial purposes, in 1998 [39]. It includes 6 tests given in Table 1 [38]. SPRNG test battery was proposed in 2000. It includes some tests from Knuth test battery and some new additions [26]. SPRNG has 10 randomness tests mentioned in Table 1 to evaluate randomness of serial and parallel random number sequences. Also, SPRNG includes some RNGs tested by its test battery. Rutti [33] evaluated existing test batteries and composed a new test battery with the tests mentioned in Table 1. It consists 37 statistical and physical randomness tests [33]. This test battery is a combination of Knuth, Helsinki, Diehard, and SPRNG batteries.

3. Tests not included by a test battery

There are numerous randomness tests not included in a test battery and can be used to test randomness of an RNG. Most of these tests are proposed to formulate a universal test rather than testing randomness of an RNG by using batteries.

Maurer [27] proposed a statistical test for random bit generators that does not include disadvantages of preceding tests. This test is able to detect deviations from statistics of binary symmetric sources and it determines cryptographic significance of an inadequacy of an RNG by measuring per-bit entropy. This work of Maurer [27] provides a guideway to statistical testing of randomness.

Hernandez et al. [12] proposed a new test called SAC, namely Strict Avalanche Criterion, and tested some RNGs by SAC. It is possible to apply the SAC test to sequences longer than 32 bits efficiently in terms of computational time. Hernandez et al. [12] compared performance of SAC test with Frequency, GCD, Bday, Gorilla, and Collision tests over seven RNGs and concluded that the SAC test is more powerful than some of the classical randomness tests.

Ryabko and Monarev [35] proposed an adaptive chi-square test that is suitable for testing with smaller sample sizes than those required for the classical chi-square test. They conducted an experimental study with Rijndael and RC6 block chippers over English and Russian texts of various lengths to evaluate performance of their new test. Consequently, they obtained that their test identifies non-randomness more efficiently than chi-square test in small samples [35].

Ryabko and Monarev [34] proposed “Book Stack” and “Order” tests for testing binary random bit sequences. These tests are based on the contexts of entropy and universal coding, respectively. They compare these tests with some standard tests and the one proposed by Ryabko et al. [35]. After a limited simulation study, it was observed by Ryabko and Monarev [34] that Book Stack test rejects the null hypothesis stating randomness of an RNG easier than Order test. Thus, Book Stack test is found more conservative than Order test. Accordingly, it is obtained that an information theoretic approach is useful in randomness tests of RNGs.

Random walk is a stochastic process composed of a sequence of variations with random magnitudes and directions. It is possible to analyse binary sequences in detail by using random walk process. In the literature, there are various tests based on random walk. Because related test statistics are based on approximate distributions, these tests are not applicable to short sequences. Doganaksoy et al. [7] proposed three randomness tests based on random walk process. In these tests, it is possible to calculate exact probabilities used to make decision on the null hypothesis. Therefore, in contrast to existing tests based on random walk, those of Doganaksoy et al. [7] are applicable to short sequences. Doganaksoy et al. [8] proposed another group of randomness tests based on randomness postulates of Golomb. Sequences satisfying the postulates of Golomb are called pseudonoise sequences [8]. Recently, Doganaksoy et al. [8] utilized postulates on the runs of lengths and proposed tests based on runs of lengths one, two, and three. The chi-square test is employed as the goodness-of-fit test within their test process.

Alcover et al. [2] proposed "Topological Binary Test" to test randomness in bit sequences. This test works on different bit patterns of pre-determined length in the sequence of interest. If many different bit patterns are obtained, this leads to acceptance of the null hypothesis stating the randomness of considered RNG. Distribution of the resulting test statistic is exact rather than approximate; hence, this test is also applicable to short bit sequences. Besides, fast computing is another advantage of this test [2].

4. Statistical issues in testing the randomness

In this section, we focus on selection and interpretation of significance level and multiplicity problem in simultaneous application of more than one hypothesis tests.

4.1. Significance level and Type-I error

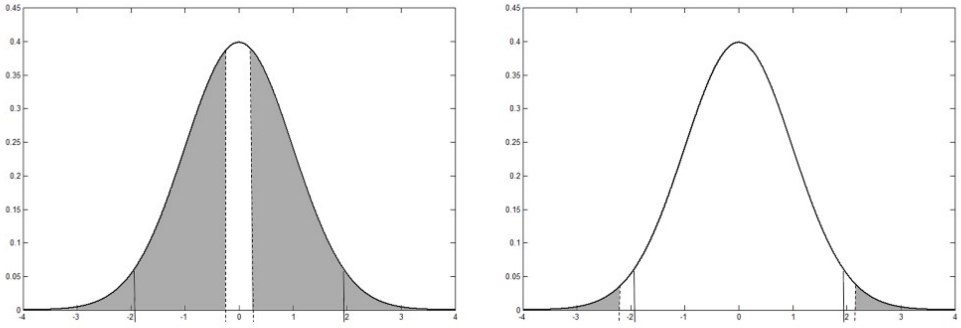
Significance level and Type-I error rate are briefly defined in the introduction. A more formal definition of Type-I error, denoted by α, and related concepts are given in Table 3 [10]. In the cryptography context, if we decide non-randomness of an RNG while it is actually generating random numbers, we commit a Type-I error, which is also called false positive decision. Whereas, if we decide randomness of an RNG while it is not generating random numbers, we commit a Type-II error, which is called false negative decision and denoted by β. The context of statistical power of a hypothesis test is closely related with Type-II error. For a randomness hypothesis, power, denoted by $1-\beta$, is the probability of deciding non-randomness of an RNG while it is actually non-random. It measures the chance of identifying a non-random RNG correctly.

Table 3. Outcomes and related errors in hypothesis testing.

	Fail to reject null hypothesis (a non-significant result)	Reject null hypothesis (a significant result)
Null hypothesis is true	Correct decision	Type-I error
Null hypothesis is false	Type-II error	Correct decision

Significance level constitutes a pre-determined value for the Type-I error. To decide rejection or approval of the null hypothesis, we obtain a probability, namely *p-value* that measures the degree of support for the null hypothesis provided by observed data. Let Z be the test statistic and C be the region determined by the observed value of Z. Statistically, p-value corresponds to the probability $\mathrm{P}(Z \epsilon C | H_0)$, where H_0 is the null hypothesis [11]. To reach a decision on the randomness of an RNG, p-value is compared with a pre-determined α. If p-value is smaller than α, we reject the null hypothesis. The smaller the p-value, the more confidently we decide non-randomness of an RNG (see Nuzzo [30] for a comprehensive evaluation of the context of p-value). However, this inference is not valid for the value of α. Because value of α is directly related to the decision about randomness of an RNG, it should be determined with caution.

Alani [1] proposed an approach for the interpretation of randomness test results. Alani [1] considered normal distribution as the reference distribution. It is stated by Alani [1] that "p-values near 0 or 1 indicate deviation from normal distribution." This implies that a p-value is interpreted in a symmetric fashion. Based on this symmetric interpretation, he divides the set [0,1) into safe (0.25,0.75), doubt ((0.1,0.25] or [0.75,0.9)), and failure ((0,0.1] or [0.9,1]) areas, and uses the numbers of test results fall in these areas to judge the degree of deviation from randomness. We illustrate the relationship between rejection region and p-value in Figure 1. A p-value near 1, namely $\mathrm{P}(Z \epsilon C | H_0) \cong 1$, implies that the probability of having a value of employed test statistic within the rejection region is nearly impossible; hence, it is nearly impossible to either reject a null hypothesis or infer a deviation from randomness. Based on a p-value near 1, one can conclude randomness of an RNG with virtual certainty. An example of this situation is seen in panel (a) of Figure 1. For a p-value near zero, we can confidently reject the null hypothesis as seen in panel (b) of Figure 1. Consequently, a p-value should always be interpreted in one-sided manner by directly comparing a p-value with α.



(a) (b)

Figure 1. Illustration of rejection region and p-value under the standard normal curve. In each panel, solid lines and shaded areas represent boundaries of rejection region for $\alpha = 0.05$ and region corresponding to p-value, respectively. In panel (a), p-value is equal to $0.72 (> 0.05)$; hence, null hypothesis cannot be rejected. In panel (b), p-value is equal to 0.036 (< 0.05); hence, null hypothesis is rejected.

L'Ecuyer and Simard [20] provided results on periods, CPU times, and numbers of statistical tests with a p-value outside the interval$[10^{-10}, 1 - 10^{-10}]$. This interval implies that the nominal α level used for testing is $2 \cdot 10^{-10}$. This choice of α is far from the common manner and it is somewhat problematic. Difference between using $2 \cdot 10^{-10}$ and a common choice of 0.05 is illustrated in Figure 2 over the chi-square distribution with 10 degrees of freedom. As seen in Figure 2, we have an invisible rejection region at $2 \cdot 10^{-10}$ significance level. Actually, total area under the curve is $2 \cdot 10^{-10}$ for any degrees of freedom. Thus, it is nearly impossible to reject the null hypothesis stating the randomness of an RNG. However, we have a clear and appropriate rejection region, shaded area under the curve in Figure 2, at 0.05 α level. Although any value greater than zero can be assigned to α, it should be chosen appropriately. When we decrease the value of α, correspondingly, we increase the probability of deciding that the RNG of interest generates random numbers while it is not generating random sequences actually. Therefore, it is very important to use an appropriate value for α regarding the delicacy of matter in cryptography.

4.2. Multiple testing problem on test batteries

Multiple testing problem, also called multiplicity problem, is one of the basic problems seen in multiple hypothesis testing. To illustrate the problem, let us have k tests in a test battery and suppose that tests in the battery are conducted at a significance level of α. We have the following result on the probability of having at least one significant result:

$$\mathrm{P}(\{\text{at least one significant result}\}) = 1 - \mathrm{P}(\{\text{no significant result}\}) = 1 - (1 - \alpha)^k \qquad (1)$$

For example, with $k = 5$ and $\alpha = 0.05$, we have a 23% chance of deciding that sequences generated by an RNG of interest is not random in at least one of the tests, even if all of the tests actually indicate that the sequences are random. When we simultaneously use more than one test to evaluate randomness of an RNG, the probability of rejecting the null hypothesis simply due to chance increases with increasing values of k. It is apparently seen that one should regard the multiple testing problem in statistical testing of cryptographic randomness.

For a test battery, L'Ecuyer and Simard [20] highlight importance of having tests with different characteristics to identify deviations from randomness in different conditions. They remark the need for small batteries to increase computational efficiency, and state that small batteries may be more efficient in the detection of gross defects in RNGs or errors in their implementation. In fact, L'Ecuyer and Simard [20] intuitively describe the multiplicity problem in terms of computational efficiency. As the result of decreasing the number of tests in a battery, we not only gain computational speed but also decrease the

probability of incorrectly deciding non-randomness of a battery. Thus, we make more reliable decisions on the randomness of an RNG.

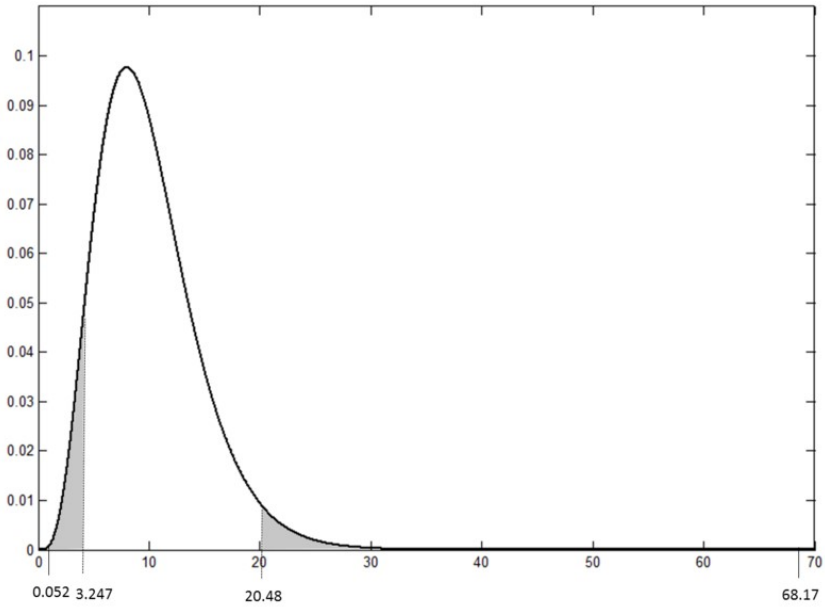


Figure 2. Critical values of a two-sided chi-square test for $2 \cdot 10^{-10}$ (solid lines) and 0.05 (dashed lines) α levels. Shaded areas represent rejection regions where the null hypothesis is rejected at 0.05 α level. Because L'Ecuyer and Simard (2007) do not provide information about degrees of freedom of chi-square statistics, we use a moderate value 10 as degrees of freedom for illustrative purpose.

Alcover et al. [2] claim that instead of a specific test, multiple tests together constitute a useful tool to study randomness of an RNG. They base this idea on the fact that some of the tests in a test battery may commit a Type-II error while the rest reach a correct decision. Thus, each test has its own strengths, and use of tests as a battery gives us the chance of utilizing strengths of tests. However, when we use more than one test at the same time, we increase the chance of committing a Type-I error. In this case, it is hard to take advantages of some tests due to the increasing Type-I error.

For the considered test batteries, probabilities of incorrectly identifying at least one random sequence as non-random for various values of significance level are presented in Table 4. For a reliable and appropriate hypothesis testing, we consider two main characteristics of the considered test. The first is the ability of controlling nominal significance level and the second is the power of test. As seen in Table 4, none of the batteries is able to control α due to the multiplicity problem. Decreasing the nominal α level seems to be a solution of the problem of incorrectly identifying a random sequence as non-random. However, in this case, we decrease the area of rejection region and it becomes almost impossible to reject the randomness hypothesis.

The most common solution of multiplicity problem is to use a Bonferroni correction [9, 13]. Although it is mentioned in the literature of cryptography (see for example, Bogdanov et al. [4]), to the best of our knowledge, Bonferroni correction has not been used in randomness tests of a test battery.

Bonferroni correction is based on scaling nominal significance level for each test in a set of hypothesis tests. Let p_i be the p-value obtained for ith test for $i = 1, \ldots, k$. In a Bonferroni corrected test, we reject ith null hypothesis if $p_i < \alpha/k$. For example, with $k = 5$ and $\alpha = 0.05$, the probability of having at least one significant result is obtained from eq. (1) as $1 - (1 - 0.05/5)^5 = 0.049$. Thus, it is possible to control α level successfully with a Bonferroni correction. However, Bonferroni corrected tests are found to be conservative in many studies (see for example Demirhan et al. [6]). There are modifications of Bonferroni correction such as Dunn-Sidak correction provided that the tests are independent. In a Dunn-Sidak corrected test, we reject ith null hypothesis if $p_i < 1 - (1 - \alpha)^{1/k}$ [9, 37]. In the example with

$k = 5$ and $\alpha = 0.05$, the probability of having at least one significant result is obtained from eq. (1) as $1 - \left(1 - \left(1 - (1 - 0.05)^{1/5}\right)\right)^5 = 0.05$. To control α, there are also sequential versions of the family of Bonferroni corrections and other approaches working with a similar logic [3]. Although these approaches successfully control α, their probability of correctly deciding non-randomness of a sequence while it is actually non-random is not as high as desired. This is the common deficiency of tests with Bonferroni correction and its adaptations [6].

Table 4. Probability of incorrectly identifying at least one random sequence as non-random.

		Nominal significance level (α)			
Test Battery	k	0.001	0.01	0.05	0.1
Knuth	11	0.011	0.105	0.431	0.686
Diehard	16	0.016	0.149	0.560	0.815
Dieharder	26	0.026	0.230	0.736	0.935
NIST	15	0.015	0.140	0.537	0.794
TestU01-Small Crush	10	0.010	0.096	0.401	0.651
TestU01-Crush	96	0.092	0.619	0.993	1.000
TestU01-Big Crush	106	0.101	0.655	0.996	1.000
TestU01-Rabbit	38	0.037	0.317	0.858	0.982
TestU01-Alphabit	17	0.017	0.157	0.582	0.833
TestU01-Block Alphabit	17	0.017	0.157	0.582	0.833
Helsinki	4	0.004	0.039	0.185	0.344
ENT	2	0.002	0.020	0.098	0.190
Crypt-X	6	0.006	0.059	0.265	0.469
SPRNG	13	0.013	0.122	0.487	0.746

k is the number of tests applied in each battery.

Application of multiple randomness tests to evaluate randomness of an RNG's output seems to be reasonable to utilize strength of included tests in different cases. However, one should use a correction method or an approach that does not affected by multiplicity to make a reliable decision on the randomness of an RNG. A test battery including a small number of tests with high power can be applied with Bonferroni corrections.

5. Conclusions

Randomness of a sequence generated by a random number generator is evaluated by using some statistical hypothesis tests. These randomness tests are applied either individually or together. Some tests are collected under test batteries or test suites including batteries. The logic behind simultaneous application of more than one test is to utilize different advantages of tests for different cases seen in number sequences. However, there are vital inconveniences of application of more than one statistical hypothesis test. Also, appropriate use and interpretation of some notions of statistical hypothesis tests is of crucial importance.

In this article, we consider statistical testing of randomness of a sequence generated by a random number generator. We review test batteries and basic and recent individual randomness tests in the cryptography literature. Then, we focus on some misuses of statistical notions in the randomness tests of random number generators, and mention multiple testing problem and its possible solutions for test batteries.

Pre-determined significance level of a randomness test should be chosen and interpreted cautiously. An appropriate value of significance level may be chosen as 0.05 or 0.01. Lower values than 0.01 make rejection of null hypothesis harder; and hence, the probability of incorrectly deciding randomness of a random number generator synthetically increases.

As for the multiple testing problem, one should use a Bonferroni correction or another method to control observed Type-I error level at a nominal value. A small test battery composed of tests with high power can be employed. As another possible solution, multiple comparison procedures used to conduct multiple pairwise hypothesis tests can be adapted for randomness testing by a test battery.

Acknowledgements

This work was supported by The Scientific and Technological Council of Turkey (TUBITAK) under the project 114F249 of ARDEB-3001 programme.

References

[1] M.M. Alani, 2010, Testing randomness in ciphertext of block-ciphers using diehard tests, *International Journal of Computer Science and Network Security*, 10:53–57.

[2] P.M. Alcover, A. Guillamon, M.C. Ruiz, 2013, A new randomness test for bit sequences, *Informatica*, 24:339–356.

[3] B. Walsh, 2006, Multiple comparisons: Bonferroni corrections and false discovery rates. http://nitro.biosci.arizona.edu/workshops/Aarhus2006/pdfs/Multiple.pdf, Lecture notes for EEB 581, [Online; accessed 19-December-2014].

[4] D. Bogdanov, L. Kamm, S. Laur, V. Sokk, 2014 Rmind: a tool for cryptographically secure statistical analysis. *IACR Cryptology*, ePrint Archive, 512.

[5] R.G. Brown, D. Eddelbuettel, D. Bauer, 2014, Dieharder: A random number test suite (version 3.31.1). URL:
http://www.phy.duke.edu/ rgb/General/dieharder.php, [Online; accessed 25-February-2014].

[6] H. Demirhan, N.A. Dolgun, Y. Parlak Demirhan, M.O. Dolgun, 2010, Performance of some multiple comparison tests under heteroscedasticity and dependency, *Journal of Statistical Computation and Simulation*, 80:1083–1100.

[7] A. Doganaksoy, C. Calik, F. Sulak, M.S. Turan, 2006, New randomness tests using random walk. In *Proceedings of National Cryptology Symposium II*, Turkey.

[8] A. Doganaksoy, F. Sulak, M. Uguz, O. Seker, Z. Akcengiz, 2015, New statistical randomness tests based on length of runs, Mathematical Problems in Engineering, http://dx.doi.org/10.1155/2015/626408.

[9] O.J. Dunn, 1961, Multiple comparisons among means, *Journal of the American Statistical Association*, 56:52–64.

[10] J.E. Freund, B.M. Perles, 2003, *Statistics A First Course*, Prentice Hall, New Jersey, 8 edition.

[11] S. Geza, 2007, Introduction to Probability with Statistical Applications, Birkhauser, Boston.

[12] J.C. Hernandez, J.M. Sierra, A. Seznec, 2004, The sac test: a new randomness test, with some applications to PRNG analysis. *Proceedings of the International Conference Computational Science and Its Applications*, 960–967.

[13] Y. Hochberg, A.C. Tamhane, 1987, *Multiple Comparison Procedures*, Wiley, New York.

[14] I. Jang, H.S. Yoo, 2006, Pseudorandom number generators using optimal normal basis. In Kumar V. Tan C.J.K. Taniar D. Lagan A. Mun Y. Choo H. Gervasi, O. (eds.), *Computational Science and Its Applications* - ICCSA 2006, Part II, 206–212. Springer-Verlag.

[15] C. Kenny, 2005, Random number generators: An evaluation and comparison of random.org and some commonly used generators, Trinity College Dublin, management science and information systems studies project report. https://www.random.org/analysis/Analysis2005.pdf, 2005. [Online; accessed 24-February-2014].

[16] D.E. Knuth, 1969, *The Art of Computer Programming, Volume 2 / Seminumerical Algorithms*, Addison-Wesley, Reading, Massachusetts, 1 edition.

[17] D.E. Knuth, 1981, *The Art of Computer Programming, Volume 2 / Seminumerical Algorithms*, Addison-Wesley, Reading, Massachusetts, 2 edition.

[18] D.E. Knuth, 1998, *The Art of Computer Programming, Volume 2 / Seminumerical Algorithms*, Addison-Wesley, Reading, Massachusetts, 3 edition.

[19] P. L'Ecuyer, P. Hellekalek, 1998, *Random number generators: Selection criteria and testing*, Random and Quasi-Random Point Sets Lecture Notes in Statistics, 138:223–265.

[20] P. L'Ecuyer, R. Simard, 2007, Testu01: A c library for empirical testing of random number generators. ACM Transactions on Mathematical Software, 33: Article 22.

[21] P. LEcuyer, R. Simard, 2014, Testu01: *A C library for empirical testing of random number generators - user's guide, compact version*. Testu01. http://www.iro.umontreal.ca/ simardr/testu01/guideshorttestu01.pdf, [Online; accessed 24-February-2014].

[22] H. Maaranen, K. Miettinen, M.M. Mkel, 2003, Using Quasi Random Sequences in Genetic Algorithms, *Optimization and Inverse Problems in Electromagnetism*, 33–44. Springer, New York.

[23] G. Marsaglia, 2014, *Diehard: A battery of tests of randomness*. http://stat.fsu.edu/ geo/diehard.html, 1996. [Online; accessed 25-February-2014].

[24] G. Marsaglia, W.W. Tsang, 2002, Some difficult-to-pass tests of randomness, *Journal of Statistical Software*, 7:3.
[25] K. Marton, A. Suciu, C. Sacarea, O. Cret, 2012, Generation and testing of random numbers for cryptographic applications, *The Publishing House of the Romanian Academy*, 4:368–377.
[26] M. Mascagni, A. Srinivasan, 2000, Algorithm 806: SPRNG: A scalable library for pseudorandom number generation, *ACM Transactions on Mathematical Software*, 26:436–461.
[27] U.M. Maurer, 1992, A universal statistical test for random bit generators, *Journal of Cryptology*, 5:89–105.
[28] U.M. Maurer, J.L. Massey, 1991, Local randomness in pseudorandom sequences, *Journal of Cryptology*, 4:135–149.
[29] B.D. McCullough, 2006, A review of Testu01, *Journal of Applied Econometrics*, 21:677–682.
[30] R. Nuzzo, 2014, Scientific method: Statistical errors, *Nature*, 506(7487):150.
[31] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S. Vo., 2001, http://csrc.nist.gov/rng/. [Online; accessed 25-February-2014].
[32] A.L. Rukhin, 2001, Testing randomness: A suite of statistical procedures, *Theory of Probability and Its Applications*, 45:111–132.
[33] M. Rutti, 2004, *A Random Number Generator Test Suite for the C++ Standard*. Diploma Thesis. Institute for Theoretical Physics, ETH Zurich.
[34] B.Y. Ryabko, V.A. Monarev, 2005, Using information theory approach to randomness testing, *Journal of Statistical Planning and Inference*, 133:95–110.
[35] B.Y. Ryabko, V.S. Stognienko, Yu.I. Shokin, 2004 A new test for randomness and its application to some cryptographic problems, *Journal of Statistical Planning and Inference*, 123:365–376.
[36] J.K.M. Sadique, U. Zaman, R. Ghosh, 2012 Review on fifteen statistical tests proposed by NIST, *Journal of Theoretical Physics and Cryptography*, 1:18–31.
[37] Z.K. Sidak, 1967, Rectangular confidence regions for the means of multivariate normal distributions, *Journal of the American Statistical Association*, 62:626–633.
[38] J. Soto, 1999, Statistical testing of random number generators. *Proceedings of the 22nd National Information Systems Security Conference*, USA, National Institute of Standards and Technology.
[39] M. Sys, Z. Ryha, 2014, Faster randomness testing with the NIST statistical test suite. In Chakraborty R.S. et al. (eds.), *Security, Privacy, and Applied Cryptography Engineering Lecture Notes in Computer Science*, 272–284, Portugal, Springer.
[40] M. Sys, P. Svenda, M. Ukrop, V. Matyas, 2014, Constructing empirical tests of randomness. In Andreas Holzinger Mohammad S. Obaidat and Pierangela Samarati (eds.), *SECRYPT 2014 Proceedings of the 11th International Conference on Security and Cryptography*, 229–237, Portugal, SCITEPRESS Science and Technology Publications.
[41] I. Vattulainen, T. Ala-Nissila, K. Kankaala, 1995 Physical models as tests of randomness, *Physical Review Engineering*, 52:3205–3213.
[42] J. Walker, 2014, *Ent - a pseudorandom number sequence test program*. https://www.fourmilab.ch/random/, [Online; accessed 25-February-2014].