Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 655 – 665
© TÜBİTAK
doi:10.3906/elk-1503-214

Mutual correlation of NIST statistical randomness tests and comparison of their sensitivities on transformed sequences

Ali DOĞANAKSOY[1], **Fatih SULAK**[2], **Muhiddin UĞUZ**[1], **Okan ŞEKER**[1], **Ziya AKCENGİZ**[1,*]
[1]Institute of Applied Mathematics, Middle East Technical University, Ankara, Turkey
[2]Mathematics Department, Atılım University, Ankara, Turkey

Received: 24.03.2015 • **Accepted/Published Online:** 03.09.2015 • **Final Version:** 10.04.2017

Abstract: Random sequences are widely used in many cryptographic applications and hence their generation is one of the main research areas in cryptography. Statistical randomness tests are introduced to detect the weaknesses or nonrandom characteristics that a sequence under consideration may have. In the literature, there exist various statistical randomness tests and test suites, defined as a collection of tests. An efficient test suite should consist of a number of uncorrelated statistical tests each of which measures randomness from another point of view. 'Being uncorrelated' is not a well-defined or well-understood concept in the literature. In this work, we apply Pearson's correlation test to measure the correlation between the tests.

In addition, we define five new methods for transforming a sequence. Our motivation is to detect those tests whose results are invariant under a certain transformation. To observe the correlation, we use two methods. One is the direct correlation between the tests and the other is the correlation between the results of a test on the sequence and its transformed form. In light of the observations, we conclude that some of the tests are correlated with each other. Furthermore, we conclude that in designing a reliable and efficient suite we can avoid overpopulating the list of test functions by employing transformations together with a reasonable number of statistical test functions.

Key words: Cryptography, statistical randomness tests, correlation, transformations, NIST test suite

1. Introduction

The numbers selected from a finite set equally likely are called random numbers. Random numbers are used in many fields of cryptography such as authentication systems, digital signature schemes, and zero knowledge protocols. Moreover, key generations rely on random values. Using weak random values in key generations can lead to some deficiencies in the system and hence an adversary can gain enough information to break the system. Testing the randomness of a sequence is an essential part of the security evaluation of a cryptographic algorithm and it is performed by statistical randomness tests. Each statistical randomness test is designed to evaluate a different characteristic of the sequence.

Random values can be generated by using a true random source such as radioactive decay and atmospheric noise. On the other hand, using a true random source can cause various problems, as reproducing them is nearly impossible. Due to this disadvantage, deterministic processes can be used to produce pseudorandom, that is, random looking sequences.

Throughout the paper, we assume that the numbers are chosen from the set $\{0,1\}$. A randomness test

[*] Correspondence: ziyaakcengiz@gmail.com

defines a measure of being random. In order to define the randomness of a binary sequence three important properties are stated as follows [1]:

– Uniformity: All zeros and ones in the sequence should be uniformly distributed.

– Independence: The terms of random sequences should be independent.

– Unpredictability: It should be infeasible to predict the following bits when t bits of the sequence are known.

Moreover, the well-known postulates of Golomb [2] form a base for defining randomness. In the literature, there exist a number of statistical test packages to test randomness. The first collection of randomness tests, presented by Knuth, involves 11 statistical tests, CRYPT-X involves 8 statistical tests, DIEHARD involves 15 tests, and a new version of it, DIEHARDER, involves 26 statistical tests, TESTU01 involves 16 tests, and the test suite published by NIST originally involves 16 tests [3–8].

For different purposes, different sets of tests can be arranged. The test suite should be large enough to obtain reliable results. However, a blindly populated test suite does not necessarily mean it will be reliable. Therefore, selecting the appropriate tests is the basic and crucial step of forming a test suite. As stated by Soto [9], the tests included in the suite should be independent. Therefore, the selection should be done in such a manner that tests should not be correlated with each other. Such a selection increases the efficiency of the suite.

In the literature, there exist several papers dealing with the classification and correlation of the tests. In 2008 Turan et al. [10] focused on the independence of randomness tests and its effect on the coverage of test suites. Moreover, the concept of sensitivity is introduced in order to analyze the effects of transformations. Doğanaksoy et al. [11] deal with the dependency of the tests in the NIST test suite and analyze the sensitivity of these tests and in this paper emphasize the correlation test. Moreover, a new method of evaluating the correlation of tests is proposed by Fan et al. [12]. An important result is achieved in [13]. Accordingly, two tests included in the NIST test suite are highly correlated.

Throughout this study, we propose a method for analysis of statistical randomness tests. For the experiments we use 14 (NIST discarded the fast Fourier transform test from the test suite due to a problem discovered in 2009 and the Lempel–Ziv complexity test is not included in the last updated version) statistical tests included in the NIST test suite. Pearson's correlation coefficient is used in order to measure the correlation between the tests. Moreover, five new transformation methods are defined to observe the sensitivity of the tests. Our main aim is to detect whose results are associated with each other on the same transformations. In order to observe the classification of statistical tests we apply them to pseudo-random binary sequences and the transformed form of these sequences. According to the test results, we investigate the correlations. In section 1 we set out the introduction. In section 2, as a preliminary work, we explain correlation of randomness tests and define transformations. We also explain other correlation methods in the literature. In section 3 we give the correlation results and before we conclude our paper in chapter 4 we state the correlation results. Finally, in section 5, we summarize the paper and state the topics for future work.

2. Preliminaries

2.1. Statistical randomness tests

In this section we define and state the importance of the correlation for tests and then state the transformation methods. Statistical randomness tests are defined to observe characteristic vulnerabilities that a sequence could have. A characteristic of a random variable, whose distribution is known for a sequence, is chosen for each statistical test. Output of a statistical test, called the P-value, between 0 and 1 is calculated according to the distribution of the random variable. The closer the value of the P-value to 1, the stronger the randomness of the sequence is considered.

2.2. Correlation

The correlation coefficient, which is between –1 and 1, actually shows the linear dependence between P-values. The closer the absolute value to 1 the stronger the correlation. One should be aware that if two tests are independent, then the corresponding correlation coefficient is 0; however, the reverse is not true in general [14]. The only difference between correlation coefficient –1 and 1 is that the former corresponds to a negative correlation, i.e. inverse proportion, and the latter means a positive correlation, i.e. direct proportion.

As stated in Section 1, there exist different studies on correlation and independence of statistical randomness tests. In these studies different types of correlation methods are used. Turan et al. [10] made this correlation by focusing on the proportion of regions in which P-values are lower than 0.01 in each test. Moreover, some transformations are defined, according to which the sensitivity of tests is shown. In a similar way, correlation is defined by examining at the distribution of a test's results in the region where the other test's results are lower than 0.01 [11]. Another method is defined by Fan et al. [12]. In this method, firstly P-values are calculated by using two different sets. For each sequence, the difference between 2 P-values corresponding to two different tests is calculated and according to the distribution of difference correlation is found.

As stated in Section 1, we use Pearson's correlation coefficient. Correlation is measured by examining at mutual correlation of test results. For the experiments we use about 200,000 sequences of length 2^{10} in order to observe the relation among tests applicable to short sequences. As for tests applicable for long sequences we use 200 sequences of length 2^{20} to see the relation between tests for long sequences. For both groups, pseudorandom sequences are generated by adding Advanced Encryption Standard (AES) [15] outputs consecutively. We also examine the same correlations for tests results on transformed sequences. In doing so we not only examined the test in itself but we also studied the relations of these tests on transformed sequences' results with each other.

2.3. Transformation methods

In this section we define the transformations that we use in correlation analysis of the statistical tests. The purpose of this study is to compare the results given by the tests on transformed sequences. If the results of two tests for the transformed sequences are similar, then it can be said that there is a structural similarity between these tests.

When complex transformations are applied, if some relations between the tests are found, it will not be easy to understand the relations between tests. Therefore, the simplest level of transformations is used to observe how the tests results vary with the transformations. After this point of the section, we give explanations of the described transformations and simple examples for each transformation. Throughout the paper, we denote a binary sequence of length n as $S_n = a_0, a_1, \ldots, a_{n-1}$ and its transformed sequence as $S_n^{'} = a_0^{'}, a_1^{'}, \ldots, a_{n-1}^{'}$.

Reversing: This transformation reverses the order of the sequences. In other words,

$a_i^{'} = a_{n-1-i}$ for $i = 0, 1, 2, \ldots, (n-1)$.

Binary derivative: This transformation generates a new sequence by *XOR*ing two consecutive bits. The transformed sequence can be constructed as follows:

$$a_i^{'} = \begin{cases} a_i + a_{i+1}\, if\, i = 0, 1, \ldots, n-2 \\ a_0\, if\, i = n-1 \end{cases}$$

Remark 1. The generation of the last bit can be changed. For example, the last bits can be fixed with 1; however, this can lead to loss of information for higher order derivatives.

Example 1. Let $S_{16} = 1001011010011110$ be a binary sequence of length 16.

$$a_0^{'} = a_0 \oplus a_1\,, \quad a_1^{'} = a_1 \oplus a_2, \ldots, \quad a_{14}^{'} = a_{14} \oplus a_{15}, \quad a_{15}^{'} = a_0 = 1.$$

$$S_{16} = \ 1001011010011110 \Rightarrow S_{16}^{'} = \ 1011101110100011.$$

t-Rotation: This transformation moves the first t bits of the sequence to the end. Bits of sequence are cyclically shifted from the beginning and added at the end.

$$a_i^{'} = a_{(t+1) mod\, n} \text{ for } i = 0,\, 1,\, 2, \ldots,\, (n-1).$$

Example 2. 4-rotation of S_{16} is

$$S_{16} = \mathbf{1001}011010011110 \Rightarrow S_{16}^{'} = 011010011110\mathbf{1001}$$

Masking bits: Masking a sequence a_n with a mask M_n^t means taking *XOR* of corresponding bits. In other words,

$$a_i^{'} = a_1 + m_i^t \text{ for } i = 0,\, 1,\, 2, \ldots,\, (n-1).$$

– Mask-0: This transformation is also called complementation. M_n^0 is defined as follows:

$$m_i^0 = 1 \text{ for } i = 0,\, 1,\, 2, \ldots,\, (n-1).$$

– Mask-t: For sequences of length n we define n different masking sequences denoted by M_n^1, M_n^2, M_n^4,..., M_n^n, where M_n^t is defined by the following formula:

$$m_i^t = \begin{cases} 1 & \text{if} \quad (imod2t) < t \\ 0 & \text{if} \quad (imod2t) \geq t \end{cases} .$$

Example 3. Let $a_{16} = 1001011010011110$; then for each mask we get a different transformed sequence. That is,

Mask-0=1111111111111111, Mask-1=1010101010101010

Mask-2=1100110011001100, Mask-4=1111000011110000

Mask-8=1111111100000000.

$$a_i^{'} = a_i \oplus m_i^t \text{ for } i = 0,\, 1,\, 2, \ldots,\, (n-1) \text{ and } t = 0,\, 1,\, 2,\, 4,\, 8.$$

Swapping: This transformation swaps exactly $\sqrt{n}$ -bits of a sequence of length n. We choose those bits to be swapped in a deterministic way rather than chosen random bits. Initially, the sequence is divided into subsequences of length $2k$, where $k = [\sqrt{n}/2]$. For each $i = 0,\, 1,\, 2, \ldots,\, (k-1)$ we interchange the first bits of $(2i)^{th}$ and $(2i+1)^{th}$ subsequences. In this study, k is chosen as $\sqrt{n}$. The following algorithm explains the details of the transformation.

Example 4. Let $S_{16} = 1000100111010010$ be a binary sequence of length 16. Therefore, $k = \left[\sqrt{16}/2\right] = 2$; then subsequences are

$A_0 = 10,\ A_1 = 00,\ A_2 = 10,\ A_3 = 01,\ A_4 = 11,\ A_5 = 01,\ A_6 = 00,\ A_7 = 10$Then the transformed sequence is

$$S_n^{'} = 0010001101111000$$

Algorithm 1 Swap Transformation

```
let $a_n$be a binary sequence of length $n$.

$k \leftarrow Sqrt(n)/2$
$i \leftarrow 0,$
While $i \leq 2 \cdot k$ do
        $temp = a_{2\cdot k\cdot i}$
        $a_{2\cdot k\cdot i} = a_{k\cdot(2\cdot i+1)}$
        $a_{k\cdot(2\cdot i+1)} = temp$
        $i \leftarrow i + 1$
End while
$a_n' \leftarrow a_n$
Return $a_n'$
```

Flipping: We define n different flipping transformations. The t^{th}-flip transformation changes the t^{th} bit of the sequences. In other words, if the t^{th}bit is 0 it is changed to 1 or vice versa. Formally the t^{th}-flip is defined as follows:

$$a_i^{'} = \begin{cases} a_i \oplus 1\, if\ i = t \\ a_i\, otherwise \end{cases}$$

3. Mutual correlations of randomness tests

In this section we give the correlation results of statistical randomness tests included in the NIST [8] test suite. As serial and cumulative sum tests have two different versions, we consider them as separate tests and we compare 16 statistical randomness tests (instead of 14 statistical randomness tests included in the NIST test suite). First, in order to investigate the tests properly, we divide the tests into two parts: tests that can be applied to short sequences and tests that can be applied to long sequences (length of such sequence is greater than 2^{20}). Note that the tests that can be applied to short sequences can also be applied to long sequences. For the tests that can be applied to short sequences such as frequency test and runs test, the number of produced P-values is approximately 200,000. On the other hand, for the tests that can only be applied to long sequences, 200 P-values are calculated. The two groups of statistical tests are stated in Table 1.

In the first experiment we use 200,000 pseudo-random sequences of length 1024 and implement the first set of tests. For example, frequency test and cumulative sum tests are found highly correlated. Two serial tests are also found correlated. This experiment shows the relation between some statistical randomness tests included in the NIST test suite suitable for short sequences. Accordingly, the tests that are found correlated have similar results on the same sequences. In the suite, if necessary, block size is chosen as 128. More details about the test properties are given in Table 2.

Table 1. Two groups of tests.

Applicable for both long and short sequences	Applicable only for long sequences
Frequency test	Binary matrix rank test
Frequency test within a block	Linear complexity
Runs test	Nonoverlapping template matching
Test for the longest run of ones in a block	Overlapping template matching test
Serial^1 - Serial^2	Maurer's "universal statistical" test
Approximate entropy	Random excursions test
Cumulative sums (backward and forward)	Random excursions variant test

Table 2. Properties of statistical randomness tests that can be applied to short sequences.

Statistical randomness tests	Block length	Parameter
Frequency test	-	-
Frequency test within a block	128	-
Runs test	-	-
Test for the longest run of ones in a block	128	-
Serial^1 - Serial^2	-	16
Approximate entropy test	-	14
Cumulative sums (Cusum) test (backward and forward)	-	-

The $(i, j)^{th}$ entry of Table 3 and Figure 1 corresponds to correlation between the i^{th} and j^{th} statistical randomness test. Clearly, all the entries in the diagonal are 1.000.

Table 3. Mutual correlation of randomness tests that can be applied to short sequences.

Tests	Frequency	Block frequency	Runs	Longest run of ones	Serial^1	Serial^2	Appr. entropy	CusumF	CusumB
Frequency	1.000	0.281	0.006	0.280	0.002	0.000	0.198	0.767	0.768
Block frequency	0.281	1.000	0.001	0.107	0.005	0.001	0.082	0.465	0.466
Runs	0.006	0.001	1.000	0.079	0.004	–0.001	0.191	0.006	0.007
Longest run of ones	0.280	0.107	0.079	1.000	0.005	0.001	0.231	0.244	0.247
Serial^1	0.002	0.005	0.004	0.005	1.000	0.681	0.020	0.005	0.001
Serial^2	0.000	0.001	–0.001	0.001	0.681	1.000	0.002	0.000	–0.001
Appr. entropy	0.198	0.082	0.191	0.231	0.020	0.002	1.000	0.176	0.177
CusumF	0.767	0.465	0.006	0.244	0.005	0.000	0.176	1.000	0.723
CusumB	0.768	0.466	0.007	0.247	0.001	–0.001	0.177	0.723	1.000

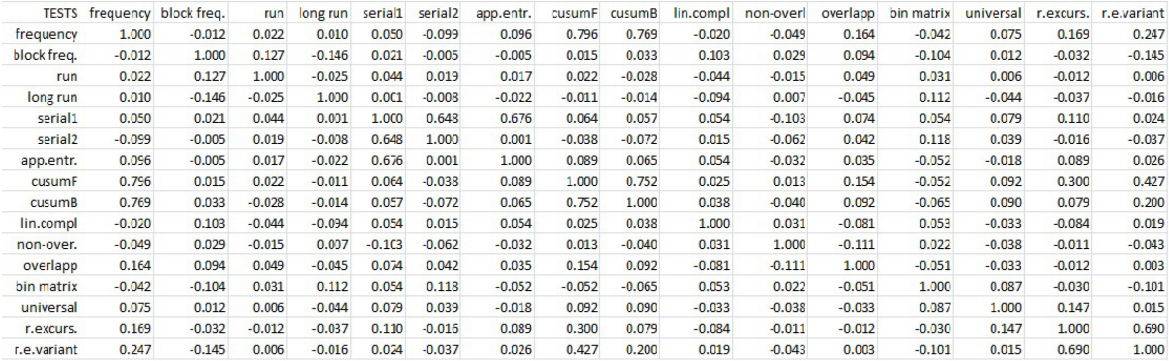
– Frequency test and cumulative sum tests (both backward and forward),

– Cumulative SUM TESTs,

– Serial^1 and serial^2 tests.

To guarantee the correlations existed, we choose coefficients greater than 0.5 because if the chosen number is small then the number of correlated tests increases and it will be difficult to attain reliable conclusions.

Similarly, if we choose a large threshold such as 0.9, the number of correlated tests decreases. The high correlation between these tests shows that our concern is right. Obtaining these results shows that test suite can be minimized by extracting one correlated test. For example, one cumulative sum test, backward or forward, or both can be discarded because not only are they correlated with each other but also they are correlated with frequency test. Moreover, the relations of these tests are found in previous works [11]. Moreover, finding two types of serial test are correlated shows that using one type of serial test is sufficient.



TESTS	frequency	block freq.	run	long run	serial1	serial2	app.entr.	cusumF	cusumB	lin.compl	non-overl	overlapp	bin matrix	universal	r.excurs.	r.e.variant
frequency	1.000	-0.012	0.022	0.010	0.050	-0.099	0.096	0.796	0.769	-0.020	-0.049	0.164	-0.042	0.075	0.169	0.247
block freq.	-0.012	1.000	0.127	-0.146	0.021	-0.005	-0.005	0.015	0.033	0.103	0.029	0.094	-0.104	0.012	-0.032	-0.145
run	0.022	0.127	1.000	-0.025	0.044	0.019	0.017	0.022	-0.028	-0.044	-0.015	0.049	0.031	0.006	-0.012	0.006
long run	0.010	-0.146	-0.025	1.000	0.001	-0.008	-0.022	-0.011	-0.014	-0.094	0.007	-0.045	0.112	-0.044	-0.037	-0.016
serial1	0.050	0.021	0.044	0.001	1.000	0.648	0.676	0.064	0.057	0.054	-0.103	0.074	0.054	0.079	0.110	0.024
serial2	-0.099	-0.005	0.019	-0.008	0.648	1.000	0.001	-0.038	-0.072	0.015	-0.062	0.042	0.118	0.039	-0.016	-0.037
app.entr.	0.096	-0.005	0.017	-0.022	0.676	0.001	1.000	0.089	0.065	0.054	-0.032	0.035	-0.052	-0.018	0.089	0.026
cusumF	0.796	0.015	0.022	-0.011	0.064	-0.038	0.089	1.000	0.752	0.025	0.013	0.154	-0.052	0.092	0.300	0.427
cusumB	0.769	0.033	-0.028	-0.014	0.057	-0.072	0.065	0.752	1.000	0.038	-0.040	0.092	-0.065	0.090	0.079	0.200
lin.compl	-0.020	0.103	-0.044	-0.094	0.054	0.015	0.054	0.025	0.038	1.000	0.031	-0.081	0.053	-0.033	-0.084	0.019
non-over.	-0.049	0.029	-0.015	0.007	-0.103	-0.062	-0.032	0.013	-0.040	0.031	1.000	-0.111	0.022	-0.038	-0.011	-0.043
overlapp	0.164	0.094	0.049	-0.045	0.074	0.042	0.035	0.154	0.092	-0.081	-0.111	1.000	-0.051	-0.033	-0.012	0.003
bin matrix	-0.042	-0.104	0.031	0.112	0.054	0.118	-0.052	-0.052	-0.065	0.053	0.022	-0.051	1.000	0.087	-0.030	-0.101
universal	0.075	0.012	0.006	-0.044	0.079	0.039	-0.018	0.092	0.090	-0.033	-0.038	-0.033	0.087	1.000	0.147	0.015
r.excurs.	0.169	-0.032	-0.012	-0.037	0.110	-0.016	0.089	0.300	0.079	-0.084	-0.011	-0.012	-0.030	0.147	1.000	0.690
r.e.variant	0.247	-0.145	0.006	-0.016	0.024	-0.037	0.026	0.427	0.200	0.019	-0.043	0.003	-0.101	0.015	0.690	1.000

Figure 1. Mutual correlation of randomness tests that can be applied to long sequences.

Algorithm 2 Experiment 1

```
Produce 200,000 random binary sequence of length 1024
Let test $s_i$,  $s_i = \{s_1, s_2, \ldots, s_9\}$ $i \leftarrow 1$, $j \leftarrow 1$ ($s_i$ implies the one of NIST test)
While $i \le 9$ do
    While $j \le 9$ do
        Produce 200,000 $p - values$ for test $s_i$ and $s_j$ say sequence $P_i$ and $P_j$
        Compute correlation $C_{i,j} = (P_i, P_j)$
        If $C_{i,j} \ge 0.5$ then
            $s_i$ and $s_j$ correlated
        Else
            $s_i$ and $s_j$ noncorrelated
        End if
        $j \leftarrow j + 1$
    End while
    $i \leftarrow i + 1$
End while
```

According to Table 3 we can summarize the highly correlated statistical tests as follows:

Furthermore, we do the same experiment on the second set of tests, which can be applied to long sequences, in addition to the first set of tests. For this experiment, we use 200 pseudo-random sequences of length 2^{20} and implement all the selected statistical tests. From Figure 1 we can say that the correlation coefficients are smaller than the ones in Table 3. Moreover, most of the tests are uncorrelated. The highly correlated statistical tests are as follows. For nonoverlapping template matching and overlapping template matching test, 9 is chosen as template length and the templates are ”000000001” and ”111111111”, respectively. For both random excursion and random excursion variant test, only one result is given. For both tests 1 is chosen as a parameter. More details about the test properties are given in Table 4.

Table 4. Properties of statistical randomness tests that can be applied to long sequences.

Statistical randomness tests block	Length	Parameter
Binary matrix rank test	-	32
Linear complexity	500	-
Nonoverlapping template matching test	-	000000001
Overlapping template matching test	-	111111111
Maurer's "universal statistical" test	-	7–10
Random excursions test	-	1
Random excursions variant test	-	1

– Serial^1 and approximate entropy tests,

– Random excursion and random excursion variant tests,

– Also the other correlation results for short sequences are valid.

Similar to tests that can be applied to short sequences, we choose tests as correlated if their correlation coefficient is greater than 0.5. The correlations found in the first experiment still exist. Although the results show that serial^1 and approximate entropy tests are correlated, serial^2 and approximate tests are not correlated. For this reason, if one of these tests is discarded from the suite, it is better to choose serial^1 since it is correlated with both serial^2 and approximate entropy tests. Moreover, we found that random excursion and random excursion variant tests are correlated. For these tests we choose a parameter as 1. After finding the correlation for other parameters, one can say that one of these tests can be discarded from the test suite. Although we look for only one type of these tests, we can say that for other parameters these test results can be correlated. Obtaining similar results in the first and second experiments shows that correlations exist and minimizing the test suite by discarding some of correlated tests is possible.

3.1. Experimental transformation results

In Section 2, we define 6 transformation methods. Using these methods, we generate transformed sequences and each sequence is tested. Accordingly P-values are generated. In order to find transformation results, test results of original sequences and transformed sequences are compared and correlation coefficients are generated for each transformation. After getting the correlations, we compared the row values in Figure 2 for each test. Finally, we get the comparison of the sensitivity of the tests.

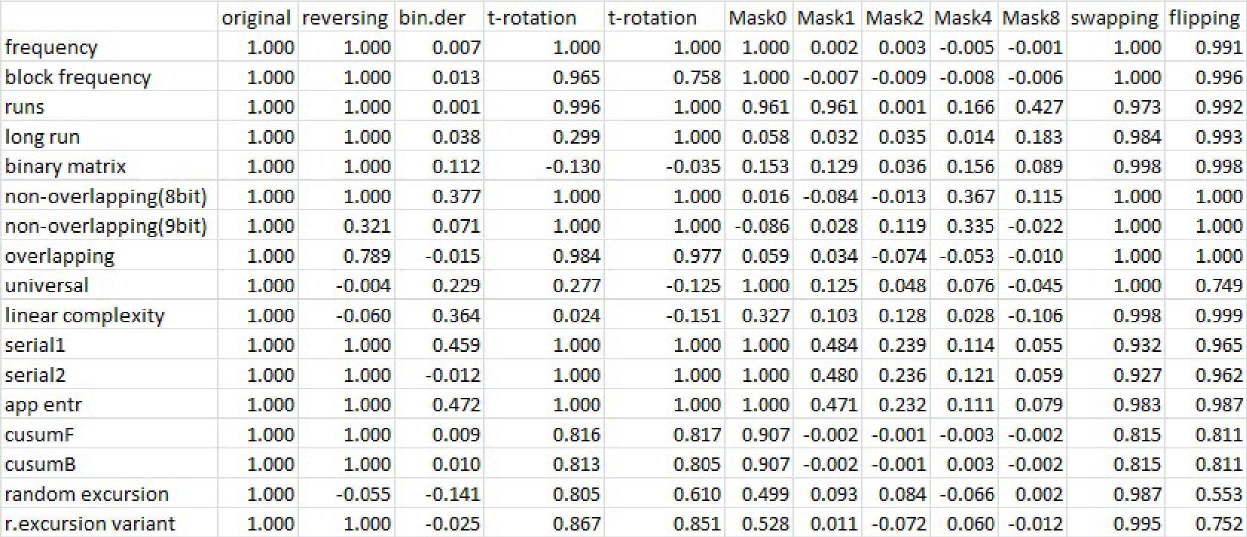
In order to get t-rotation result, we use 4 different $t$values, that is $t = 1, 2, 4, 8$, and for flipping results we use 5 different values, that is $i = 1, 2, 4, 8, 16$. In Figure 2, different $t$values are given because we get the same result for the other values of $t$and there exists only one $i$value for flipping because of the same reason.

The test properties are the same as in the correlations. The only difference is that template length is chosen as 8 and the template is "00000001" for one of the nonoverlapping template matching tests. From Figure 2 we obtain results that the test groups are related:

– Frequency, frequency test within a block, cumulative sum forward, and backward test.

– Runs, serial^1 and serial^2, approximate entropy test.

Moreover, we obtained results that the frequency test applied to sequences being transformed by binary derivative and the run test applied to the original sequence are correlated. Without increasing test number, one can obtain the same results by using transformed sequences.



	original	reversing	bin.der	t-rotation	t-rotation	Mask0	Mask1	Mask2	Mask4	Mask8	swapping	flipping
frequency	1.000	1.000	0.007	1.000	1.000	1.000	0.002	0.003	-0.005	-0.001	1.000	0.991
block frequency	1.000	1.000	0.013	0.965	0.758	1.000	-0.007	-0.009	-0.008	-0.006	1.000	0.996
runs	1.000	1.000	0.001	0.996	1.000	0.961	0.961	0.001	0.166	0.427	0.973	0.992
long run	1.000	1.000	0.038	0.299	1.000	0.058	0.032	0.035	0.014	0.183	0.984	0.993
binary matrix	1.000	1.000	0.112	-0.130	-0.035	0.153	0.129	0.036	0.156	0.089	0.998	0.998
non-overlapping(8bit)	1.000	1.000	0.377	1.000	1.000	0.016	-0.084	-0.013	0.367	0.115	1.000	1.000
non-overlapping(9bit)	1.000	0.321	0.071	1.000	1.000	-0.086	0.028	0.119	0.335	-0.022	1.000	1.000
overlapping	1.000	0.789	-0.015	0.984	0.977	0.059	0.034	-0.074	-0.053	-0.010	1.000	1.000
universal	1.000	-0.004	0.229	0.277	-0.125	1.000	0.125	0.048	0.076	-0.045	1.000	0.749
linear complexity	1.000	-0.060	0.364	0.024	-0.151	0.327	0.103	0.128	0.028	-0.106	0.998	0.999
serial1	1.000	1.000	0.459	1.000	1.000	1.000	0.484	0.239	0.114	0.055	0.932	0.965
serial2	1.000	1.000	-0.012	1.000	1.000	1.000	0.480	0.236	0.121	0.059	0.927	0.962
app entr	1.000	1.000	0.472	1.000	1.000	1.000	0.471	0.232	0.111	0.079	0.983	0.987
cusumF	1.000	1.000	0.009	0.816	0.817	0.907	-0.002	-0.001	-0.003	-0.002	0.815	0.811
cusumB	1.000	1.000	0.010	0.813	0.805	0.907	-0.002	-0.001	0.003	-0.002	0.815	0.811
random excursion	1.000	-0.055	-0.141	0.805	0.610	0.499	0.093	0.084	-0.066	0.002	0.987	0.553
r.excursion variant	1.000	1.000	-0.025	0.867	0.851	0.528	0.011	-0.072	0.060	-0.012	0.995	0.752

Figure 2. Correlation results of transformed sequences.

Algorithm 3 Experiment 2

```
Produce 200 random binary sequence of length $2^{20}$
Let test $s_i$, $s_i = \{s_1, s_2, \ldots, s_{16}\}$ , $i \leftarrow 1$, $j \leftarrow 1$
While $i \leq 16$ do
    While $j \leq 16$ do
        Produce 200 P-values for test $s_i$ and $s_j$ say sequence $P_i$ and $P_j$
        Compute correlation $C_{i,j} = (P_i, P_j)$
        If $C_{i,j} \geq 0.5$ then
            $s_i$ and $s_j$ correlated
        Else
            $s_i$ and $s_j$ noncorrelated
        End if
        $j \leftarrow j + 1$
    End while
    $i \leftarrow i + 1$
End while
```

We found mutual correlation of the frequency test and cumulative sum forward and backward test in Section 3. After finding that correlation of these tests still exists in the transformations, we can say that discarding one of these tests does not affect the test suite's result or decrease the reliability of the test suite.

In Section 3 we obtained that there exists a relation between serial^1, serial^2, and approximate entropy test. In this section we found these tests are correlated under transformations. Moreover, the run test is correlated with these tests. According to the results given above, there is a relation between the frequency and run tests. We can say that the run test can be discarded and transformed sequences can be used.

While using transformations we already know that some tests do not give any reaction for some transformations. In Figure 2 the cells that are equal to 1.000 show that the test does not give any reaction to that transformation. For instance, the frequency test does not give any reaction to reversing, t-rotation (not depending on value t), Mask-0, or swapping. On the other hand, although we expect that the linear complexity test shows a reaction to all transformations, it does not show a reaction to swapping or flipping. This is caused by the application method of the linear complexity test.

Furthermore, in this section we determined that it is not necessary to use some transformation methods in defined form. For example, flipping one bit does not change test results. Instead of this, changing different numbers of bit can be more useful.

4. Conclusions

In this paper, we suggest a new approach to find correlations of randomness tests and we apply them to 16 tests in the NIST test suite. Our main contribution is finding the correlations between two sets of tests by defining new transformation methods. We measure the direct correlation of each pair of test in the suite. In order to find direct correlation, we use Pearson's correlation coefficient. We produce P-values for each test and compare the P-values to find coefficients. In this method we observe that the sets "frequency test, cusum(F), cusum(B)", "serial^1, serial^2, and approximate entropy test", and "random excursion test, random excursion variant" test are correlated.

Furthermore, we define simple transformation methods to classify the tests according to their reactions to these transformations. Namely, for each test we find the correlations of the test results for the original sequences with the test results for the transformed sequence. Afterwards, we determine the tests that react similarly to the transformations, and we find that two sets of tests, "frequency, block frequency, cusum(F), cusum(B), random excursion test, random excursion variant test" and "runs test, serial^1, serial^2, and approximate entropy test", show similar reactions.

In future work, new transformation methods can be defined. The methods described in this paper can be applied to other statistical test suites. Moreover, new methods can be proposed to classify the statistical randomness tests.

References

[1] Turan MS. On statistical analysis of synchronous stream ciphers. PhD, Middle East Technical University, Ankara, Turkey, 2008.

[2] Golomb SW. Shift Register Sequences. Laguna Hills, CA, USA: Aegean Park Press, 1982.

[3] Knuth DE. The Art of Computer Programming, Volume 2 (3rd ed.): Seminumerical Algorithms. Boston, MA, USA: Addison-Wesley Longman Publishing Co, Inc., 1997.

[4] Caelli W. Crypt-X package documentation, technical report, Information Security Research, 1992.

[5] Marsaglia G. The Marsaglia random number CDROM including the DIEHARD battery of tests of randomness, http://www.stat.fsu.edu/pub/diehard/, 1995.

[6] Eddelbuettel D, Brown RG. DIEHARDER: an r interface to the dieharder suite of random number generator tests, 2007.

[7] L'Ecuyer P, Simard R. Testu01: a c library for empirical testing of random number generators. ACM T Math Softw 2007; 33.

[8] Bassham LE, Rukhin AL, Soto J, Nechvatal JR, Smid ME, Barker EB, Leigh SD, Levenson M, Vangel M, Banks DL et al. Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications, technical report, NIST, Gaithersburg, MD, United States, 2010.

[9] Soto J, Bassham L. Randomness testing of the advanced encryption standard finalist candidates, in NIST IR 6483, National Institute of Standards and Technology, 1999.

[10] Turan MS, Doğanaksoy A, Boztaş S. On independence and sensitivity of statistical randomness tests. In International Conference on Sequences and Their Applications (SETA), Lecture Notes in Computer Science. Springer, 2008.

[11] Doğanaksoy A, Ege B, Mus K. Extended results for independence and sensitivity of NIST randomness tests, in Information Security and Cryptography Conference, ISC Turkey, 2008.

[12] Fan L, Chen H, Gao S. A general method to evaluate the correlation of randomness tests. In: Kim Y, Lee H, Perrig A, editors. Information Security Applications, Lecture Notes in Computer Science, Springer International Publishing, 2014. pp. 52-62.

[13] Hellekalek P, Wegenkittl S. Empirical evidence concerning AES. ACM T Model Comput Simul 2003; 13: 322-333.

[14] Moore DS. The Basic Practice of Statistics with Cdrom. 2nd edition, New York, NY, USA: W. H. Freeman & Co., 1999.

[15] Daemen J, Rijmen V. The Design of Rijndael: AES – The Advanced Encryption Standard. Berlin, Germany: Springer, 2002.