Using Chernoff bound to statistical tests independence checking

Lyudmila Kovalchuk[1,†], Mariia Rodinko[2,*,†], Roman Oliynykov[2,†] and Tatiana Klymenko[1,†]

[1] *G.E. Pukhov Institute for Modelling in Energy Engineering, General Naumov Str. 15, Kyiv, 03164, Ukraine*

[2] *V. N. Karazin Kharkiv National University, Svobody Sq. 4, Kharkiv, 61022, Ukraine*

Abstract

In our work, we proposed, a strictly justified, method for testing independence verification and created a corresponding algorithm. We applied this algorithm to different test suits and obtained rather expected results, which indirectly confirms the correctness of our method. As we mentioned above, the proposed methods have several advantages: require fewer sequences, than the method based on approximation with normal distribution, may verify not only pairwise independence, in the most important cases, gives a more precise critical region, than the method based on Chebyshev inequality.

Using this method, we show that statistical tests from the widely used suits are independent. In this connection, it is interesting to note that the older version of NIST tests (published in 1999), which consists of more tests than the recent version, turned out to be dependent. There are no explanations from the authors of the updated version, to why they removed some tests, but our very method explained this fairly.

Keywords

pseudorandom number generators, statistical tests, NIST STS, cryptology

1. Introduction

A necessary (and most important) condition for solving problems in the field of information security is the use of sets of statistical tests developed separately for each type of problem. At the same time, it should be noted that although this topic is actively discussed in the world scientific literature (numerous references to scientific works of recent years will be given below), there is no complete and satisfactory answer to the questions. In addition, as Ukraine's experience in purchasing some strategically important products from other countries has shown, it is better to have your own developments and products in strategically important areas, rather than depending on the decisions of partners about their deliveries. Thus, the importance of ensuring the development and application of domestic systems and means of cryptographic protection of information for the cyber security of state information resources and objects of critical information infrastructure is especially emphasized in the Cyber Security Strategy of Ukraine (2021-2025) project [1]. Considering the above, it can be immediately stated that the topic of statistical methods of monitoring the operation of crypto primitives is and will always remain relevant.

Only in the last few years, many serious scientific works have been published on this topic. The best review and analysis of such works can be found in the recently published [2], which is a serious review and comparative work. It provides an overview of the main sets of statistical tests

Information Technology and Implementation (IT&I-2024), November 20-21, 2024, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

lusi.kovalchuk@gmail.com (L. Kovalchuk); m.rodinko@gmail.com (M. Rodinko); roliynykov@gmail.com (R. Oliynykov); klimenko-t@ukr.net (T. Klymenko).

0000-0003-2874-7950 (L. Kovalchuk); 0000-0003-4692-9811 (M. Rodinko); 0000-0002-3494-0493 (R. Oliynykov).

that are currently used for the analysis of crypto primitives and the generation of key data, performs a comparative analysis of them, and provides recommendations for their application. From this work, as well as similar works [3-6] we can draw the following conclusions.

1. Currently, for various tasks in the field of cryptology, only three main sets of statistical tests are used in the world, which have practically not changed over the past 10-20 years: NIST STS SP 800-22 (developed in the 2000s, the latest modification [7]); Diehard test set [8] and its minor and few modifications (e.g., [9]); a set of 5 easy-to-implement entropy tests ENT [10] and its modern modification ENT-string [11].
2. One of the main issues on the way to optimizing the process of verifying the cryptographic qualities of a crypto primitive is the optimization of the set of statistical tests itself, in particular, increasing its speed [4,5]. As shown by the work [5], the most effective way to optimize an arbitrary set is to eliminate the so-called "redundant" tests from it (this was the modification [7], where the redundant test was removed). The best way to solve such questions is to use the previously introduced notion of independence of statistical tests [12,13], which has proven itself well in practical applications. The approach proposed in our works to the issue of independence of tests is significantly more general than in the work [5], because, unlike this work, analyses the independence of tests in the aggregate, and not pairwise, as in [5].
3. Another important issue is the choice of a set of tests that is best suited for a specific task. So, for the testing of RNG/PRNG during admission to operation and first implementation, the largest and most "demanding" set is required; for quality control of key data – smaller, but one that prevents directed sorting of keys; for constant control of the allowed RNG/PRNG – significantly smaller than during admission. These questions are raised in [5-6] with an indication of their importance, but a concrete answer to them is not provided.
4. An important direction, which is not sufficiently covered in the scientific literature, is the use of statistical tests to check the independence of the sequence of internal states of a crypto primitive and its output sequences. Such a correlation significantly reduces the property of unpredictability of the original sequence and makes the algorithm vulnerable to statistical attacks. The method of conducting such a correlation analysis was proposed by us for the first time in [14], it can be expanded and generalized for wider use.

From all of the above, it is clear that, on the one hand, the issue of creating an effective (from the point of view of quality and speed) set of statistical tests for evaluating the cryptographic qualities of RNG/PRNG and their individual sequences is very relevant and is widely studied in modern scientific publications of a high scientific level; on the other hand, there are still many unanswered questions; one of these issues is the verification of the independence of statistical tests and the recognition of "redundant" tests that increase the time of testing but practically do not affect its results.

Intuition. For a better understanding of the problem statement, consider the following example. Let a certain package of statistical tests be used to check the cryptographic qualities of RNG/PRNG, well, for example, the NIST package consisting of 15 tests (not including subtests). Let’s denote this package P1. Practically all tests from this set are rather laborious and time-consuming. Suppose that, while studying the results of testing a large set of sequences, we notice that one of the tests never makes independent decisions. Formally, this means the following: if we remove this test from the set, the set of sequences that passed all tests will not change. That is, if we create a new package, P2, by removing such a redundant test, then the set of sequences that pass all tests from set P1 coincides with the set of sequences that pass all tests from set P2. That is, by removing the test, we did not deteriorate the quality of testing, but significantly reduced its time.

Generalizing this example, we can consider a situation where the test "almost" does not make independent decisions. Or if some subset of tests from that set has tests that are redundant to that subset.

It is impossible to "manually" go through all possible subsets and try to remove unnecessary tests from them. And here the methods of statistical analysis come to the rescue. Because all the situations described above are partial cases of what can be called the dependence of tests in the aggregate, and the tools of mathematical statistics can be used to detect such dependence.

Our impact. The results obtained in this work improve the methods proposed in the works [12,13]. For example, in the work [12], a method for checking the independence of statistical tests was proposed, which uses the asymptotic approximation of the probability distribution of a certain sum of random variables by the standard normal distribution. The disadvantage of this method is that there is no estimate of the rate of convergence of the distribution to normal, except for the Barry-Essen formula [15], which is considered a rather rough estimate. Therefore, to guarantee the use of such an approximation, it is necessary to take a very large number of sequences – about 100,000, which requires a very long testing time and a large computing resource. And if, instead of approximation, Chebyshev's inequality is used, as done in the work [13], then the critical area is significantly narrowed, since this inequality is a rather rough estimate. Moreover, in some cases this method can even give trivial estimates and become unusable. In this paper, we propose another method using Chernov's inequality. It does not require as many sequences as the first named, and in some cases gives a more accurate critical region than the second. By comparing these two methods (based on the Chebyshev and Chernov inequalities), we will analyse in which cases which method gives more accurate estimates.

2. Materials and Methods

In what follows, we will use the terms Random Number Generator (RNG) and Pseudorandom Number Generator (PRNG) for such types of number generators, which use some physical source during their work (for RNG) and use only random seed and then works deterministically. Often, we will formulate statements of definitions, which may be applied to both these types. In such cases, we will use the abbreviation (P)RNG or the term “generator”, without detailing its type.

Let us have some set T of statistical tests, $T = \{T_1, \dots, T_m\}$, and some set of sequences, $X = \{X^{(j)}\}_{j=1}^{n}$, where $X^{(j)} = \left\{x_1^{(j)}, \dots, x_l^{(j)}\right\}, j = \overline{1,n}$, is a binary sequence with the lengths l fit for this test suit, obtained from some (P)RNG *G*.

For a test T_i and some sequence $X^{(j)}$, taken from the corresponding suits, we define the event $\xi_i^{(j)} = I\{sequance\ X^{(j)}\ passed\ the\ test\ T_i\}$. For some fixed T_i, we can consider the sequences $\xi_i^{(j)}$ as the realization of some random variable (RV) ξ_i, which describes the behavior of the test T_i on the sequence obtained from the generator *G*.

Definition 1. Tests A and B are called independent (in relation to some fixed (P)RNG *G*), if ξ_A, ξ_B are statistically independent.

In other words, the independence of the test means that the result of the application of the test A for a sequence gives us no information about the result of the test B.

Note that, according to Definition 1, the conclusion of tests independence may be different for different generators. In what follows, we also assume that the (P)RNG that we use is *perfect*, i.e. indistinguishable from the *true random generator.* There exists a lot of such generators, for example, BBS [16], or standardized generators, described in [17] and [18]. The case of the creation of tests suit, which are independent w.r.t. perfect generators, is of the most interest because only perfect generators are used in cryptographic applications. Two perfect generators are undistinguished, so the tests which are independent w.r.t one of such generators are independent w.r.t. to any other.

Similarly, we can define a mutual tests independence.

Definition 2. Tests from suite $T = \{T_1, \ldots, T_m\}$ are called independent (w.r.t. some fixed generator *G*) if the corresponding RVs are mutually independent.

Let m be the number of tests in a suite, $\alpha_i, i = \overline{1,m}$ be the significance levels of the relevant test; n be the number of sequences we use to check the tests' independence; H_0 be a hypothesis that all tests from the suit are mutually independent; β be the probability to reject H_0 under the condition that it is correct. The alternative hypothesis is complicated and may be formulated as "tests from the suit are not mutually independent".

In what follows, we will use Chernoff bound to find the edges of critical region. There exist a lot of different versions of Chernoff inequality, among which we choose the one in the form given in Corollary 5 of [19].

Chernoff inequality [19].

Let $X_1, \ldots, X_m$ be independent random variables, which take binary values. Define

$X = \sum_{i=1}^{n} X_i$ and set $EX = \mu$.

Then for arbitrary $\delta \in (0,1)$ the next inequality holds:

$$P\left(\left|X-\mu\right| \geq \delta \cdot \mu\right) \leq 2 \cdot e^{-\frac{\delta^2 \cdot \mu}{3}}.$$

The proof of the next Proposition is based on Chernoff inequality.

Proposition 1.

Let statistical tests $T_1, \ldots, T_m$ are independent. Define ξ the RV, equal to the number of sequences from the set $\{X^{(j)}\}_{j=1}^{n}$, which passed all the tests, for some preset significance level α (the same for all tests). Then, for arbitrary $\beta \in (0,1)$, the next equality holds:

$$P\left(\left|\zeta-\mu\right| > \delta_\beta \cdot \mu\right) \leq \beta,$$

where $\delta_\beta = \sqrt{\frac{3}{\mu} \cdot ln \frac{2}{\beta}}$ and $\mu = n \cdot (1-\alpha)^m$.

Proof.

Introduce RVs

$$\xi_i^{(j)} = \begin{cases} 1, if\ the\ j-th\ sequence\ passes\ T_i; \\ 0, \quad else. \end{cases}$$

Next, define RV

$$\xi_i^{(j)} = \begin{cases} 1, if\ the\ j-th\ sequence\ passes\ all\ tests; \\ 0, \quad else. \end{cases}$$

Note that $\xi^{(j)} \in \{0,1\}$. Using this fact and independence of RVs $\xi_i^{(j)}$, we get

$$E\xi^{(j)} = \prod_{i=1}^{m} E\xi_i^{(j)} = (1-\alpha)^m,$$

$$Var\xi^{(j)} = (1-\alpha)^m \cdot \left(1-(1-\alpha)^m\right).$$

Finally, define the RV

$$\xi = \sum_{j=1}^{n} \xi^{(j)},$$

equal to the number of sequences passed all tests.

Note that $\mu = E\xi = n \cdot (1 - \alpha)^m$ and $Var\xi = n \cdot (1 - \alpha)^m \cdot (1 - n \cdot (1 - \alpha)^m)$.

Then apply Chernoff inequality to RV ξ and define δ in a such way that the right part of the equality be equal to A; obtain the inequality

$$P\left(\xi \notin \left[\mu - \delta_\beta \cdot \mu;\ \mu + \delta_\beta \cdot \mu\right]\right) \le \beta,$$

for $\delta_\beta = \sqrt{\frac{3}{\mu} \cdot ln\frac{2}{\beta}}$ and $\mu = n \cdot (1 - \alpha)^m$.

The Proposition is proved.

Based on Proposal 1, we can create the next algorithm for tests mutually independence.

Algorithm 1. Tests' independence verification based on the sample correlation matrix Chernoff inequality.

Input: number of tests m;

number of tests n;

set of tests $T = \{T_1, \dots, T_m\}$;

set of sequences $\{X^{(j)}\}_{j=1}^n$;

significance level α for testing sequences;

significance level β for verifying hypothesis H_0.

Step 1. Calculate and $\mu = n \cdot (1 - \alpha)^m$ and $\delta_\beta = \sqrt{\frac{3}{\mu} \cdot ln\frac{2}{\beta}}$.

Step 2. Calculate $C = \delta_\beta \cdot \mu$.

Step 3. Applying tests from the suit to input sequences, find the number k of sequences, which passed all tests.

Step 4. Calculate credential interval as

$$(I_1, I_2) = (\mu - C, \mu + C).$$

Step 5. If $k \in (I_1, I_2)$, then H_0 is accepted, otherwise it is rejected.

Output: "1" for accept, "0" for reject.

Example 1. Verification tests independence for NIST using PRNG defined in [20] and in Appendix A in DSTU 9041:2020 [18].

The input data were chosen as:

- the number of the sequences is $n = 300$;
- the significance level (for each test) is $\alpha = 0.01$;
- the number of tests in the suit is $m = 41$ (counting all subtests);
- the significance level for hypothesis H_0 verification is $\beta = 0.0001$.

For this data, according to the Algorithm 1, we calculate the credential interval:

$$(I_1, I_2) = (121.9,\ 275.5).$$

The number of sequences, which passed all tests, is 239. We can conclude that the tests from the suit are mutually independent.

If we reduce the critical region, setting $\beta = 0.01$, we get the credential interval

$$(I_1, I_2) = (143,\ 255).$$

It is essentially smaller than the previous one, but even for such a significance level tests may be considered mutually independent.

Example 2. "6 tests"

Verification tests independence for the set of 6 simple tests described in [13]. These tests are:
- frequency monobit text;
- frequency bigram test;
- number of series test;
- the maximal series length test;
- the sum of places of symbols test;
- inverse test.

The input data were chosen as:

- the number of the sequences is n = 300;
- the significance levels (the same level for all tests) are: α = 0.001; α = 0.005; α = 0.01; α = 0.05;
- the number of tests in the suit is m = 6 (counting all subtests);
- the significance level for hypothesis H_0 verification is β = 0.01.

For this data, according to the Algorithm 1, we obtained the next results.

1. For α = 0.001.

In this case $\mu = 298.2$ and $\delta_\beta \cdot \mu = 68.9$, so the credential interval is

$$\left(I_1, I_2\right) = \left(229.3,\ 300\right).$$

The number of sequences, which passed all tests, is 300. We can conclude that the tests from the suit are mutually independent.

2. For α = 0.005.

In this case $\mu = 291.1$ and $\delta_\beta \cdot \mu = 68$, so the credential interval is

$$\left(I_1, I_2\right) = \left(223.1,\ 300\right).$$

The number of sequences, which passed all tests, is 298. We can conclude that the tests from the suit are mutually independent.

3. For α = 0.01.

In this case $\mu = 282.4$ and $\delta_\beta \cdot \mu = 67$, so the credential interval is

$$\left(I_1, I_2\right) = \left(215.4,\ 300\right).$$

The number of sequences, which passed all tests, is 295. We can conclude that the tests from the suit are mutually independent.

4. For α = 0.05.

In this case $\mu = 221$ and $\delta_\beta \cdot \mu = 59.3$, so the credential interval is

$$\left(I_1, I_2\right) = \left(161.7,\ 280\right).$$

The number of sequences, which passed all tests, is 249. We can conclude that the tests from the suit are mutually independent.

As we see, tests may be considered as independent even for relatively large value of β. Indeed, the number of tests is relatively small, and in such cases tests usually are independent. So in these examples we obtained expected results, which indirectly confirms the correctness of the method and corresponding algorithm.

Now we are going to show that the proposed method of test independence verification may give tighter credential intervals, than a similar method based on Chebyshev inequality [13].

Proposition 2.

In our designations, if $\beta \leq 0.05$ and the values α and m are such that $(1-\alpha)^m < 0.442$, then the critical region, obtained using Chernoff inequality, is larger, than using Chebyshev one.

Proof.

According to Proposition 1, the hypothesis H_0 is accepted if

$$k \in (\mu - C_1,\ \mu + C_1),$$

where $C_1 = \delta \cdot \mu$, $\mu = n \cdot (1-\alpha)^m$, and $\delta_\beta = \sqrt{\frac{3}{\mu} \cdot ln \frac{2}{\beta}}$.

We may rewrite C_1 as

$$C_1 = \sqrt{\frac{3}{\mu} \cdot \ln \frac{2}{\beta}} \cdot \mu = \sqrt{3 \cdot \mu \cdot \ln \frac{2}{\beta}}.$$

If we use Chebyshev inequality instead of Chernoff inequality, we get the other credential interval:

$$k \in (\mu - C_2,\ \mu + C_2),$$

where (using (1))

$$C_2 = \sqrt{\frac{Var\xi}{\beta}} = \sqrt{\frac{\mu \cdot \left(1 - \frac{\mu}{n}\right)}{\beta}} = \sqrt{\frac{n \cdot (1-\alpha)^m \cdot \left(1-(1-\alpha)^m\right)}{\beta}}.$$

In these designations to prove that critical region with C_1 is larger than with C_2 is the same as to prove that $C_1 < C_2$, or that the next inequality holds:

$$\sqrt{3 \cdot \ln \frac{2}{\beta}} < \sqrt{\frac{\left(1-(1-\alpha)^m\right)}{\beta}}.$$

First, note that for $x \geq e$ the function $\frac{\ln x}{x}$ decreases if x increases. Then, for $e \leq a \leq x$ (for some $a \in R$) we have $\frac{\ln x}{x} \leq \frac{\ln a}{a}$.

In our conditions, $\beta \leq 0.05$, then for $x = \frac{2}{\beta} \geq 40$ we have:

$$\frac{\ln \frac{2}{\beta}}{\frac{2}{\beta}} \leq \frac{\ln 40}{40} < 0.093$$

and

$$3 \cdot \ln \frac{2}{\beta} < 3 \cdot 0.093 \cdot \frac{2}{\beta} = \frac{0.558}{\beta}.$$

On the other hand,

$$\frac{\left(1-(1-\alpha)^m\right)}{\beta} \geq \frac{1-(1-\alpha)}{\beta} = \frac{\alpha}{\beta} > 0.558 \cdot \beta,$$

according to the proposition assumption, and the Proposition is proved.

Example 3: the conditions of the Proposition 2 hold in such cases:

- $\beta \leq 0.05,\ \alpha = 0.05, m = 16$;
- $\beta \leq 0.001, \alpha \geq 0.01, m \geq 4$.

As common recommendations, we may say that the method based on Chernoff inequality works better for the cases when we have a small value of $\beta$and/or relatively large (w.r.t. β) α and/or large number m of tests. Note that in the overwhelming majority of applications, the value of β is chosen as 0.001 or even smaller, which is just the case for applying the proposed method. But in case when the number of tests is relatively small (less than 10, for example), and at the same time the value of β is not large than 0.01, the method which uses Chebyshev inequality is more preferable.

Conclusions

Tests independence is a very important and useful property. First, by avoiding using redundant tests, we may significantly reduce testing time, so may create key data for users more efficiently. Secondly, if the tests being applied are independent, we may, with some insignificant error, predict, what proportion of sequences will be rejected, and, therefore, understand how many sequences we need to generate to have the required volume of key data. For example, if we need to have 1000 key sequences, and know that the tests which are used are independent, then, if the significance level is α, the proportion of rejected sequences is, on average, $r = 1 - (1 - \alpha)^m$, where m is the number of tests. Then to get k sequences for key data, we need to generate about $\frac{k}{(1-\alpha)^m}$ sequences.

Note that in the NIST document [7] one can find some considerations about the importance of tests independence and even some kind of verification is given. But this verification can't be considered as sounded: it consists of calculating P-values for each pair test/sequence, then creating a matrix of these P-values (one row corresponds to one test) and checking the linear independence of the rows. This approach has no justification, and it seems very unlikely that for dependent tests such rows will be linearly dependent.

In our work, we proposed, a strictly justified, method for testing independence verification and created a corresponding algorithm. We applied this algorithm to different test suits and obtained rather expected results, which indirectly confirms the correctness of our method. As we mentioned above, the proposed methods have several advantages:

- require fewer sequences, than the method based on approximation with normal distribution [12];
- may verify not only pairwise independence, as methods, proposed in [21] and based on some ideas from [22];
- in the most important cases, gives a more precise critical region, than the method based on Chebyshev inequality [13].

Using this method, we show that statistical tests from the widely used suits are independent. In this connection, it is interesting to note that the older version of NIST tests (published in 1999), which consists of more tests than the recent version, turned out to be dependent. There are no explanations from the authors of the updated version, to why they removed some tests, but our very method explained this fairly.

Also note that we could not obtain the corresponding numerical examples for DIEHARD suit [9]. The matter is that the tests in this suit are created in other way, which gives no opportunity to get one result for one sequence. For such suit the method of results processing should be completely different.

As we mentioned in Definition 1, the notion of test independence may depend on the type of generator, more precisely – on the type and properties of probability distribution on its outputs.

Though we develop the methodic for most common use case, such as perfect generator testing, the similar approaches may be developed for other cases of output distribution. Generally speaking, for different types of output distributions we may obtain different sets of independent tests. But our experimental results, which we provided for generators with different types of output distributions (we did not include extend version of such investigation because of volume restriction) shows that two test suits, described above, turned out to be independent for several non-perfect types of generators, which output distribution was artificially biased from equiprobable. Of course, such experiments can't be considered as rigorous proofs of universality of definition of tests independence, but show that the test suit may be the same for different cases of generators.

The other interesting question, directly connected with the topic of presented research, is the next: when the test suit turned out to contain dependent tests, what tests should be considered as redundant and removed from the suit? Informally speaking, our answer is: such tests, that make the least amount of "independent" decisions. When we remove such tests, the number of sequences passing all tests will not change essentially, so the mutual first type error will remain the same too. Usually, the number of the tests in suit allows to check all tests decisions and to remove redundant tests using some kind of "brute force".

Acknowledgements

The results of this work were obtained within the project 2023.04/0020 Development of methods and layout of the "DEMETRA" ARM for constant and periodic control of the functioning of cryptographic applications using statistical methods.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] Cabinet of Ministers of Ukraine. (2021). Cybersecurity Strategy of Ukraine. Secure Cyberspace is the Key to the Successful Development of the Country. [Online]. Retrieved from https://zakon.rada.gov.ua/laws/show/447/2021#n12

[2] E. Almaraz Luengo, Statistical tests suites analysis methods. Cryptographic recommendations, Cryptologia 48(3) (2023) 219–251. doi:10.1080/01611194.2022.2155093.

[3] E. Almaraz Luengo, J. Román Villaizán, Cryptographically Secured Pseudo-Random Number Generators: Analysis and Testing with NIST Statistical Test Suite, Mathematics 11(23):4812 (2023). doi:10.3390/math11234812

[4] M. Sýs, Z. Říha, Faster Randomness Testing with the NIST Statistical Test Suite. In: R.S. Chakraborty, V. Matyas, P. Schaumont, (Eds.), Security, Privacy, and Applied Cryptography Engineering. SPACE 2014, volume 8804 of Lecture Notes in Computer Science, Springer, Cham, 2014, pp. 272–284. doi:10.1007/978-3-319-12060-7_18.

[5] E. A. Luengo, B.A. Olivares, L. J. G. Villalba, J. Hernandez-Castro, Further analysis of the statistical independence of the NIST SP 800-22 randomness tests, Applied Mathematics and Computation 459 128222 (2023). doi:10.1016/J.AMC.2023.128222

[6] E.A. Luengo, L.J.G. Villalba, Recommendations on Statistical Randomness Test Batteries for Cryptographic Purposes, ACM Comput. Surv. 54, 4, Article 80 (2021). pp.1-34. doi:10.1145/3447773

[7] L.E. Bassham III, A.L. Rukhin, J. Soto, J.R. Nechvatal, M.E. Smid, E.B. Barker and others, A statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST Special Publication 800-22, Revision 1a, (2010). URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf

[8] G. Marsaglia, The Marsaglia random number CDROM including the diehard battery of tests of randomness, (2008). http://www. stat. fsu. edu/pub/diehard/.

[9] R.G. Brown, D. Eddelbuettel, D. Bauer, Dieharder. Duke University Physics Department Durham NC 27708-0305 (2018).

[10] J. Walker, A pseudorandom number sequence test program, (2008). https://www.fourmilab.ch/random/

[11] E. Almaraz Luengo, B. Alaña Olivares, L.J. García Villalba, J. Hernandez-Castro, D. HurleySmith, StringENT test suite: ENT battery revisited for efficient P value computation, Journal of Cryptographic Engineering 13(2) (2023) 235-249. doi:10.1007/s13389-023-00313-5

[12] L. Kovalchuk, V. Bezditnyi, A statistical tests independence checking intended for cryptographic properties of RNG estimation, Ukrainian Information Security Research Journal 2 (29) (2006) 18-23.

[13] R. Kochana, L. Kovalchuk, O. Korchenko, N. Kuchynska, Statistical Tests Independence Verification Methods, Procedia Computer Science Volume 192 (2021). 2678-2688. doi:10.1016/j.procs.2021.09.038.

[14] L.V. Kovalchuk, I.V. Koriakov, A.N. Alekseychuk, Krip: High-Speed Hardware-Oriented Stream Cipher Based on a Non-Autonomous Nonlinear Shift Register, Cybernetics and Systems Analysis 59(1) (2023). 16-26. doi:10.1007/s10559-023-00538-6

[15] W. Feller. An introduction to probability theory and its applications, Vol. 2 (Vol. 81). John Wiley & Sons. (1991).

[16] V. Pareek, An overview of cryptographically secure pseudorandom number generators and BBS." International Journal of Computer Applications (IJCA)(0975–8887) (2014).

[17] Information technologies. Cryptographic protection of information. A digital signature based on elliptic curves. Formation and verification, DSTU 4145-2002.

[18] Information technologies. Cryptographic protection information. Short message encryption algorithm based on Edwards twisted elliptic curves, DSTU 9041:2020.

[19] M. Goemans, Chernoff bounds, and some applications, 18.310 lecture notes February 21, 2015. URL: https://math.mit.edu/~goemans/18310S15/chernoff-notes.pdf

[20] Information technologies. Cryptographic protection of information. Algorithm of symmetric block transformation, DSTU 7624:2014.

[21] Kovalchuk L., Davydenko A., Bespalov O. New statistical criteria for checking independence of bit random variables and sequences. In *Advances in Information-Control Systems and Technologies*, ODESA, 2024, p. 344-362. ISBN 978-617-7857-33-3.

[22] T.W. Anderson, An Introduction to Multivariate Statistical Analysis, 3rd Edition, Wiley, (2003) 380-410.