A new extended strategy of processing of statistical testing results

Lyudmila Kovalchuk[1,†], Hanna Nelasa[1,†] , Mariia Rodinko[2,*,†] and Oleksii Bespalov[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

This article proposes The Strategy for processing of testing which may be applied for arbitrary tests suit in which tests are independent and based on limit distributions. Testing parameters, used in Strategy, may be chosen depending on different factors, such as sphere of application of generator, our confidence of its quality, terms between planned testings, existing of other quality checkings. We also may change the number of tests in suit, reducing their number for regular everyday testing and increase it for testing before generator adoption. The Strategy summarizes testing results in one decision about quality of (P)RNG and possibility of its usage in cryptographic applications.

Keywords

pseudorandom number generators, statistical tests, NIST STS, cryptology

1. Introduction

Random/pseudorandom number generators (P)RNG are integral part of cryptology. The outputs of (P)RNG are used for creation of cryptosystem parameters, key materials, initialization vectors, auxiliary values for digital signatures. The necessary condition of cryptosystem security is high cryptographic properties of (P)RNG which outputs are used for this cryptosystem. In particular, the outputs of (P)RNG must be unpredictable, which involves the requirements on their statistical properties. To check these properties the suits of statistical tests are used, which consists of the different statistical tests, where each of them checks some specific property of sequence – equiprobable distribution of symbols, independence of elements, etc. Note that statistical testing doesn't cover all necessary investigation of cryptosystem security, this is just the initial phase in assessing if a (P)RNG is appropriate for a specific cryptographic use.

There are two fundamental types of generators for producing random sequences: random number generators (RNGs) and pseudorandom number generators (PRNGs). In case when it is not necessary to distinguish these two types, we will use abbreviation (P)RNG. Both of these generator types produce streams of binary values, and such stream that may be divided into blocks or transformed into random numbers.

As an example of true random bit sequence, we may consider the result of the flips of an unbiased "fair" coin with outcomes labelled "0" (for tail) and "1" (for head), with each flip having a probability of exactly ½ for each of outcomes. Each coin flip is independent of the others: the

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); annanelasa@gmail.com (H. Nelasa); m.rodinko@gmail.com (M. Rodinko); alexb5dh@gmail.com (O. Bespalov).
0000-0003-2874-7950 (L. Kovalchuk); 0000-0002-3708-0089 (H. Nelasa); 0000-0003-4692-9811 (M. Rodinko); 0000-0001-7126-6752 (O. Bespalov).

outcome of any previous coin flip does not influence future coin flips. Therefore, the value of the next element in the sequence remains unpredictable, no matter how many elements have already been generated. Thus, the unbiased "fair" coin is thus the perfect random bit generator, since the "0" and "1" values are randomly distributed.

In modern cryptography, generating sequences of 1,000,000 to 10,000,000 bits is often required, making the use of unbiased coins impractical for cryptographic purposes. Nonetheless, the hypothetical output of an ideal true random bit sequence generator serves as a benchmark for evaluating random and pseudorandom number generators.

The RNG employs a non-deterministic source (entropy source) combined with a processing function to generate randomness. This processing function is necessary to address any weaknesses in the entropy source that may lead to non-random numbers, such as extended sequences of zeros or ones.

The outputs of an RNG can be used directly or as input for a PRNG. If the output is used without further processing, it needs to satisfy strict randomness criteria, which is verified using corresponding statistical tests. Be aware that certain physical sources (e.g., date/time vectors) can be quite predictable. To address this issue, combining outputs from various types of sources can be used as inputs for an RNG. However, this process may be too time-consuming, making it impractical when a large amount of random bits is required.

To produce large quantities of random bits, PRNGs are more preferable. A PRNG uses one or more inputs and generates multiple "pseudorandom" numbers. Inputs to PRNGs are known as seeds. When unpredictability is essential, the seed must be both random and unpredictable. Therefore, a PRNG should typically acquire its seeds from the outputs of an RNG, meaning a PRNG relies on an RNG.

The outputs of a PRNG are usually deterministic functions of the seed, meaning all true randomness is limited to seed generation. The deterministic nature of this process is what gives rise to the term 'pseudorandom.'

To verify the cryptographic quality of (P)RNG, the suit of statistical tests should be applied to outputs of generator, which purpose, informally speaking, is to compare the output sequence to a truly random sequence. The characteristics of a random sequence can be expressed through probability. The expected results of statistical tests, when applied to a genuinely random sequence, are known in advance and can serve as a basis for comparison. There exists a huge number of different statistical tests and several test suites [1-17], but at the same time no specific finite tests suit is deemed "complete." The results of statistical testing should be interpreted carefully and cautiously to prevent drawing incorrect conclusions about a particular generator.

Typically, the testing procedure may be described as follows. We formulate some hypothesis, usually defined as H_0, that the sequence under testing is truly random. The alternative hypothesis H_1 is composite and may be formulated as "the sequence is not truly random". Then we introduce some value, called statistics, which may be calculated from the sequence elements and which, under the H_0 assumption, has some known probability distribution. After that we set some small value $\alpha \in (0,1)$ called "the significance level of the test" or "the 1st type error" and find critical region of criterion – such subset of the set of all possible values taken by statistics, which has probability α. Therefore, the probability that for true random sequence the obtained statistics gets to the critical region is very small (usually we choose $\alpha = 0.01$ or smaller). Then we calculate statistics for tested sequence and accept H_0, if statistics is outside the critical region, and reject it in opposite case. So the 1st type error is the probability to reject H_0 if it is true. The probability of the 2nd type error, to accept H_0 if it is not true, is impossible to calculate in case of composite hypothesis H_1.

There are huge number of articles, which develop new tests, or test suits, of investigate and analyse the results of testing. This paper also analyses some aspects of testing (P)RNGs, more precisely – the Strategy of processing of testing results (below – Strategy). Here we are not

considering the questions about the structure of test suit or about creating new statistical tests. Instead, we are concentrating on the question about how to process the results of testing and to obtaining the justified conclusion about the quality of (P)RNG.

We take the Strategy, proposed in NIST SP 800-22, Revision 1a [1], as the base for our investigation. The proposed Strategy has no explanation and justification, which cause the impossibility of analysis of its consistence. Because of this, the main purposes of our work are:

1. to analyse the Strategy and consider what rationale may be behind it;
2. to create corresponding justifications for each step of the Strategy;
3. to analyse possible incorrections and fix them;
4. to modify the Strategy, according to the obtained results, and extend it, if necessary, with additional steps.

The article is organized as follows. In the Section 1 we give relative work survey. In Section 2 we give brief overview of testing procedure and Strategy for the Statistical Analysis, proposed in NIST. Then we explain main issues of the Strategy. In Section 3 we proof several Propositions, needed for formulation and justification of new modified and extended Strategy. Then, in Section 4, we formulate this Strategy step-by-step, omitting such trivial steps as sequences creation and generation. Finally, we give the results of its application to certified (P)RNG DSTU 7624:2014. We conclude with summery of our results.

2. Analysing issues in NIST Strategy of processing of testing results

The revised version of NIST Test Suite (2010) [1] consists of the following 15 tests: the Frequency (Monobit) Test; frequency Test within a Block; the Runs Test; Tests for the Longest-Run-of-Ones in a Block; the Binary Matrix Rank Test; the Discrete Fourier Transform (Spectral) Test; the Non-overlapping Template Matching Test; the Overlapping Template Matching Test; Maurer's "Universal Statistical" Test; the Linear Complexity Test; the Serial Test; the Approximate Entropy Test; the Cumulative Sums (Cusums) Test; the Random Excursions Test; the Random Excursions Variant Test. These tests were developed to test the randomness of binary sequences produced by (P)RNG. They try to check different types of non-randomness that could exist in a sequence.

Note that the initial version of NIST tests, developed in 2000, contains one more test – Ziv-Lempel complexity test.

For interpretation of testing results, NIST uses Strategies for the Statistical Analysis (section 4.1), which consists of 5 steps. The 1^{st} step is (P)RNG Selection, the 2^{nd} is generating sufficient number of sequences of required length (not less than 300 sequences, but 1000 is more preferable), and the 3^{rd} step is testing all generated sequences with all tests from the suit. The Analysis itself consists of the 4^{th} step, where the uniform distribution of P-values is checked, The proposed Strategy have some issues, the most important are:

- absence of justification;
- absence of explanations how credential intervals were chosen;
- inconsistency of significance levels for required intervals;
- the Strategy doesn't take into account, that for some tests (like Maurer's test) the significance level may not coincide with the probability for statistics of truly random sequence to get into critical region;
- the Strategy analyses only the results of separate tests, without their mutual results.

In the next sections, we are going to give more details about these issues and to fix them, giving modified and extended Strategy with comprehensive justification.

3. Materials and Methods

In this section we give and prove several statements, which are basic for formulation and justification of improved version of testing Strategy for processing of testing results (below – NIST Strategy), proposed in NIST STS.

In what follows, we will use the expression "experiment with statistical test T (for given significance level α)", which means the procedure of testing some binary sequence with the test T. The outcome of the experiment is 0, if the hypothesis H_0 for this sequence is rejected with the test, and 1, if it is accepted. We will use the term "Truly RNG" (TRNG) for some ideal RNG, which generates independent equiprobably distributed bits, and "Perfect (P)RNG" (P(P)RNG) for such (P)RNG which is indistinguishable from Truly RNG.

We use abbreviation SND for "Standard Normal Distribution" and designation $\Phi(x)$ for its cumulative distribution function.

We will say "the test T is based on some limit distribution" if the test calculates statistics which approximately has some definite distribution, like SND of χ^2, and the probability to pass the test may be expressed using this distribution. Note that the majority of NIST tests are based on limit distributions, but not all of them. For example, the well-known and widely used Maurer's "Universal Statistical" Test [18], formally speaking, is not based on limit distribution, because its justification is partially empirical: the authors use Normal Distribution for approximation of sum of dependent RVs (more details may be found in test description). For such tests correspondence between significance level and critical region may be not precise.

The next Propositions are strongly proved only for tests which are based on limit distributions, because we will assume that the significance level is equal to the probability of sequence to get to the critical region. But it need be noted that NIST Strategy is implicitly based on the similar propositions and is applied for all tests without restrictions. It makes the Strategy partially empirical, and also may cause the situation when P(P)RNG may be rejected. In such cases, when significance level of test can't be calculate directly using corresponding limit distribution, it may be recommended to define the significance levels for different statistics values using some "standardized" PRNG, like BBS [2], as it was done for Maurer's test.

Proposition 1. Let us do n independent experiments with test T for sequences obtained from P(P)RNG for some preset significance level α. Define k the number of "1" among all outcomes. Then, for sufficiently large n and chosen $A \in (0,1)$, the next equality holds:

$$P\left(\frac{S_n}{n} \in \left[1-\alpha-C_A\cdot\sqrt{\frac{(1-\alpha)\cdot\alpha}{n}};\ 1-\alpha+C_A\cdot\sqrt{\frac{(1-\alpha)\cdot\alpha}{n}}\right]\right)=1-A,$$

where C_A is defined from the equality $\Phi(C_A) = 1 - \frac{A}{2}$.

Proof. Let $\xi = \{\xi_i\}_{i=1}^n$ be the sequence of independent equally distributed random variables (RVs), where $\xi_i \in \{0,1\}, i = \overline{1,n}$ are defined as

$$\xi_i = \begin{cases} 1, if\ outcome\ e\ of\ i-th\ experiment\ is\ 1; \\ 0, \quad otherwise. \end{cases} \tag{1}$$

Then, as experiments used the sequences from P(P)RNG, for all $i = \overline{1,n}$ we get:

$$E\xi_i = P(\xi_i = 1) = 1-\alpha, \qquad (\sigma)^2 = Var(\xi_i) = \alpha\cdot(1-\alpha), \tag{2}$$

for some $\alpha \in (0,1)$. Note that the second equality in (2) follows from the first one and from assumption that $\xi_i \in \{0,1\}$.

Define the new RV as

$$S_n = \sum_{i=1}^{n} \xi_i \quad (3)$$

Then, as $\xi_i, i = \overline{1,n}$, are independent and equally distributed,

$$ES_n = \sum_{i=1}^{n} E\xi_i = n \cdot (1-\alpha) \text{ and } Var(S_n) = \sum_{i=1}^{n} Var(\xi_i) = n \cdot \alpha \cdot (1-\alpha). \quad (4)$$

In these designations, the RV is the number of experiments with outcome 1 among all n experiments, and the value $\frac{S_n}{n}$ is the proportion of experiments with such outcome.

According to Central Limit Theorem [19], for sufficiently large n, the probability distribution of RV

$$\eta_n = \frac{S_n - n \cdot (1-\alpha)}{\sqrt{n \cdot (1-\alpha) \cdot \alpha}} = \frac{\frac{S_n}{n} - (1-\alpha)}{\sqrt{\frac{(1-\alpha) \cdot \alpha}{n}}}$$

may be approximated with SND as

$$P(|\eta_n| \geq x) = (1 - \Phi(x)) + \Phi(-x) = 2 - 2 \cdot \Phi(x).$$

For some quantile A define such C_A that $2 - 2 \cdot \Phi(C_A) = A$, or $\Phi(C_A) = 1 - \frac{A}{2}$.

Then $P(|\eta_n| \geq C_A) = A$, which may be rewritten as

$$P\left(\frac{S_n}{n} \in \left[1-\alpha - C_A \cdot \sqrt{\frac{(1-\alpha) \cdot \alpha}{n}};\ 1-\alpha + C_A \cdot \sqrt{\frac{(1-\alpha) \cdot \alpha}{n}} \right] \right) = 1 - A,$$

and the Theorem is proved.

Note that in [1] (section 4.2.1, Proportion of Sequences Passing a Test) the value C_A is chosen $C_A = 3$, which corresponds to $1 - \frac{A}{2} = 0.99865$, or $A = 0.0027$. It means that the probability that the proportion of sequences is outside the interval is about 0.0027.

Proposition 2. Let us have n outcomes of independent experiments with test T for sequences obtained from P(P)RNG for some preset significance level α. Consider RV $P_T = P(T, \alpha)$ which takes values which are equal to corresponding P-values, obtained in experiments. Then RV P_T is uniformly distributed on [0,1].

Proof. Let $F_T(x)$ be cumulative distribution function of test statistics U_T:

$$P(U_T \in (a, b)) = F_T(b) - F_T(a).$$

Then for arbitrary $(x, x+\delta) \subset [0,1]$:

$$P(P_T \in (x, x+\delta)) = P(U_T \in (F_T^{-1}(x+\delta), F_T^{-1}(x))) =$$
$$= F_T(F_T^{-1}(x+\delta)) - F_T(F_T^{-1}(x)) = x + \delta - x = \delta,$$

which means that P_T has uniform distribution.

To verify uniform distribution of P-values for each test, NIST Strategy proposes to use χ^2-criterion (more precisely – its modification with gamma-function) with significance level $A = 10^{-4}$. Because of this, it is unclear why the Strategy proposes much more higher significance level, A = 0.0027, for its previous step. To remove such unfairness, it`s better to use the same significance level, $A = 10^{-4}$, for both steps. In this case we get $C_A = 4$ instead of $C_A = 3$, and the corresponding interval for proportion of sequences passed the test will be

$$\left[1-\alpha - 4 \cdot \sqrt{\frac{(1-\alpha) \cdot \alpha}{n}};\ 1-\alpha + 4 \cdot \sqrt{\frac{(1-\alpha) \cdot \alpha}{n}} \right].$$

As we mentioned before, these two statements may be used to justify the NIST Strategy, described in section 4.2.1 [1], but only for such separate tests which are based on some limit distributions. Below we give one more statement, which allows to extend NIST Strategy in a such way, that take into account not only separate tests behaviour, but also the mutual behaviour of them. In what follows we will use the notation of tests independence, introduced in [11] and then developed in [13]. The strict definition of tests independence is rather complicated and is detailly described and may be found in [11]. Informally speaking, the tests from some sets are considered to be independent, if their decisions about acceptance/rejection of hypothesis are independent. It is the same as RVs, which reflect tests decisions, are mutually independent. Note that [1] also mentions tests independence, but don't give neither straight definition of the notion, no justified approach to verifying tests independence.

In what follows, we will use Chernoff inequality in the form given in Corollary 5 of [20].

Chernoff inequality. Let $X_1, \dots, X_n$ are independent RVs taking values in {0,1}. Define

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

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

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

Proposition 3. Let independent statistical tests $T_1, \dots, T_m$ were applied for testing of n sequences obtained from P(P)RNG for some preset significance level α (the same for each test). Define k the number of sequences which pass all the tests. Then, for sufficiently large n and chosen $A \in (0,1)$, the next equality holds:

$$P\left(k \in \left[\mu - \delta_A \cdot \mu;\ \mu + \delta_A \cdot \mu\right]\right) \geq 1 - A,$$

where $\delta_A = \sqrt{\frac{3}{\mu} \cdot ln\frac{2}{A}}$ 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

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

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

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

$$P\left(k \notin \left[\mu - \delta_A \cdot \mu;\ \mu - \delta_A \cdot \mu\right]\right) \leq A,$$

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

The Proposition is proved.

4. New Strategy for processing of testing results and its justification

Above we gave three statements which allow to justify partially NIST Strategy, define its weakness and incorrectness, and proposed to add some new step in the Strategy. Now we are going to formulate Algorithm which realizes the New Extended Strategy.

Input:

- the number n of the tested sequences ($n \geq 300$);
- the set of sequences $X^{(j)} = \left\{x_1^{(j)}, \ldots, x_l^{(j)}\right\}, j = \overline{1,n}$, of sufficient length l, obtained from investigated (P)RNG;
- the significance level α (for testing);
- the number m of tests in the suit;
- the significance level A (for analysing testing results).

Step 1 (Testing). Test all sequences; for each test $T_i, i = \overline{1,m}$, and each sequence $X^{(j)}, j = \overline{1,n}$, obtain corresponding P-value $P_i^{(j)}$.

Step 2 (Calculated quantiles and auxiliary values). Calculate the next values:

- the quantile C_A such that $\Phi(C_A) = 1 - \frac{A}{2}$;
- the quantile χ_A^2 such that $F(\chi_A^2) = 1 - A$, where $F(x)$ is cumulative distribution function of χ^2-distribution with 9 degrees of freedom;
- the edges of credential interval (for analyzing results of separate tests), corresponding to the significance level A:

$$I_1 = 1 - \alpha - C_A \cdot \sqrt{\frac{(1-\alpha)\cdot\alpha}{n}} \quad \text{and} \quad I_2 = \min\left\{1,\ 1 - \alpha + C_A \cdot \sqrt{\frac{(1-\alpha)\cdot\alpha}{n}}\right\};$$

- the values $\mu = n \cdot (1-\alpha)^m$ and $\delta_A = \sqrt{\frac{3}{\mu} \cdot ln\frac{2}{A}}$;
- the edges of credential interval (for analyzing results of testing with tests suit), corresponding to the significance level A: $V_1 = \mu - \delta_A \cdot \mu$ and $V_2 = \mu + \delta_A \cdot \mu$.

Step 3 (Checking uniform distribution of P-values for each separate test).

For each test $T_i, i = \overline{1,m}$, do the next sub-steps:

3.1. find the values F_k, $k = \overline{0,9}$, equal to the number of P-values $P_i^{(j)}$, $j = \overline{1,n}$, which belong to the interval $\left[\frac{k}{10}, \frac{k+1}{10}\right)$;

3.2. calculate χ^2-statistics as

$$\chi_i^2 = \sum_{k=0}^{9} \frac{\left(F_k - \frac{n}{10}\right)^2}{\frac{n}{10}}$$

3.3. if $\chi_i^2 \leq \chi_A^2$ then output "P-values obtained using test T_i are uniformly distributed"; else output "P-values obtained using test T_i are not uniformly distributed".

Step 4 (Checking proportion of sequence passing the test for each separate test).

For each test $T_i, i = \overline{1,m}$, do the next sub-steps:

4.1. calculate the value k_i equal to the number of sequences passing the test;

4.2. if $I_1 < \frac{k_i}{n} < I_2$ than output "proportion of sequences passed T_i lies inside the correct interval"; else output "proportion of sequences passed T_i lies outside the correct interval".

Step 5 (Checking proportion of sequence passing all tests).+

5.1. calculate the value k equal to the number of sequences passed all the tests;

5.2. if $V_1 < k < V_2$ than output "number of sequences passed all tests lies inside the correct interval"; else output "number of sequences passed all tests lies outside the correct interval".

If on each step the algorithm gave positive answers, then we may consider the corresponding (P)RNG as perfect.

Results of Strategy application.

We applied the Strategy to the set of sequences generated from the certified generator, described in Appendix A in DSTU 9041:2020 [21]. The input data were the next:

- the number of the tested sequences n = 300;
- the significance level (for testing) α = 0.01;
- the number of tests in the suit m = 41 (with all subtests);
- the significance level (for analysing testing results) A = 0.0001.

Now we give the step-by-step results of New Strategy application, according to Algorithm, given in Section 4.

Step 1 (Testing). After testing each of these 300 sequences using 41 tests from [1], with significance level α = 0.01 for each test, we obtain the matrix of 300x41 size.

Step 2 (Calculated quantiles and auxiliary values). For chosen significance level (for analysing testing results) A = 0.0001 we find, using Standard Normal distribution table, the corresponding quantile C_A such that $\Phi(C_A) = 1 - \frac{A}{2} = 1 - 0.00005 = 0.99995$, and obtain $C_A = 4$.

For chosen significance level (for analysing testing results) A = 0.0001 we find, using χ^2-distribution table, the corresponding quantile χ^2_A such that $F(\chi^2_A) = 1 - A = 1 - 0.0001 = 0.9999$, where $F(x)$ is cumulative distribution function of χ^2-distribution with 9 degrees of freedom (because the number of intervals, for which we calculate the number of P-values, was chosen as 10): $\chi^2_A = 33{,}7199484$.

Next, for chosen α = 0.01, A = 0.0001, given number of sequences n = 300 and obtained value $C_A = 4$ we calculate the critical region (outside the interval (I_1, I_2)) for the proportion of sequences which pass each test as

$$I_1 = 1 - \alpha - C_A \cdot \sqrt{\frac{(1-\alpha)\cdot\alpha}{n}} = 0.96 \quad \text{and} \quad I_2 = \min\left\{1,\ 1 - \alpha + C_A \cdot \sqrt{\frac{(1-\alpha)\cdot\alpha}{n}}\right\} = 1.$$

Then we calculate auxiliary values $\mu = n \cdot (1-\alpha)^m$ and $\delta_A = \sqrt{\frac{3}{\mu} \cdot ln\frac{2}{A}}$ and use them to calculate the critical region (outside the interval (V_1, V_2)) for the number of sequences which pass all the tests as $V_1 = \mu - \delta_A \cdot \mu = 121.9$ and $V_2 = \mu + \delta_A \cdot \mu = 275.5$.

Step 3 (Checking uniform distribution of P-values for each separate test).

For each test T_i, $i = \overline{1,m}$, we calculate the number of P-values, which lie in each of 10 intervals, and applied Pearson criterion for obtained values, to check uniformity of their distribution. For all tests, the corresponding statistics were not larger than 22.4, which is smaller than limit statistic $\chi^2_A = 33{,}7199484$. Then the distribution of P-values may be considered uniform (for each test), and the first requirement of Strategy is met.

Step 4 (Checking proportion of sequence passing the test for each separate test).

For each test T_i, $i = \overline{1,m}$, we calculate the value k_i equal to the number of sequences passing the test. The maximal value of k_i, obtained on this step, is equal to 5, which corresponds to the proportion 0.983, which lies inside the interval $(I_1, I_2) = (0.96, 1)$. So, the second requirement of Strategy is met.

Step 5 (Checking proportion of sequence passing all tests).

We calculate the value k which is equal to the number of sequences passed all the tests: k = 239. This value lies inside the interval $(V_1, V_2) = (121.9, 275.5)$, so the third requirement of Strategy is met. **We can conclude that the tested PRNG is perfect.**

Conclusions

The Strategy for processing of testing results, proposed in the article, is extended and fully justified modification of the Strategy proposed in [1]. It may be applied for arbitrary tests suit in which tests are independent and based on limit distributions. In other case we can't state that the conclusion about properties of generator, based on results of testing, is correct.

We may choose testing parameters, used in Strategy, depending on different factors, such as sphere of application of generator, our confidence of its quality, terms between planned testings, existing of other quality checkings. We also may change the number of tests in suit, reducing their number for regular everyday testing and increase it for testing before generator adoption. The Strategy summarizes testing results in one decision about quality of (P)RNG and possibility of its usage in cryptographic applications. But the perfectness of generator does not guarantee that all its sequences also are "perfect". For example, the probability of the long zero string is not zero. So even in case when we use perfect generator in cryptographic applications, we still should test each separate sequences before using it for key data creation.

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 authors have not employed any Generative AI tools.

References

[1] 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

[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] G. Marsaglia, The Marsaglia random number CDROM including the diehard battery of tests of randomness, (2008). http://www. stat. fsu. edu/pub/diehard/.
[8] R.G. Brown, D. Eddelbuettel, D. Bauer, Dieharder. Duke University Physics Department Durham NC 27708-0305 (2018).
[9] J. Walker, A pseudorandom number sequence test program, (2008). https://www.fourmilab.ch/random/
[10] E. Almaraz Luengo, B. Alaña Olivares, L.J. García Villalba, J. Hernandez-Castro, D. Hurley-Smith, 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
[11] L. Kovalchuk, V. Bezditnyi, Inspection of statistical tests independence intended for PRNG cryptographic qualities evaluation, Ukrainian Information Security Research Journal 2 (29) (2006) 18-23.
[12] 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.
[13] K. Bhattacharjee, S. Das, A search for good pseudo-random number generators: Survey and empirical studies, Computer Science Review, 45 (2022) 100471 doi:10.1016/j.cosrev.2022.100471
[14] S.H. AbdELHaleem, S.K. Abd-El-Hafiz, A.G. Radwan, Analysis and Guidelines for Different Designs of Pseudo Random Number Generators, in IEEE Access 12 (2024). 115697-115715. doi: 10.1109/ACCESS.2024.3445277.
[15] M. Sigit, To what extent are multiple pendulum systems viable in pseudo-random number generation?, arXiv preprint 2404.16860 (2024). URL:https://arxiv.org/pdf/2404.16860. doi:10.48550/arXiv.2404.16860
[16] R.B. Naik, U. Singh, A Review on Applications of Chaotic Maps in Pseudo-Random Number Generators and Encryption, Annals of Data Science 11(1) (2024). 25–50. doi:10.1007/s40745-021-00364-7
[17] 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
[18] U.M. Maurer. A universal statistical test for random bit generators. Cryptology (5) (1992) 89–105.
[19] W. Feller. An introduction to probability theory and its applications, Vol. 2 (Vol. 81). John Wiley & Sons. (1991).
[20] C. Grosu. Some applications of Chernoff bounds. Mathematical Modeling in Civil Engineering, (3) (2010).
[21] Information technologies. Cryptographic protection information. Short message encryption algorithm based on Edwards twisted elliptic curves, DSTU 9041:2020.