Tetiana OKHRIMENKO[1], Sergiy DOROZHYNSKYI[2], Bohdan GORBAKHA[3]

Scientific supervisor: Sergiy GNATYUK[4]

DOI: https://doi.org/10.53052/9788366249868.16

# ALGORYTM I NARZĘDZIA OPROGRAMOWANIA DO GENEROWANIA LICZB PSEUDOLOSOWYCH I OCENA LOSOWOŚCI

**Streszczenie:** W artykule zaproponowano podejście do generowania i oceny jakości trójskładnikowych liczb pseudolosowych. Takie podejście umożliwia generowanie i fundamentalną ocenę ogólnie przyjętych parametrów i wzorców statystycznych. Wyniki te pozwalają ocenić bezpieczeństwo kryptograficzne generatorów liczb pseudolosowych Trit i ich praktyczną zdolność do różnych zastosowań kryptograficznych.

**Słowa kluczowe:** cyberbezpieczeństwo, kryptografia, kryptografia kwantowa, trit, PRNG, losowość

# ALGORITHM AND SOFTWARE TOOLS FOR TERNARY PSEUDORANDOM NUMBERS GENERATION AND RANDOMNESS ASSESSMENT

**Abstract:** In this paper an approach to the generation and evaluation of ternary pseudo-random numbers quality was proposed. This approach makes it possible to generate and fundamentally evaluate the generally accepted statistical parameters and patterns. This results allow to evaluate the cryptographic security of trit pseudo-random numbers generators and their practical ability for various cryptographic applications.

**Keywords:** cybersecurity, cryptography, quantum cryptography, trit, PRNG, randomness

[1] National Aviation University, PhD, Associate Professor, Chief of the NAU Cybersecurity R&D Lab, taniazhm@gmail.com

[2] National Aviation University, PhD Student, Junior Researcher of the NAU Cybersecurity R&D Lab, dorozhun1706@gmail.com

[3] National Aviation University, Technical Support Specialist of the NAU Cybersecurity R&D Lab, jmorosr2@gmail.com

[4] National Aviation University, DSc, Professor, Scientific Advisor of the NAU Cybersecurity R&D Lab, s.gnatyuk@nau.edu.ua

## 1. Introduction

Recently, of great interest is quantum cryptography, which does not depend on the computational power of the violator, uses the specific unique properties of quantum particles and is based on the inviolability of the laws of quantum physics [1]. The main advantages of quantum cryptography methods are the ability to accurately identify the violator and ensure theoretical and informational (absolute) stability.
Some quantum cryptography protocols require the use of non-binary number systems, in particular ternary (trit) [2]. Such protocols require the generation of high-quality ternary pseudo-random numbers (PRN) [3].

## 2. Related papers analysis

The analysis revealed a sufficient number of existing PRN generators (PRNG) that can be used for various applications [4-6]. The ISO / IEC 18031 standard, which sets out the conceptual models, terminology and requirements related to the structural elements and properties of the systems used to generate random bits in cryptographic applications, defines two types of generators: *indeterminate* (a random bit generation mechanism that uses an entropy source to generate a random bit stream) and *determined* (a bit generation mechanism that uses deterministic mechanisms, such as cryptographic algorithms, on an entropy source to generate a random bit stream. Uses special inputs and, if necessary, some optional inputs that may be publicly available depending on their application) random bit generators.

According to the method of obtaining PRNG are divided into three fundamentally different classes:

- *tabular* – the main drawback is finite,
- *physical* (radioactive emission, physical noise generators, quantum generators, pulse sequence generators, etc.) – common and most significant shortcomings that complicate their application are limited performance, low stability of the main probabilistic characteristics, due to the instability of primary sources, drift parameters of converter circuits, power supplies and requires periodic statistical quality control; complexity of hardware implementation,
- *algorithmic* (PRNG, for example, the method of median squares, the method of median products, the method of mixing, the linear congruent method).

PRNG can also be divided into *crypto-resistant* and *not crypto-resistant*. Which in turn include

*1) crypto-resistant:*

- based on streaming ciphers (for example, Dragon-128, SEAL, RC4, RC5, RC6, Grain, Yamb, Phelix);
- based on block ciphers (for example, GOST 28147-89, AES, ANSI X9.17, DES);
- based on one-way functions (for ex. generators BBS, RSA, Dual_EC_DRBG (elliptic curves), GPSSD (linear codes) etc.)

*2) not crypto-resistant:*

- based on elementary recurrents (for example, linear congruent generator, polynomial congruent generator, additive Fibonacci generator, additive Fibonacci generator with delay, multiplicative Fibonacci generator with delay);
- based on operations in finite fields (for example, Galois generators, De Brain, Fibonacci, additive, Golman, compressor, etc.).

Among the most popular approaches to assessing the randomness of PRN are the following [7-9]:

- NIST Statistical Test Suit;
- Diehard tests by George Marsaglia;
- Dieharder: A Random Number Test Suite by Robert G. Brown;
- TestU01;
- Knuth tests and others.

## 3. Problem statement

However, all currently developed methods of PRN generation (PRNG) are focused on binary sequences, and therefore the development of a method of generating trit PRN is an urgent scientific task. In addition, assessing the quality of ternary PRN requires the development of appropriate approaches. Therefore, **the aim** of this work is to develop methods, algorithms and tools for generating and evaluating the quality of ternary PRN.

## 4. The main part of the study

### 4.1. Method and algorithm of formation of ternary PRN

A method for generating trit PRN is proposed $\xi$ with many vectors of internal states $V_p$ ($V_p = \{0, 1, 2\}^p$), a set of secret keys $V_n$, multiple initialization vectors $V_e$ and a plurality of source sequences $V_m$, where $p = 14 \cdot l$, $n = 4 \cdot l$, $e = p - n = 10 \cdot l$, $m = b \cdot n$, $l = d \cdot s$ and $b$, $d$, $s$ are natural numbers.

Based on the proposed method of generating trit PRN $\xi$ developed an algorithm TriGen (pseudocode of the algorithm see in Fig. 1). In the algorithm TriGen the following parameters are used $d = 4$, $s = 6$, $l = d \cdot s = 24$, $p = 14 \cdot l = 336$, $n = 4 \cdot l = 96$, $e = p - n = 10 \cdot l = 240$, $m = b \cdot n = 96 \cdot b$, $r = 4$ , $b$ is natural number.

In operation $Sbox(X)$ a nonlinear replacement of every six trits of the number $X$ is performed to the corresponding value of the substitution table. Only one substitution table is used, which is constructed by calculating the inverse element of the field $(X)^{-1} \in GF(3^6)$ with subsequent execution of the affine transformation over the field

$GF(3)$: $S(X)=M\cdot\left(X\right)^{-1}+V$, where $X,V\in GF(3^6)$, and $M$ is square non-degenerate matrix over the field $GF(3)$ size $6\times 6$. The final field GF($3^6$) is fixed by a ring of polynomials with operations modulo an irreducible polynomial $m(x)=x^6+x+2$.

The following matrix values were chosen to construct the proposed substitution table $M$ and vector $V$:

$$M=\begin{pmatrix}1&2&0&1&1&2\\2&1&2&0&1&1\\1&2&1&2&0&1\\1&0&2&1&2&0\\0&1&0&2&1&2\\2&0&1&1&2&1\end{pmatrix},\ V=\begin{pmatrix}0\\2\\2\\1\\0\\2\end{pmatrix}.$$

In operation $Mix\left(X\right)$ square non-degenerate matrix $M'$ over the field $GF(3)$ size $24\times 24$ tritium multiplied by $X$ (presented as a column vector) above the field $GF(3)$.

```
TriGen
Input: initialization vector $VI$, secret key $K$, $VI\in V_{240}$, $K\in V_{96}$, parameter $b$.
Output: source sequence $M=\left(M_1,...,M_b\right)$, $M\in V_{96b}$, $M_q\in V_{96}$, $q\in\overline{1,b}$.
1. $x_i=VI_i$, $y_j=VI_{6+j}$, $k_j=K_j$, $i\in\overline{1,6}$, $j\in\overline{1,4}$.
2. $For\ q=1;\ q\le b;\ q++\ do$
    2.1. $For\ j=0;\ j<4;\ j++\ do$
        2.1.1. $x_1=\left(Sbox\left(x_1+k_1\right)\oplus x_4\right)<<<k_4$; $x_2=\left(Sbox\left(x_2+k_2\right)+x_5\right)>>>k_3$;
               $x_3=Mix\left(\left(x_3+x_6\right)\oplus y_3\right)<<<x_1$;
        2.1.2. $k_1=Sbox\left(\left(Sbox\left(x_1\oplus k_1\right)+x_5\right)\oplus y_1\right)$;
               $k_2=Sbox\left(Mix\left(x_2+k_2+x_6\right)\oplus y_2\right)$;
        2.1.3. $y_1=Sbox\left(\left(\left(k_1+y_1\right)<<<x_2\right)\oplus k_3\right)$;
               $y_2=Mix\left(Sbox\left(\left(\left(k_2+y_2\right)>>>x_3\right)\oplus k_4\right)\right)$;
        2.1.4. $x_4=\left(Sbox\left(x_4+k_3\right)\oplus x_1\right)<<<k_2$; $x_5=\left(Sbox\left(x_5+k_4\right)+x_2\right)>>>k_1$;
               $x_6=Mix\left(\left(x_6+x_3\right)\oplus y_1\right)<<<x_4$;
        2.1.5. $k_3=Sbox\left(\left(Sbox\left(x_4\oplus k_3\right)+x_2\right)\oplus y_3\right)$;
               $k_4=Sbox\left(Mix\left(x_5+k_4+x_3\right)\oplus y_4\right)$;
```

```
2.1.6 $y_3 = Sbox\left(\left(\left(k_3 + y_3\right) <<< x_5\right) \oplus k_1\right)$;
      $y_4 = Mix\left(Sbox\left(\left(\left(k_4 + y_4\right) >>> x_6\right) \oplus k_2\right)\right)$.
2.2. $M_q = \left(y_1 \mid y_2 \mid y_3 \mid y_4\right)$
```

*Figure 1. Pseudocode of the TriGen algorithm*

Matrix $M'$ built on the basis of an array $U$ so that: $M'[i][j] = U\left[\left(j + 24 - i\right) \bmod 24\right]$ where $i, j = \overline{0,...,23}$, and an array $U$ takes values: $U$ = {1, 0, 2, 2, 1, 0, 2, 0, 1, 1, 1, 2, 0, 1, 2, 1, 0, 2, 0, 0, 1, 2, 0, 2 }.

### 4.2. Method and means of assessing the quality of ternary PRN

The proposed method of assessing the quality of PRN (Fig. 2), which makes it possible to assess the parameters and patterns of trit PRN as well as PRNG quality, is implemented in the following stages:

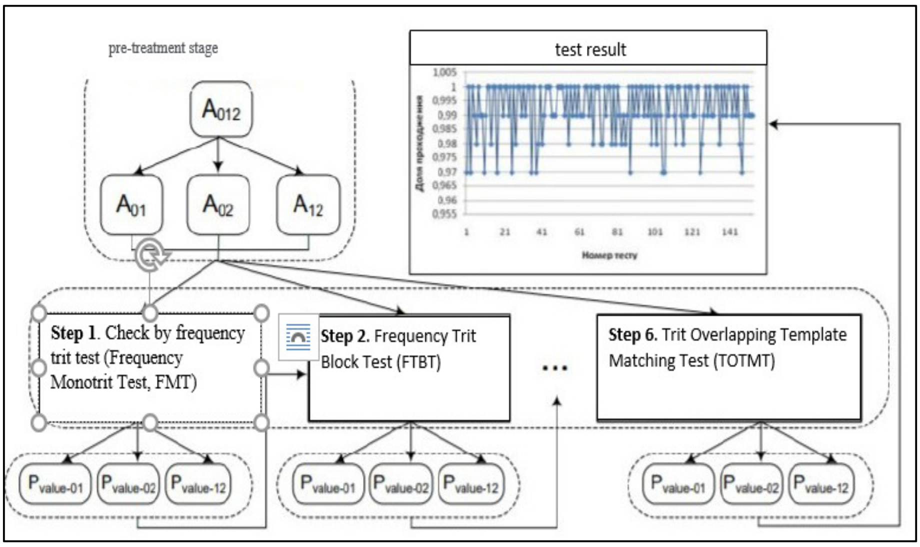

*Figure 2. Scheme of proposed method for ternary PRN quality assessment*

**Stage 1.** Frequency Monotrit Test, FMT.
**Stage 2.** Frequency Trit Block Test, FTBT.
**Stage 3.** Trit Runs Test, TRT.
**Stage 4.** Trit Test for the longest run in a block, TTLROB.
**Stage 5.** Non-Overlapping Template Matching Trit Test, NTMTT.
**Stage 6.** Trit Overlapping Template Matching Test, TOTMT.

At each of these six stages, the sequence is checked as follows:

1. First, each input ternary sequence $A_{012}$ is divided into 3 subsequences: $A_{01}$ (sequence $A_{012}$ with deleted 2), $A_{02}$ (sequence $A_{012}$ with deleted 1), $A_{12}$ (sequence $A_{012}$ with deleted 0).

2. Each of the obtained sequences is separately checked by trit tests, similar to the tests NIST STS. As a result of checking each test we get 3 values $P-value$: $P-value_{01}$, $P-value_{02}$, $P-value_{12}$. As in the tests NIST STS $P-value_{XY}$ (under $XY$ here and further we understand one of three possible combinations of sequences: "01", "02" and "12") will correspond to the probability that the studied sequence $A_{XY}$ not worse than true-accidental, i.e. if $P-value_{XY}=1$, then the generated sequence is perfectly random, and if $P-value_{XY}=0$, then the sequence is completely predictable.

3. Defined values $P-value_{01}$, $P-value_{02}$, $P-value_{12}$ each test is compared with $\alpha$ (the first kind of error is the probability that the random sequence is rejected). If $P-value_{XY} \geq \alpha$, then the sequence $A_{XY}$ is random with the confidence level of 99%, otherwise $P-value_{XY} \leq \alpha$ – sequence $A_{XY}$ rejected with a level of confidence of 99%. We will consider each test passed sequence $A_{012}$, if all the values obtained $P-value_{01}$, $P-value_{02}$, $P-value_{12}$ will be random with the confidence level of 99%, that is, inequalities will be satisfied $P-value_{01} \geq \alpha$, $P-value_{02} \geq \alpha$ and $P-value_{12} \geq \alpha$.

If the studied trit sequence passes all these tests, we will consider it pseudo-random. In addition, if at least one of the steps is not completed, the verification is completed and the sequence is considered predictable and unsuitable for any cryptographic applications.

Based on the proposed method of PRN quality assessment and the above input parameters, the TritSTS console software (in the C ++ programming language, analogically to NIST STS technique) was developed, which allowed to evaluate Trit sequences for randomness. The results (for one generated sequence TriGen-1) are listed in Table 1 and reflected in the statistical portrait (Fig. 3). During experimental study various sequences were generated and assessed by TritSTS as well NIST STS after ternary –> binary transformation.

*Table 1 – Example of the results of checking the generated sequence TriGen-1*

| Test | $P_{value_{01}}$ | $P_{value_{02}}$ | $P_{value_{12}}$ | The test is passed |
|---|---|---|---|---|
| FrequencyTritTest | 0.783309 | 0.783309 | 1.000000 | + |
| FrequencyTritTestWithinBlock | 0.625491 | 0.173564 | 0.149750 | + |
| RunsTritTest | 0.921404 | 0.921404 | 1.000000 | + |
| LongestRunOfTrit | 0.677032 | 0.901010 | 0.890900 | + |
| NonOverlappingTemplateTritTest | 0.931334 | 0.931334 | 0.931334 | + |
| NonOverlappingTemplateTritTest | 0.054678 | 0.330388 | 0.330388 | + |
| NonOverlappingTemplateTritTest | 0.054678 | 0.330388 | 0.931334 | + |
| NonOverlappingTemplateTritTest | 0.931334 | 0.636565 | 0.931334 | + |
| OverlappingTemplateTritTest | 0.330388 | 0.636565 | 0.636565 | + |

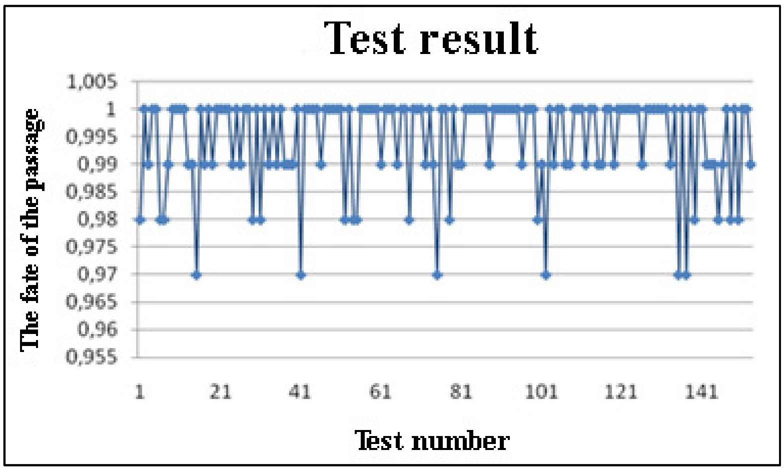

*Figure 3. Statistical portrait of the TriGen-1 sequence*

TritSTS testing results showed (Fig. 3) that sequence has passed complex statistical security control. Also we can specify that transformed binary PRN hasn't passed complex statistical security control by NIST STS.

## 5. Conclusions

This paper develops an approach to the generation and evaluation of PRN as well as PRNG quality, uses a comprehensive interpretation of generated PRN, differentiated probabilities $P-value_{01}$, $P-value_{02}$, $P-value_{12}$, ternary coefficients for the *erfc* error function and the incomplete gamma function *igamc*.

This approach makes it possible to generate and fundamentally evaluate the generally accepted statistical parameters and patterns of FMT, FTBT, TRT, TTLROB, NTMTT, TOTMT for trit PRN and, accordingly, to evaluate the cryptographic security of trit PRNG and their practical ability for various cryptographic applications.

## Acknowledgments

This research study was conducted as a part of research project titled № 0120U101400 “System of confidentiality ensuring for critical infrastructure based on deterministic quantum protocols”, funded by Ministry of Education and Science of Ukraine during 2020-2021 years.

## REFERENCES

1. NIELSEN M., CHUANG I.: Quantum Computation and Quantum Information, Cambridge University Press, Cambridge, United Kingdom, 2000, 676 p.
2. BOSTROM K., FELBINGER T.: Deterministic secure direct communication using entanglement, Physical Review Letters, **89** (2002) 18, 187902.
3. CHEN Y., BJORK G.: Generation and detection of photonic qutrits, 2007 European Conf. on Lasers and Electro-Optics and the Inter. Quantum Electronics Conf., 2007, 1-1, doi: 10.1109/CLEOE-IQEC.2007.4386921.
4. McGINTHY J., MICHAELS A.: Further Analysis of PRNG-Based Key Derivation Functions, IEEE Access, 7 (2019), 95978-95986, doi: 10.1109/ACCESS.2019.2928768.
5. CHEN S., HWANG T., LIN W.: Randomness Enhancement Using Digitalized Modified Logistic Map, IEEE Transactions on Circuits and Systems II: Express Briefs, **57** (2010) 12, 996-1000, doi: 10.1109/TCSII.2010.2083170.
6. HUA Z., ZHOU Y.: One-Dimensional Nonlinear Model for Producing Chaos, IEEE Transactions on Circuits and Systems I: Regular Papers, **65** (2018) 1, 235-246, doi: 10.1109/TCSI.2017.2717943.
7. A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications. NIST Special Publication 800-22, May 15, 2001, 164 p.
8. YUTAO F., GUIPING S.: A new testing method of randomness for true random sequences, 2014 IEEE 5th Inter. Conf. on Software Engineering and Service Science, 2014, 537-540, doi: 10.1109/ICSESS.2014.6933624.
9. AKCENGIZ Z., ASLAN M., KARABAYIR Ö. et al: Statistical Randomness Tests of Long Sequences by Dynamic Partitioning, Inter. Conf. on Information Security and Cryptology (ISCTURKEY), 2020, 68-74, doi: 10.1109/ISCTURKEY51113. 2020.9308005.