УДК 003.26.09

O.V. Krulikovskyi, S.D. Haliuk, L.F. Politanskyi

PRNG BASED ON MODIFIED TRATASHYPERCHAOTIC SYSTEM

In this paper we exploit discrete hyperchaotic systemfor pseudo-randomnumbersgenerator. Proposed chaotic PRNG was implemented in FPGA *Cyclone* IV EP4CE115 with Q5.27 fixed point arithmetic. The statistical properties of the generated pseudo-random sequences have been verified by means of the test suites NIST. Also, sensitivity to the parameters and initial conditions proposed PRNG is determined. Test results have show that there is a sense in further research PRNG's based on discrete chaotic systems.

Keywords:deterministic chaos; pseudorandom number generator; NIST tests; hyperchaos.

I. Introduction

Fast progress of telecommunication technologies in recent decades has led to total implementation of all aspects of human life. Among the derived advantages, appeared new challenges for the protection data with limited access against unauthorized access. This confirmed by the modernization of attacks on databases and information systems, state, private, and other users. The most effective means for data protection is its cryptographic encryption.

Increasing opportunities of computer facilities lead to increasing requirements for cryptographic means. These facts stimulate development and research of new areas in cryptology.One of perspective areas of research is the possibility of using deterministic chaos theory to information security [1]. As in classic cryptography among means based on chaos theory, a significant part is designed to generating pseudo random numbers. Pseudorandom number generator (PRNG)is an algorithm that generates a sequence of numbers, the elements of which are almost independent of each other and are subject to a given distribution (generally uniform). Chaotic system characterized a long-term unpredictable behavior and high sensitivity to initial conditions [1]. The trajectories of chaotic systems are unpredictable at large time intervals. Using of these properties is a prerequisite for development of new cryptographic algorithms [1, 2].

In the last decades many works devoted to the study chaos based cryptography were published. One of the first were the works of Wolfram, Matthews, Wheeler and others [3]. The first implementation of chaotic pseudorandom number generator (CPRNG) based on deterministic chaos had several disadvantages among which should be mentioned weak statistical properties of the pseudorandom sequences. The new research area attracted big interest and very quickly emerge works dedicated cryptanalysis CPRNG based on chaotic systems. Also, as mentioned above fulfillment of Moore's Law for development of computer technology has led to the development of new, more powerful tools for testing and cryptanalysis CPRNG. Consequently, the requirements were raised to a qualitatively new development PRNG and RNG. Therefore, many of the solutions proposed PRNG based on chaotic systems have lost their relevance.

All proposed CPRNG based on deterministic chaotic systems can be divided into four classes: PRNG based on continuous systems; CPRNG based on discrete maps, CPRNG based on cellular automata and CPRNG based on quantum chaos. In terms of cryptography for build secure CPRNG appropriate to use discrete systems as their implementation is simpler and they similar to traditional cryptosystems. Therefore, in this paper for build CPRNG we suggest using a discrete chaotic system.

The aim of this work is to develop a pseudo-random sequence generator based on modification of discrete hyperchaoticsystem introduced in [4, 5]. The section II presents a brief description of the chaotic system. In section III we present modification of chaotic system and provide details of proposed CPRNG.In section IV, the statistical properties sequences generated using FPGA implementations of CPRNG are validated via the test suite NIST [6]. The application of proposed CPRNG for image encryption is described in section V.Finally, conclusions are presented in Section VI.

II. Description of discrete hyperchaotic system

In [4, 5] was presented in general form a two-dimensional discrete chaotic system, described by (1):

$$\begin{cases} x(n+1) = a_1 x(n) - b_1|y(n)| + 1, \\ y(n+1) = a_2 y(n) - b_2|x(n)| + 1, \end{cases} \quad (1)$$

where $a_1, a_2, b_1$and b_2 are system parameters. Computed Lyapunov exponents of this system for $a_1 = a_2 = 0{,}23$ and $b_1 = b_1 = 1{,}439$ are equal 0,4075 and 0,4077. The system is hyperchaotic because both Lyapunov exponents are positive. Yu.H. Tratas used system (1) to analyze the developing of communication system [8], for synchronization and demodulation of chaotic signals used extended Kalman filter. Channels generator of system are symmetrical [7]. The phase portrait of the system shown in Fig. 1. Lyapunov exponents and phase portrait of system (1) was computed in mathematical modeling system MATLAB 2014a using computations with double-precision *IEEE* 754 format [8].

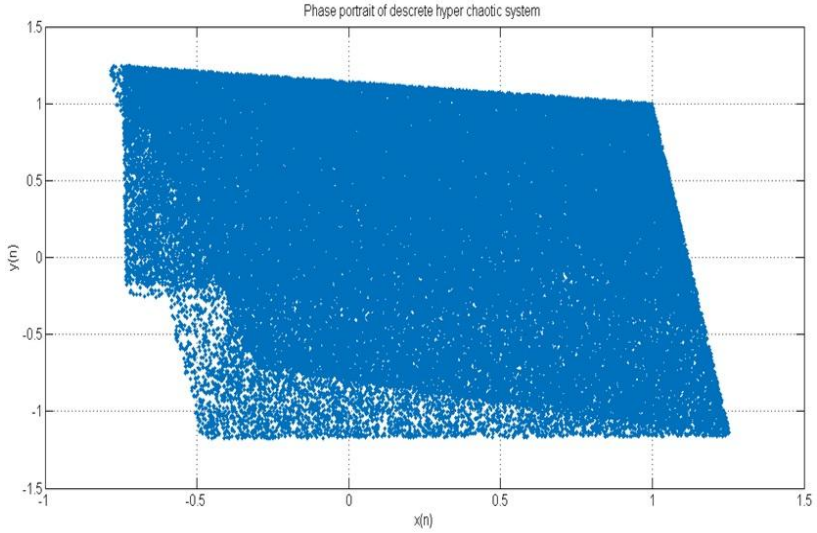


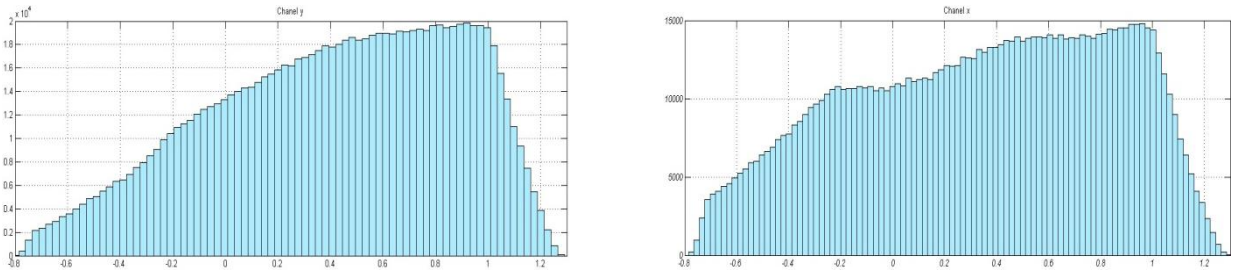
Fig. 1. Phase portrait of descrete hyper chaotic system (1)

III. PRNG based on hyper chaotic system

We propose a pseudo-random generator based on the six-dimensional modification of discrete nonlinear dynamic system (1). The modified system (1) becomes:

$$\begin{cases} x_1(n+1) = a_1|x_1(n)| - b_1|x_2(n)| + 1 \\ x_2(n+1) = a_2|x_2(n)| - b_2|x_3(n)| + 1 \\ x_3(n+1) = a_3|x_3(n)| - b_3|x_4(n)| + 1 \\ x_4(n+1) = a_4|x_4(n)| - b_4|x_5(n)| + 1 \\ x_5(n+1) = a_5|x_5(n)| - b_5|x_6(n)| + 1 \\ x_6(n+1) = a_6|x_6(n)| - b_6|x_1(n)| + 1 \end{cases} \quad (2)$$

The use of six-dimensional version of the system (1) is optimal for sensitivity and size of the key of encryption from our point of view. The six variables x_1, x_2, x_3, x_4, x_5 and x_6, that is the sequence of iterations are used to form a pseudo-random sequence. Distribution of sequence of iterations x_1, x_2, x_3, x_4, x_5 and x_6 is unbalanced (Fig. 2) and asymmetric therefore binary sequence x_1, x_2, x_3, x_4, x_5 and x_6 is unbalanced, scilicet have different number of symbols 0 and 1.



a *b*

Fig. 2. Histohram of distribution valuessequence of iterations system (1):
a – chanel *x*, *b* – chanel *y*

In [9], was identified balance, run and correlation as three basic properties of any periodic binary sequences that can be used as a test for randomness.In [10] to get a balanced binary representation of numbers, the first bits were discarded, thus received random sequences with uniform distribution. To select bits that satisfying balance we generated six matrix type (3) of size $32 \times N$ with elements $z_{i,j}, j \in [1,\ 32]$.

$$\begin{cases} z_{1,1} & z_{1,2} & \cdots & z_{1,32} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,32} \\ \cdots & \cdots & \cdots & \cdots \\ z_{i,1} & z_{i,2} & \cdots & z_{i,32} \\ \cdots & \cdots & \cdots & \cdots \\ z_{N,1} & z_{N,2} & \cdots & z_{N,32} \end{cases} \quad (3)$$

where $i \in [1,\ N]$– iterationnumber of sequences x_1, x_2, x_3, x_4, x_5 and x_6.

For each column we computed number of symbols "0" - N_0 and "1" - N_1, $N_0 + N_1 = N$, results shown in Fig. 3.

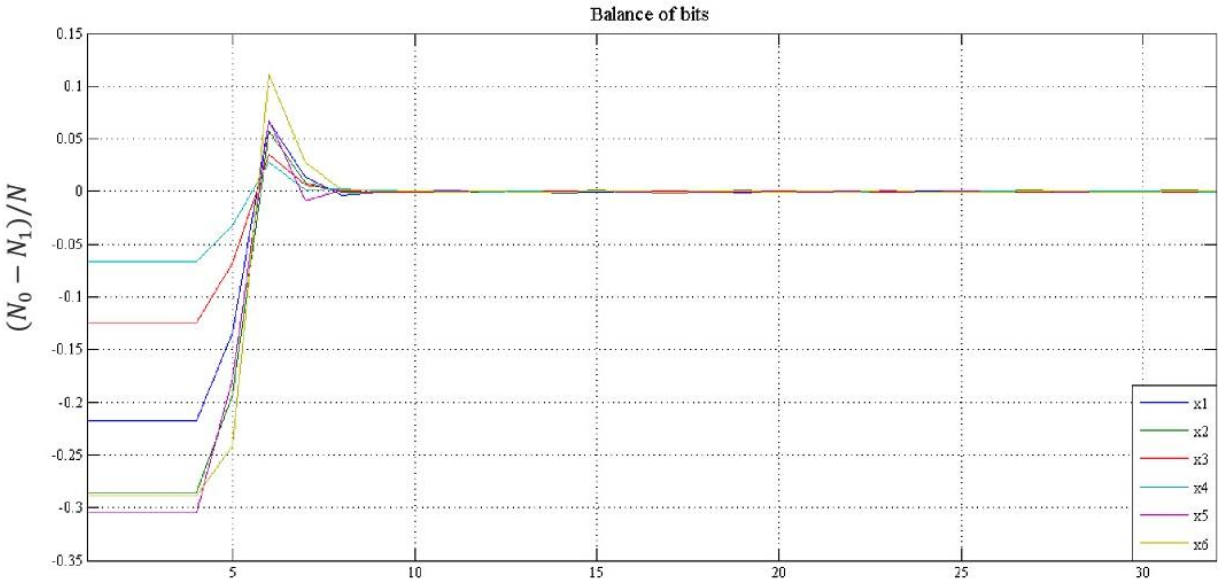


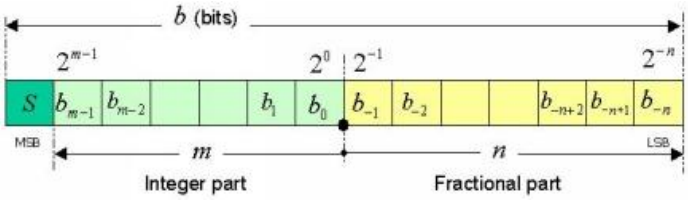
Fig. 3. The balance of bits generated chaotic system (2)

In this paper, we propose a scheme of CPRNG based on:

- Q5.27 fixed point arithmetic (Fig. 3).

- using most balanced *M* bits of fraction part of each sequences x_1, x_2, x_3, x_4, x_5 and x_6 for building true pseudorandom sequence.

- post processing of most balanced *M* bits of fraction part of each sequences of iterations x_1, x_2, x_3, x_4, x_5 and x_6



$$x = -2^{m}S + \sum_{i=-n}^{m-1} b_i 2^i$$

Fig. 4. Fixed point data specification
(*http://r2d2.enssat.fr/axes/CodageVirgFixe/codage.php*)

In our case M =16 bits. It can be concluded that to build cryptographically safely CPRNG should reject the first certain number of first bits. The block scheme of the proposed CPRNG is shown on Fig. 5.

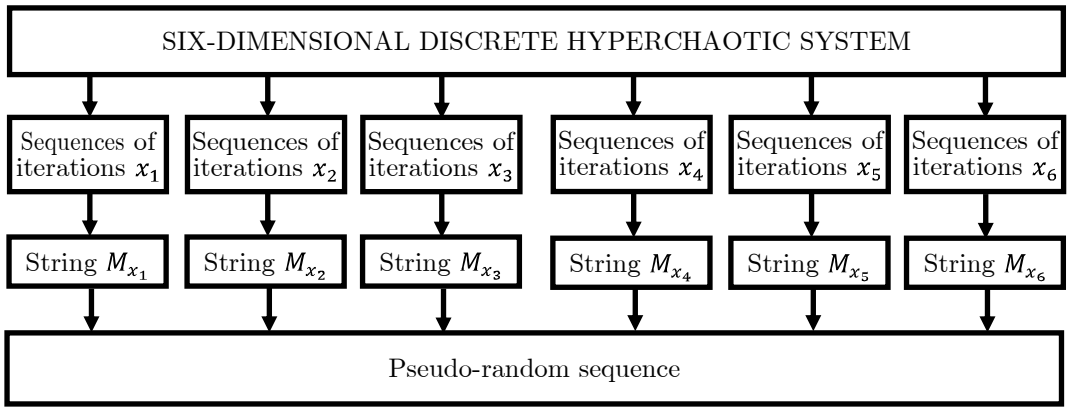


Fig. 5. Block scheme of proposed generator.

In block String M_x from Fig. 5, we took the 16 bits from each of the chaotic sequences of iterations within a range of 11 to 27 bits. As a result, we received 96 bits from all sequences of iterations.

Secret key of encryption divides on 18 sub keys: twelve parameters a_1, a_2, a_3, a_4, a_5, a_6, b_1, b_2, b_3, b_4, b_5 and b_6 and six initial conditions $x_1(0)$, $x_2(0)$, $x_3(0)$, $x_4(0)$, $x_5(0)$ and $x_6(0)$ of chaotic system (2). In this study, we used a key length of $18 * 32 = 576$ bits.

IV. FPGA implementation and of Testing of proposed PRNG

For implementation proposed CPRNG on FPGA we used Altera DE2-115 Development and Education Board which includes FPGA *Cyclone* IV EP4CE115, for post processing used MATLAB 2014a and Simulink.

The procedure of the implementation is following:

- implementation system (2) in Simulink (Fig. 6 and Fig. 7),
- generation VHDL description developed model using HDL Coder,
- Building FIL model using HDL Verifier tool,
- Quartus II project creation,
- final configuration and download the project from Simulink to FPGA.

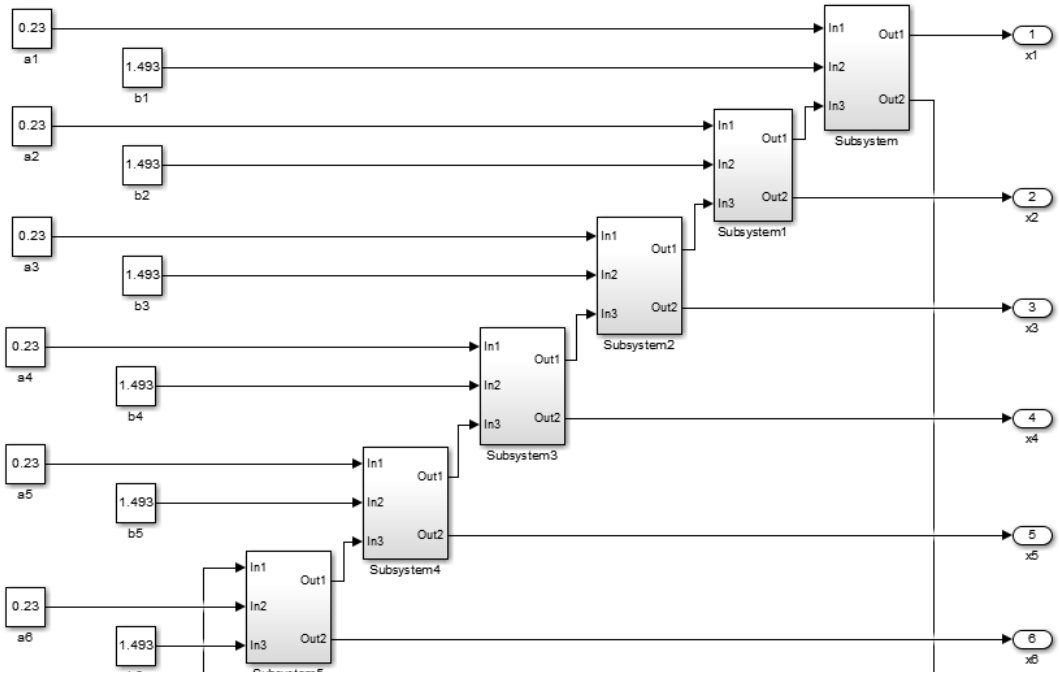


Fig. 6.Simulink model of system (2)

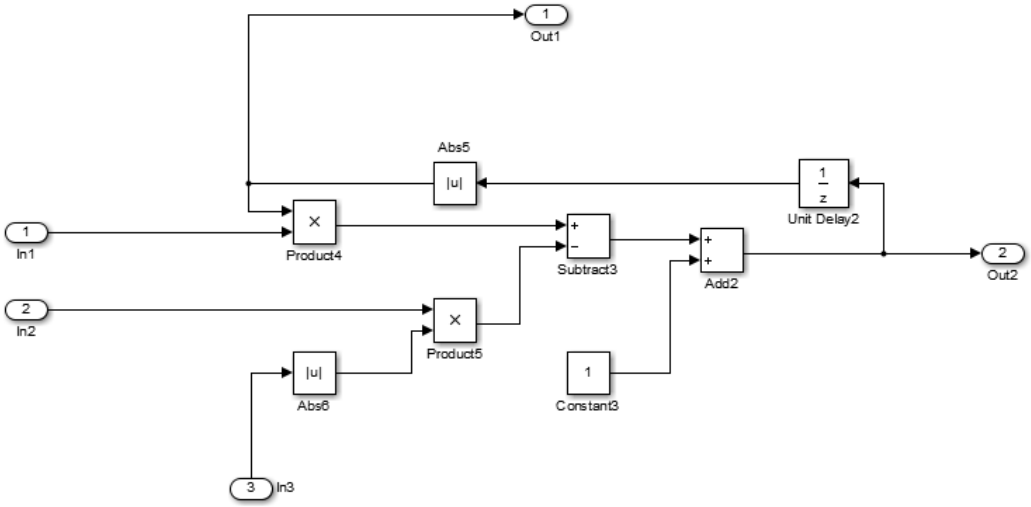


Fig. 7.Structure of subsystem in Fig.6

The one of most common statistical test is the suite of statistical tests provided by the National Institute of Standards and Technology (NIST test). This suite consists of 15 tests [6]. If all tests are passed for a sequence, then the sequence is marked as cryptographically safe [9]. Tab. 1 presents the NIST statistical tests results for binary pseudo-random sequences generated by the chaotic system (2). The NIST tests are performed on a binary sequence oflength 10^9 bits which was divided on 1,000 sequences (with 1 million strings). Used parameters and initial condition for testing PRNG is next: $a_1 = 0{,}239$, $a_2 = 0{,}231$, $a_3 = 0{,}239$, $a_4 = 0{,}3$, $a_5 = 0{,}323$, $a_6 = 0{,}123$, $b_1 = 1{,}493$, $b_2 = 1{,}495$, $b_3 = 1{,}4931$, $b_4 = 1{,}9493$, $b_5 = 1{,}4493$, $b_6 = 1{,}3$, $x_1^0 = 0{,}2$, $x_2^0 = 0{,}5$, $x_3^0 = 1$, $x_4^0 = 0{,}25$, $x_5^0 = 1{,}5$, $x_6^0 = -0{,}9$.

TABLE 1

NIST STATISTICAL TESTS RESULTS OF THE PSEUDO-RANDOM SEQUENCES GENERATED OF PROPOSED CPRNG*

№	Test	*P* - value	Proportion	Status
1	Frequency (Monobit) Test	0,468595	1,000	**Pass**
2	Frequency Test within a Block	0,253551	0,978	**Pass**

3	Runs Test	0,368773	0,989	**Pass**
4	Test for the Longest Run of Ones in a Block	0,189397	1,000	**Pass**
5	Binary Matrix Rank Test	0,804337	0,989	**Pass**
6	Discrete Fourier Transform Test	0,579479	0,989	**Pass**
7	Non-overlapping Template Matching Test	0,368773	0,989	**Pass**
8	Overlapping Template Matching Test	0,879806	0,989	**Pass**
9	Maurer's "Universal Statistical" Test	0,739918	0,967	**Pass**
10	Linear Complexity Test	0,694743	0,978	**Pass**
11	Serial Test	0,368773	0,989	**Pass**
12	Approximate Entropy Test	0,602458	1,000	**Pass**
13	Cumulative Sums Test	0,761937	0,989	**Pass**
14	Random Excursions Test	0,517442	1,000	**Pass**
15	Random Excursions Variant Test	0,392456	1,000	**Pass**

*For tests № 7, 8, 11, 13, 14, 15 we indicated the minimum value P - value and proportion among all the subtests (see details about these tests in [6]).

V. Aplication of proposed CPRNG for image encryption

We use the proposed CPRNG (Fig. 5) to encrypt images by diffusion method. Encryption is carried out according to the following algorithm:

1. Using the proposed CPRNG we generate image of the same size and type as the original image.

2. Then we add values of color components for each pixel of generated image to the values of color components corresponding pixel in the original image and thus received an encrypted image as

$$E = (M + K) mod L,$$

where M – the numerical value of the color component of pixels, K – pseudo-random numbers, L – the number whose value is determined by the number of bits to represent the color component of pixel n, and typically $L = 2^n$, E – encrypted color component of image.

The encryption process is performed according to

$$M = (E - K) mod L.$$

The block scheme of image encryption is shown on Fig. 8. To generate pseudorandom image we used the same parameters as for testing pseudorandom sequances. The size of test image is 512×512 pixels.

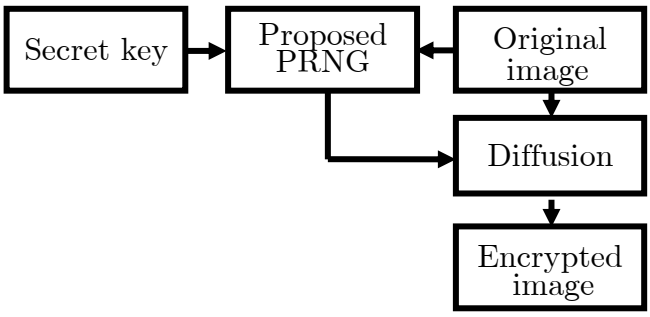
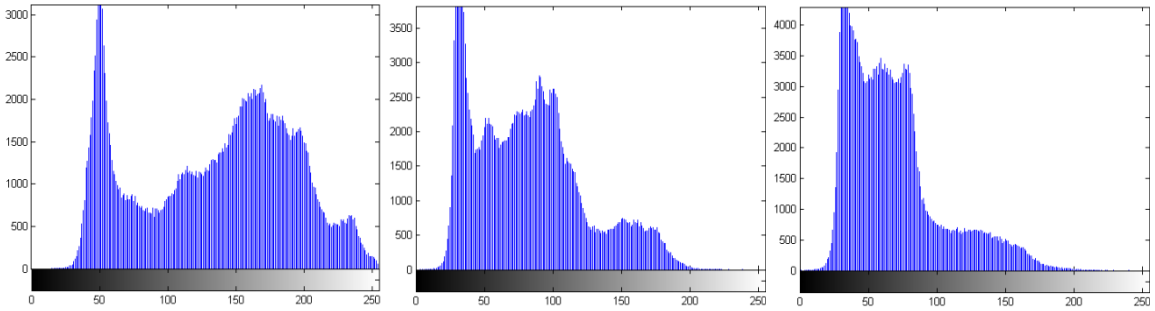


Fig. 8. Block scheme of the example of image encryption

a *b*

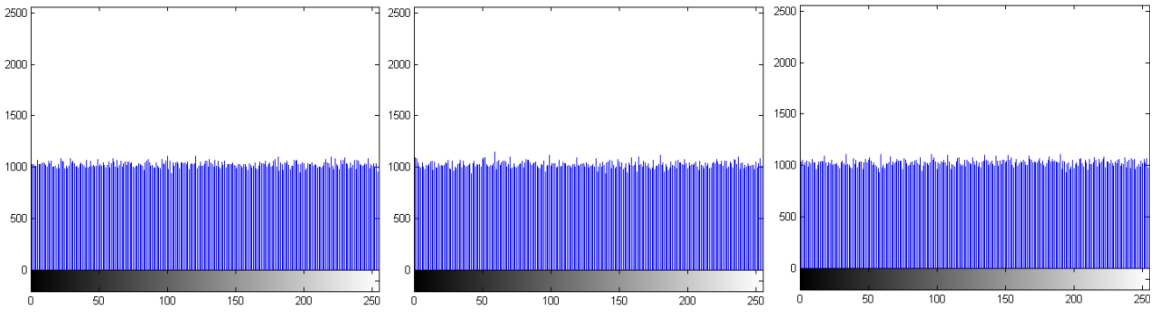
Fig. 9.*a* – test image, *b* – encrypted test image.

In Fig. 9 *b* shown the encrypted test image after one cycle of diffusion. Histograms of test image and test encrypted image are shown on Fig. 10 and 11. The histogram of encrypted image has uniform distribution that is necessary for reliable algorithms of image encryption.



a *b* *c*

Fig. 10. Histogram of test image: *a* – component *Red*, *b* – component *Green*, *c* – component *Blue*



a *b* *c*

Fig. 11. Histogram of encrypted test image: *a* – component *Red*, *b* – component *Green*, *c* – component *Blue*

The correlation between components of color the test image and encrypted image are for component *Red*: 0,001855027394912; for *Green*: -0,001608310589159; and for *Blue*: 0,001507968989894. The average value of the pixels color in test image is 94,6445. The average value of color components of pixels in encrypted test image are 127,4295, confirming uniform distribution of colors. On Fig. 9 *b* color image contours are not observed as diffusion occurs for each pixel.

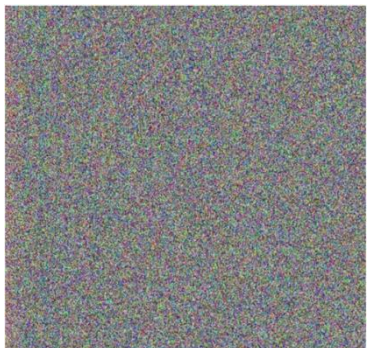
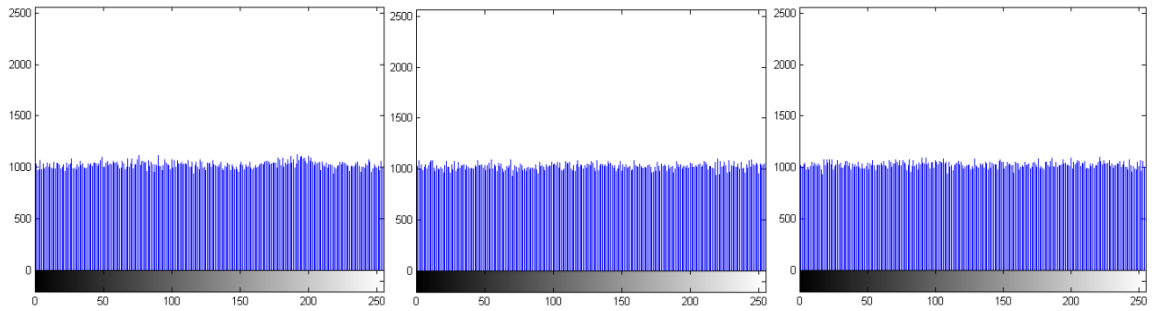


Fig. 12. Decrypted test image by changing sub key $x_0$on 2^{-27}



a *b* *c*

Fig. 13. Histogram of decrypted test image by changing sub key x_1^0 on 2^{-27} shown on Fig. 10: *a* – component *Red*, *b* – component *Green*, *c* – component *Blue*

Fig. 12 shows CPRNG sensitivity to initial conditions. In this case, the simulation software allows to show the generator sensitivity to 2^{-27}. The correlation between components of color the test image Fig. 9 *a*. and the image in Fig. 12 are for component *R*: 0,001598699445359; for *G*: -0,001654461583456; and for *B*: 0,001155895236862. The average value of color components*Blue* of pixels in Fig. 14 are 127,6415.

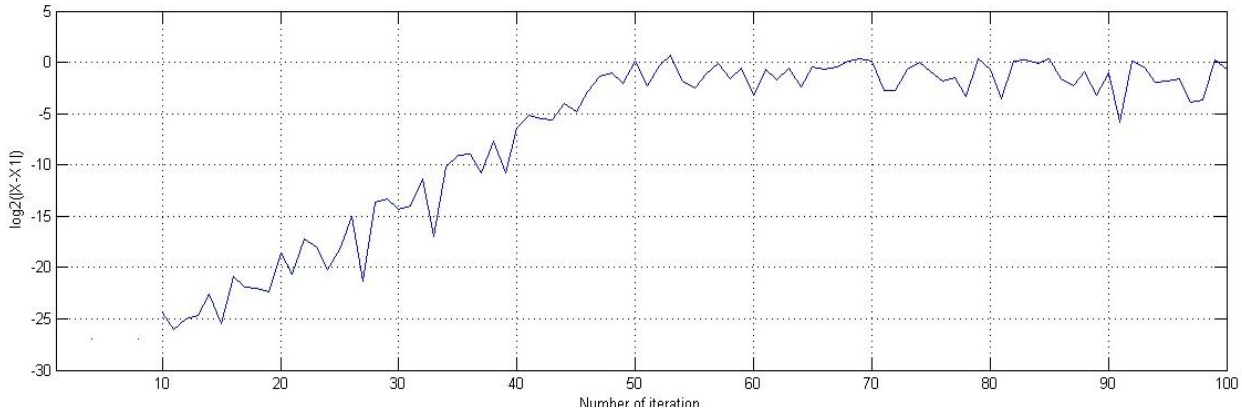


Fig. 14. Difference between two sequences of $x_1$with different parameters (2) a_1: for X1 – parameter $a_1 = 0{,}239$; for X – parameter $a_1 = 0{,}239 + 2^{-27}$

Fig. 14 confirms one of the basic properties of deterministic chaotic systems, such as high sensitivity to initial conditions and parameters in which their small change leads to changes in behavior of the system over time. The more we change the value of initial conditions relative to the faster start-chaotic system change their behavior over time. Therefore, realization in time sequences of iterations (2) will be different, if we change one of sub keys on the smallest value in within using arithmetic method.

VI. Conclusion

In this article we have proposed a CPRNG based on multidimensional discrete hyper chaotic system. In order to increase performance of the encryption systems, we used a discrete two-dimensional hyper-chaotic system and the pseudo-random sequence was generated using the part of bits (in each string) with maximum balance. Results of the study are confirmed via the NIST tests. Determined that sensitivity to the initial conditions and parameters proposed generator is2^{-27}. Eventually, we can say that received results of testing, PRNG based on discrete maps, from point of view information security are interesting for further research.

References

1. Птицын Н. Приложение теории детерминированного хаоса в криптографии, Москва, МГТУ им. Н. Э. Баумана, 2002. – 80 ст.

2. Fridrich J. Symmetric Ciphers Based on Two-Dimensional Chaotic Maps. Inter. Journal of Bif. and Chaos, Vol. 8, No. 6 (1998) 1259–1284.

3. Şefika Şule Erçetin, Santo Banerjee, “Chaos, Complexity and Leadership 2013”, Springer International Publishing, pp. 566, 2013

4. A. Rodriguez-Vazquez, J. Huertas, A. Rueda, B. Perez-Verdu, and L. O. Chua. Chaos from Switched-Capacitor Circuits: Discrete Maps, Proc. of the IEEE, Special Issue on Chaotic Systems, pp. 1090-1106, Aug. 1987

5. Yu.H. Tatras, Application of statistical communication theory to the problems of receiving chaotic oscillations, Achievements of Modern Radioelectronics, № 11, 57-80 pp., 1998.

6. National Institute of Standards and Technology, U.S Department of Commerce. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST Special Publication 800-22, Revision 1a, April 2010.

7. B.I. Shahtarin, P.I. Kobylkina, Yu.A. Sidorkina, A.V. Kondratev, S.V. Mitin, Generatory haoticheskih kolebaniy, Moscow, Gelios ARV, pp. 53-54, 2007.

8. IEEE, "IEEE standard Floating-Point Arithmetic," IEEE Std 754-2008, pp. 1-58, Aug 2008.

9. Bernard Sklar, “Digital Communications: Fundamentals and Applications (2nd Edition)”, Prentice Hall PT R, pp. 1079, January 21, 2001.

10. Lahcene Merah, Adda Ali-Pacha, Naima Hadj Said, Mustafa Mamat, APseudo Random Number Generator Based on the Chaotic System of Chua’s Circuit, and its Real Time FPGA Implementation, Applied Mathematical Sciences, Vol. 7, no. 55, pp. 2719 – 2734, 2013.

Надійшла 03.05.2016 р. Рецензент: д.т.н., проф. Горбенко І.Д.