Journal of Theoretical and Applied Information Technology
15th December 2016. Vol.94. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SEQUENTIAL AND PARALLEL COMPOSITION OF ROUND TRANS-FORMATIONS FOR CONSTRUCTION OF AN ITERATIVE ALGO-RITHM FOR STOCHASTIC DATA PROCESSING

IVANOV MIKHAIL, STARIKOVSKIY ANDREY, POPOV NIKOLAY, MAMAEV DMITRY, SKITEV ANDREY, KUTEPOV STANISLAV, BABALOVA IRINA

National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)
Kashirskoe Highway 31, 115409, Moscow, Russian Federation
E-mail: msa@dozen.mephi.ru, avstarikovskij@mephi.ru

ABSTRACT

The trend of recent years has been the advent of algorithms based on the use of 2D and 3D stochastic transformations. This article proposes a new iterative cryptographic algorithm of 3D stochastic data transformation, which is basically aimed for information security solutions. The algorithm is characterized by the high degree of parallelism at the level of elementary operations. It is proposed to design an iterative transformation based on the consistent and parallel composition of the round operations. Such an approach to the design of the iterative cipher blocks has not previously been used. Its use in a software implementation of the cryptographic transformation became possible due to the advent of the heterogeneous supercomputer technologies. It became possible to perform within a round in parallel (without performance compromising) complex transformations of different trajectories and then at the output perform the parallel composition of the results. As a result, the task of the encryption function inverting becomes computationally unsolvable with fewer rounds of transformation.

Keywords: *Cryptographic Primitives, 3D Transformation, Cube Architecture, Iterative Stochastic Trans-Formation, Mixstate Transformation*

1. INTRODUCTION

Information security threats analysis and analysis of trends in IT development allows making an unambiguous conclusion about the constantly increasing role of stochastic methods of information security. Methods are called stochastic if they are directly or indirectly based on the pseudo-random number generators (PRNG). As an example of a general stochastic method of information security (IS), a method of introducing uncertainty to the means and objects of protection can be mentioned. Every problem of IS can be successfully solved by means of PRNG. Thus, in some cases, stochastic methods are the only possible mechanism to ensure information security from an active intruder. A special case of the stochastic methods are cryptographic methods of information security.

The term "stochastic" in the context of IP for the first time, apparently, was used by S.A. Osmolovsky when developing codes for detecting and correcting errors occurring during data transmission through communication channels [1, 2].

The proposed stochastic codes have unique properties, we highlight two:

- ability to ensure the prescribed probability of correct information reception;
- ability (in addition to providing noise immunity) to solve the other two equally important tasks of IS: to ensure privacy and integrity of the transmitted information.

A stochastic transformation algorithm DOZEN of Cube architecture aimed for implementation using heterogeneous supercomputing technologies is proposed in [3-5]. The 3D transformations are characterized by the high degree of parallelism at the level of elementary operations. Building cryptographic primitives of pseudo-random numbers generation, hash, block and stream encryption is the field of application of nonlinear multiple stochastic transformations [3, 6, 7]. The new iterative transformation of the DOZEN family, which design is based on the sequential and parallel composition of the round operations is discussed.

2. DOZEN+ TRANSFORMATION

The DOZEN+ transformation sequence for a block M of data, 512 bits size (4 × 4 × 4 × 8),

Journal of Theoretical and Applied Information Technology
15th December 2016. Vol.94. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

having the structure shown in Fig. 1,a, where $a_{x,y,z}$, $x = 0, 1, 2, 3$, $y = 0, 1, 2, 3$, $z = 0, 1, 2, 3$, are bites, is as follows:

– add (XOR) round key K_0 to the data block;

– the first round (*MixLayersX* transformation): divide the resulting data block into Layers S_{x0}, S_{x1}, S_{x2}, S_{x3} along the x-axis (Fig. 1,b); conduct a two-dimensional stochastic transformation (MixLayer) of layers S_{x0}, S_{x1}, S_{x2}, S_{x3} by performing sixteen (number of bytes) *SubBytes* operations, *MixState* operation and *AddRoundSubKey* operation for each layer S_{xj};

– the second round (*MixLayersY* transformation): divide the resulting data block into Layers S_{y0}, S_{y1}, S_{y2}, S_{y3} along the y-axis; conduct a two-dimensional stochastic transformation (MixLayer) of layers S_{y0}, S_{y1}, S_{y2}, S_{y3} by performing sixteen (number of bytes) *SubBytes* operations, *MixState* operation and *AddRoundSubKey* operation for each layer S_{yj};

– the third round (*MixLayersZ* transformation): divide the resulting data block into Layers S_{z0}, S_{z1}, S_{z2}, S_{z3} along the z-axis; conduct a two-dimensional stochastic transformation *(MixLayer)* of layers S_{z0}, S_{z1}, S_{z2}, S_{z3} by performing sixteen (number of bytes) *SubBytes* operations, *MixState* operation and *AddRoundSubKey* operation for each layer S_{zj};

The round keys K_0, K_1, K_2, K_3 are generated from initial key K using the *KeyExpansion* procedure.

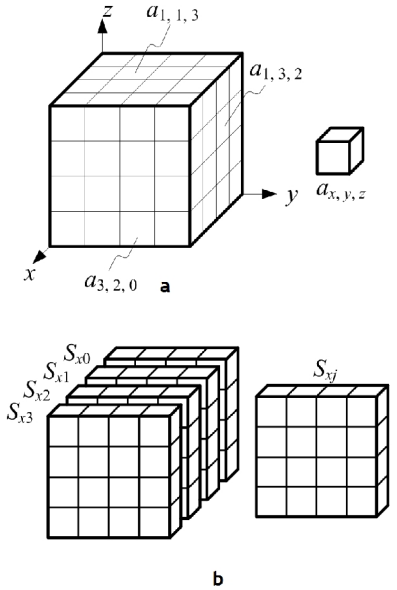

Figure 1: 3D Data Stochastic Transformation Algorithms: A – The Format Of The Data Block (State) And Round Keys K_i, B – State Division Into Layers Along The X-Axis.

Thus, the sequence of the 3D nonlinear multilayer transformation DOZEN+ has the form shown in Fig.2 and is defined as follows:

$$S = M;$$

$$C = DOZEN+(M) = MixLayersZ(K_3, MixLayersY(K_2, MixLayersX(K_1, (S \oplus K_0))));$$

$$MixLayersX(K_1, S) =$$
$$= MixLayer(K_{13}, S_{x3}) \,||\, MixLayer(K_{12}, S_{x2}) \,||\, MixLayer(K_{11}, S_{x1}) \,||\, MixLayer(K_{10}, S_{x0});$$

$$MixLayersY(K_2, S) =$$
$$= MixLayer(K_{23}, S_{y3}) \,||\, MixLayer(K_{22}, S_{y2}) \,||\, MixLayer(K_{21}, S_{y1}) \,||\, MixLayer(K_{20}, S_{y0});$$

$$MixLayersZ(K_3, S) =$$
$$= MixLayer(K_{33}, S_{z3}) \,||\, MixLayer(K_{32}, S_{z2}) \,||\, MixLayer(K_{31}, S_{z1}) \,||\, MixLayer\ (K_{30}, S_{x0});$$

$$MixLayer(K_{ij}, S_{mj}) =$$
$$= AddRoundSubKey(K_{ij}, (MixState(SubBytes\ (S_{mj})))) = MixState(SubBytes\ (S_{mj})) \oplus K_{ij};$$

$$|M| = |S| = |K_i| = 512,\ |S_{mj}| = |K_{ij}| = 128;\ i = 0, 1, 2, 3;\ j = 0, 1, 2, 3;\ m = x, y, z;$$

$$K_i = K_{i3} \,||\, K_{i2} \,||\, K_{i1} \,||\, K_{i0},\ S = S_{x3} \,||\, S_{x2} \,||\, S_{x1} \,||\, S_{x0} = S_{y3} \,||\, S_{y2} \,||\, S_{y1} \,||\, S_{y0} = S_{z3} \,||\, S_{z2} \,||\, S_{z1} \,||\, S_{z0};$$

where M is data input block, S is the state of the algorithm; K_i are round keys, *SubBytes* is the byte-substitution transformation of the layer state S_{mj}; *Mix-State* is the shuffle transformation of the layer state S_{mj}; *MixLayersX* is the layer-based transformation along the x-axis; S_{x3}, S_{x2}, S_{x1}, S_{x0} are the layers of the state along the x-axis; *MixLayersY* is the layer-based transformation along the y-axis; S_{y3}, S_{y2}, S_{y1}, S_{y0} are the layers of the state along the y-axis; *MixLayersZ* is the layer-based transformation along the z-axis; S_{z3}, S_{z2}, S_{z1}, S_{z0} are the layers of the state along the z-axis.

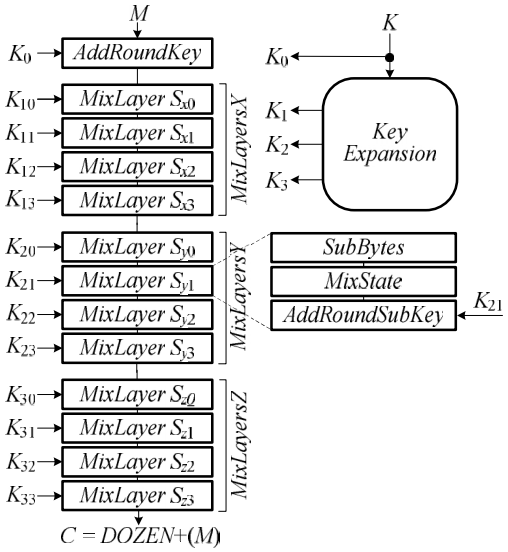

Figure 2: 3D Data Stochastic Transformation Algorithm DOZEN+.

Journal of Theoretical and Applied Information Technology
15th December 2016. Vol.94. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Let Q be the state of the layer: $|Q| = 128$, $Q = (Q_{16} \dots Q_1)$, Qi ∈ $GF(2^8)$,

$|Q_i| = 8$, $i = 1, \dots, 16$. Nonlinear transformation SubBytes (layer state byte substitution) is defined as follows:

$$SubBytes(Q) = SubBytes(Q_{16} \| \dots \| Q_1) = SubByte(Q_{16}) \| \dots \| SubByte(Q_1),$$

where $SubByte(Q_i)$ is bite Q_i substitution transformation.

3. SEQUENTIAL AND PARALLEL COMPOSITION OF ROUND TRANSFORMATIONS

Traditionally, only sequential composition of round transformations RF_i is used for constructing of the encryption function E:

$$E = RF_n \bullet RF_{n-1} \bullet \dots \bullet RF_2 \bullet RF_1,$$

where n is the number of rounds; to ensure the required level of cryptoresistability it is necessary to perform a large number of rounds.

It is proposed to construct the E function on the basis of the sequential and parallel composition of round transformations RF_i

The basic idea of it is to design a sequential machine, the number of memory elements of which exceeds the bit width of the input memory.

Such approach to construction of iterative block ciphers has not been used previously; its use in software implementations of cryptographic transformations was made possible due to the emergence of HSCT.

As the result, it became possible within a round to perform in parallel (in other words, without decrease in performance) various trajectories of complex transformations, and then to implement the parallel composition of the obtained results as an output.

As a result, the task of inverting the encryption function becomes computationally intractable with a smaller number of rounds of transformation.

Consider the principles of the iterative algorithm for stochastic data processing using parallel and sequential composition of elementary transformations (Fig. 3).

Within each round, in fact, N copies of the input block are formed, each copy C_{ij} Is subjected to stochastic transformation $C_{ij} := E_{ij}(C_{ij}, K_{ij})$, where K_{ij} are round keys of the i-th round, $j = 1, 2, \dots, N$. The transformed values C_{ij} come to the inputs of a combinational scheme F_i, a function of which is the parallel composition of different trajectories of round transformations, the result of combinational scheme $C := F_i(C_{i1}, C_{i2}, \dots, C_{iN})$ is declared the result of the i-th round.

Fig. 3, b is an example of the construction of an iterative algorithm for stochastic data processing, where parallel composition is performed using the operation of addition modulo two, and $C_{ij} = MixState \bullet SubBytes \bullet AddThreadKey$.

Fig. 4 shows variants of the combinational scheme F_i.

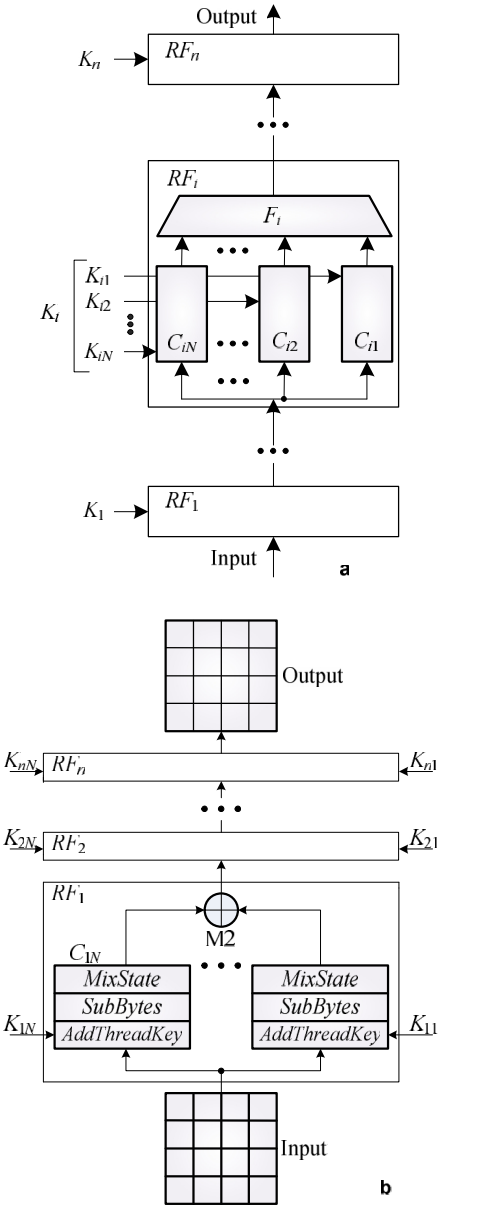

Figure 3: Sequential And Parallel Composition Of Round Transformations: A Is The General Scheme, B Is Based On The Example Of 2D Transformations.

Journal of Theoretical and Applied Information Technology
15th December 2016. Vol.94. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

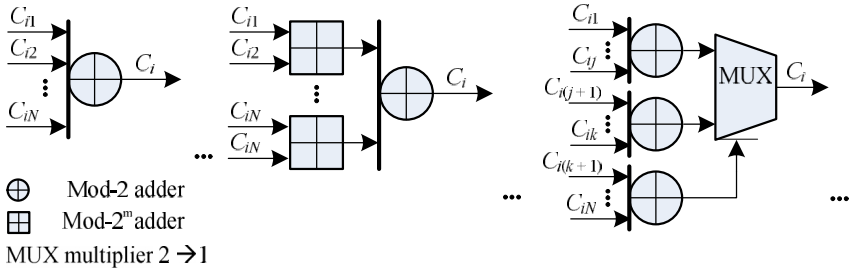

Figure 4: Variants Of The Combinational Scheme F_i.

4. MIXING LAYER STATE TRANSFORMATION MIXSTATE

Sequential LFSR.

Initial information for the construction of the LFSR (Linear Feedback Shift Register, shift register with linear feedback), in other words, PRNG, operating in the finite field $GF(p^n)$ (p – prime, n – natural), a so-called characteristic polynomial $\varphi(x) = a_N x^N + \ldots + a_1 x - 1$.

The degree of the polynomial determines the number of PRNG registers, while non-zero coefficients a_i - the nature of the feedback.

Polynomial of degree N correspond to the LFSR shown in Fig. 5,b.

The equations of sequential LFSR designed according to the diagram of Galois, have the form

$$Q_j = Q_{j+1} + a_j Q_1, j = 1, \ldots, (N-1),$$
$$Q_N = a_N Q_1.$$

or in matrix form $Q = QT$ where T is the companion matrix $N \times N$.

Parallel LFSR.

Sequential LFSR can only be used to generate a sequence of p^n number base characters, which can be removed from the output of one of the registers.

In order to implement a k-channel generator, synthesis of parallel LFSR running k times faster than the original generator is necessary (in other words, a generator that performs in one cycle changes that are performed in k cycles in the original sequential LFSR).

Consider the properties of parallel LFSR. The general form of the generator corresponding the equation $Q = QT^k$, is shown in Fig. 5, a. The extent to which there is a multiplication in each multiplier unit (MU) is defined by the corresponding coefficient $a_{ij} \in GF(p^n)$ of the accompanying matrix $V = T^k$. If $a_{ij} = 0$, this is equivalent to the absence of connection between the output of the i-th register of the generator and the input of the j-th addition block in $GF(p^n)$.

Since the zero state of all registers of the generator is forbidden, the maximum possible number of states of the device and thus the maximum possible length of the formed binary sequence extracted from the output of any register are $p^{nN} - 1$. In this case, the generator state diagram consists of one trivial cycle and one cycle of maximum length equal to $p^{nN} - 1$.

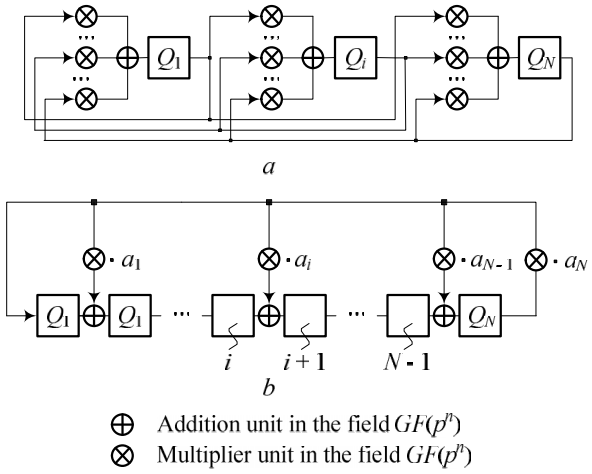

Figure 5: Mixstate Transformation: A Is A Parallel LFSR Corresponding The Equation Q(T + 1) = Q(T) T^k; B Is A Sequential LFSR Designed According To The Diagram Of Galois And Corresponding The Equation Q(T + 1) = Q(T) T.

Let φ (x) be a primitive polynomial of degree N, then the following is true. The formed sequence has a maximum period $L = p^{nN} - 1$ if and only if the numbers L and k are coprime. For $k = 1$ the $\varphi(x)$ to be primitive is a necessary and sufficient condition for obtaining a sequence of maximum length.

Consider $p = 2$, $n = 8$, $N = 16$, $k = 16$. Linear stochastic transformation MixState (mixing layer state) on the basis of PRNG operating in $GF(2^8)$ and designed according to the concept of Galois, is determined by the following expressions:

$$MixState(Q) =$$
$$= R^{16}(Q) = R^{16}(Q_{16} \| \ldots \| Q_1) = (a_{16}Q_1 \| Q_{16} + a_{15}Q_1 \| \ldots \| Q_2 + a_1 Q_1)^{16} = QT^{16},$$

where all operations are conducted in the field $GF(2^8)$, $a_i \in GF(2^8)$ are the coefficients of the characteristic polynomial $\varphi(x) = a_{16}x^{16} + a_{15}x^{15} +$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$a_{14}x^{14} + \ldots + a_2x^2 + a_1x - 1$, which is primitive over the field $GF(2^8)$, T is the square matrix of the order 16×16.

The equation of the basic linear transformation R (multiplication of the state of the layer Q by T) is like:

$$Q_j = Q_{j+1} + a_jQ_1, j = 1, \ldots, 15,$$
$$Q_{16} = a_{16}Q_1.$$

The Grasshopper algorithm [8] uses LFSR, built by Fibonacci scheme for $p = 2$, $n = 8$, $N = 16$ to implement the state mixing transformation. However, the proposed variant based on Galois scheme has better dispersing and mixing properties.

5. 3DOZEN+ TRANSFORMATION

The transformation sequence for a block M of data, 512 bits size ($4 \times 4 \times 4 \times 8$), having the structure shown in Fig. 1,a, is defined as follows:

$$S = M; S_1 = S \oplus K_0; C = 3DOZEN+(M) = RF_3(K_3, RF_2(K_2, RF_1(K_1, S_1)));$$
$$RF_i(K_i, S_i) = MixLayersX(K_i, S_i) \oplus MixLayersY(K_i, S_i) \oplus MixLayersZ(K_i, S_i);$$
$$K_i = K^x_{i3} \| K^x_{i2} \| K^x_{i1} \| K^x_{i0} = K^y_{i3} \| K^y_{i2} \| K^y_{i1} \| K^y_{i0} = K^z_{i3} \| K^z_{i2} \| K^z_{i1} \| K^z_{i0};$$

where RF_i is the round function, $i = 1, 2, 3$; M is input data block, C = *3DOZEN+(M)* is export data block, S_i is input of the i-th round. 3DOZEN + transformation sequence is shown in Fig. 6.

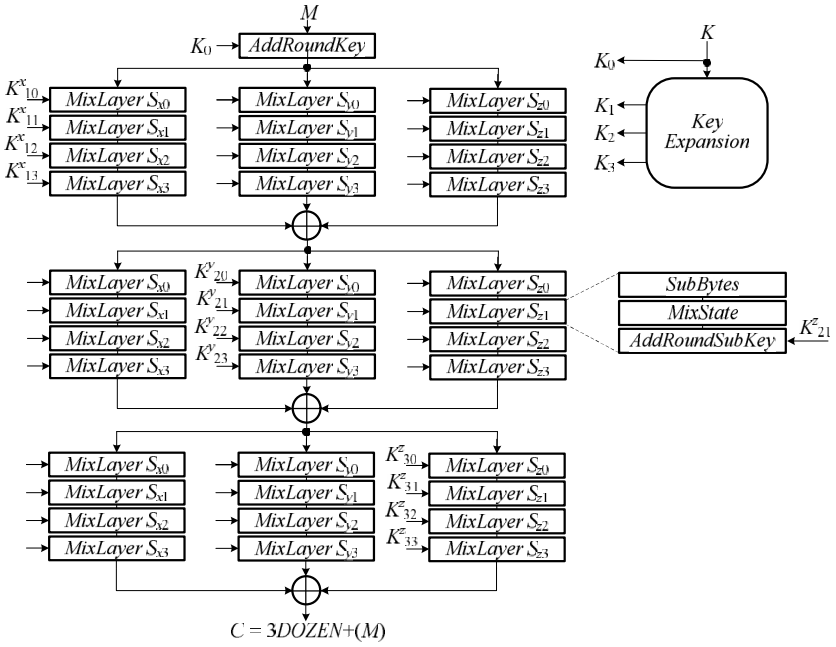

Figure 6: Transformation Sequence Of The 3DOZEN+.

6. CONCLUSION

The high degree of parallelism at the level of elementary transformations is characteristic for the proposed algorithm 3DOZEN+ due to the possibility of parallel execution of *MixLayer* operations. Thus, the use of CUDA technology [9, 10] allows to significantly simplify the process of software development. The characteristic feature of the 3DOZEN+ transformation is a hybrid architecture, the essence of which is sequential and parallel composition of round transformations *MixLayers*. As a result, it became possible to perform within the round in parallel (without degrading performance) various trajectories of complex transformations, and then at the output to carry out a parallel track of the results. Thus, the task of the encryption function inverting becomes computationally unsolvable with fewer rounds of transformation. Such a design allows to increase resistance of the transformations without decrease in performance during both, software and hardware implementation.

The round operation of the mixing *MixState* state in 3DOZEN + algorithm is proposed to implement based on the LFSR constructed on the Galois scheme not on the Fibonacci scheme as was done in [8]. In the LFSR constructed on the Fibonacci scheme at each cycle the contents of only one memory cell changes. In the LFSR constructed on the Galois scheme with an appropriate choice of the characteristic polynomial at each cycle the contents of all memory elements changes. Therefore, the proposed embodiment of the *MixState*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

transformation has better dispersing and mixing properties.

Testing 3DOZEN + algorithm by the NIST technique [11, 12] showed the statistical security of it.

ACKNOWLEDGMENT

The publication is prepared in accordance with the scientific research under the Agreement between the Federal State Autonomous Educational Institution of Higher Education "National Research Nuclear University MEPhI" and the Ministry of Education and Science № 14.578.21.0117 on 27.10.2015. The unique identifier for the applied scientific research (project) is RFMEFI57815X0117.

REFRENCES:

[1] Osmolovsky, S.A., 1991. Stochastic Methods of Data Transmission. Moscow: Radio i Svyaz.

[2] Osmolovsky, S.A., 2003. Stochastic Methods of Information Defense. Moscow: Radio i Svyaz.

[3] Three-Dimensional Pseudo-Random Number Generator for Implementating in Hybrid Computer Systems. M. A. Ivanov, N. P. Vasilyev, I. V. Chugunkov et. al. // Vestnik NRNU MEPhI, 2012, Vol. 1, № 2.

[4] Ivanov, M.A. and I.V. Chugunkov, 2012.Cryptographic Methods of Information Defense in the Computer Systems and Networks: Teaching Guide. Moscow: Na-tional Research Nuclear University MEPhI.

[5] Three-Dimensional Data Stochastic Transformation Algorithms for Hybrid Su-percomputer Implementation / Ivanov M.A., Spiridonov A.A., Chugunkov I.V., et. al. – Proceedings of 17th IEEE Mediterranean Electrotechnical Conference (MELECON), 2014, Beirut, Lebanon, pp. 451 – 457.

[6] Nakahara, Jorge Jr. 3D: A Three-Dimensional Block Cipher. Date Views 07.06.2016 infoscience.epfl.ch/record/128649/files/Nak08.pdf (07.06.2016).

[7] Keccak sponge function family. Main document. Guido Bertoni, Joan Daemen, Michael Peeters, Gilles Van Assche. Date Views 07.06.2016 keccak.noekeon.org/Keccak-main-2.1.pdf

[8] GOST R 34.12-2015. Information Technology. Cryptographic Information De-fense. Block Ciphers, 2015. Moscow: Standartinform.

[9] Boreskov A.V., Harlamov A.A. Osnovy raboty s tehnologiej CUDA. M.:DMK Press, 2011.

[10] CUDA Zone: URL http://developer.nvidia.com/category/zone/cuda-zone

[11] A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publications 800-22. Revision 1.a. April, 2010.

[12] Reducing of Memory Usage in Statistical Research into Pseudorandom Number Generators by Validation Tests. Chugunkov I.V., Kutepov S.V., Shusto-va L.I. et. al. Proceedings of The Radio-Electronic Devices and Systems for the Infocommunication Technologies (REDS-2013), Moscow, Russia, May 22-23, 2013, pp.148-152.