Far East Journal of Electronics and Communications

Published Online: August 2016
http://dx.doi.org/10.17654/EC016030573
Volume 16, Number 3, 2016, Pages 573-589 **ISSN: 0973-7006**

BLOCK SYMMETRIC CRYPTOGRAPHIC ALGORITHM BASED ON PRINCIPLES OF VARIABLE BLOCK LENGTH AND MANY-VALUED LOGIC

O. N. Zhdanov and A. V. Sokolov

Siberian State Aerospace University
(named after academician Mikhail F. Reshetnev)
Krasnoyarsk, Russian Federation

Odessa National Polytechnic University
Odessa, Ukraine

Abstract

The paper is devoted to the development of a new cryptographic algorithm, based on the principles of many-valued logic and variable block length. The results of experiments confirm the cryptographic strength of the developed algorithm in comparison with its binary analogues.

1. Introduction

We are currently experiencing a rapid increase in computing power, which facilitates the development of new concepts and methods of data encryption to protect information during transmission over open channels.

High cryptographic strength can be achieved mainly at the cost of significant complexity of non-linear relationships between the key and the cipher text, and between the plaintext and the cipher text (confusion) as well

Received: December 29, 2015; Revised: April 10, 2016; Accepted: April 19, 2016
Keywords and phrases: cryptographic algorithm, encryption, many-valued logic, variable block length.

as effective destruction of the statistics of the plaintext by the cryptographic algorithm (diffusion) [1], which often requires substantial computing costs. These facts make it necessary to develop the cryptographic algorithms used for a particular telecommunication system. For example, in systems that transmit the most important and sensitive data, often consumption of computing resources on the encryption is not significant when compared with the required level of information security. For developers of such systems, it is essential to have an arsenal of available cryptographic primitives of the highest quality.

Another characteristic of the current stage of development of information technology methods is the use of algorithms based on many-valued logic, allowing much greater freedom in the choice of transforms rather than binary analogues. So, for the values $q > 2$, there are active developments for: q-ary radio systems, the methods of synthesis of optimal q-ary signals, as well as correcting codes [2]. The problems of construction of orthogonal transforms for use in q-ary telecommunication systems are discussed in [3].

Despite so much attention of researchers to the development of q-ary data transmission techniques, the issues of construction of information security systems and developing the cryptographic algorithms that can perform encryption using the principles of many-valued logic are considered insufficiently in the literature. It should be noted that the development of methods of the synthesis of such cryptographic primitives as the three-valued logic optimal S-boxes of arbitrary length was performed in [4]. However, these are only the first steps in this direction. Development of the new non-binary cryptographic algorithm which is carried out in this paper, is actually required not only in terms of the practical application of the algorithm and the development of appropriate theoretical foundations, but also in terms of a better understanding of the mechanisms of the cryptographic systems with a base $q = 2$.

During development of the new cryptographic algorithm, we used a new concept of block symmetric encryption, based on the principle of variable

(dynamically changing at each procedure) length of the block proposed in [5].

A real and promising approach is a merging of two concepts: the many-valued logic encryption and variable length of the block into a single symmetric algorithm.

The purpose of this article is to develop a cryptographic algorithm, based on the concepts of many-valued logic encryption and variable block length.

We briefly describe the scheme of the encryption algorithm with variable block size [5]. As it is known, many modern encryption algorithms use *S*-boxes and operation of addition modulo with the round key, wherein the size of the *S*-box and its quality, that affect the performance of confusion and diffusion, are of importance.

The text block has its own structure, which to a greater or lesser extent inherits characteristics of the natural language texts or the natural data. Therefore, working with blocks of constant size in all rounds, we cannot guarantee the scattering of those properties in the cipher text, and to improve the quality of the final cryptographic transform, we must either, complicate the round operation with blocks or increase the number of rounds.

In [5], it is proposed to carry out the encoding of the text using replacement tables of different sizes in different rounds.

For the most effective organization of the encryption algorithm with dynamic resizing of cryptographic primitives, the length of input block of the algorithm must be a composite number. For example, in [5], the plaintext block has the length $L = 120 = 2^3 \cdot 3 \cdot 5$ bits. Partitioning of the plaintext block for the encryption algorithm may be performed in various ways, such as segments convenient from a computational point of view of the length of $\sigma = 6,\ 8,\ 10,\ 12,\ 15,\ 20,\ 30$ bits, wherein inside one procedure, the segment size is not changed.

In this paper, the authors developed a cryptographic algorithm for the ternary case and performed numerical experiments for the ternary case,

specifically for a block length of 240 trits (ternary digits). However, the results can be easily extended to other alphabets and block lengths.

The proposed encryption algorithm involves three basic procedures: substitution, permutation, and gamma (adapted to non-binary case). Successively described, each of these procedures, which are reversible, can therefore be used both in encryption algorithm and in decryption algorithm.

In Section 2, we describe the encryption and decryption algorithms, while in Sections 3 to 5, we describe the procedures of substitution, permutation, and gamma in the case of many-valued logic. Section 6 is devoted to the use of the proposed cryptographic algorithm data formats, while in Section 7 we present the results of mathematical modeling of the proposed cryptographic algorithm.

2. The Proposed Encryption and Decryption Algorithms

Encryption is performed iteratively, while it is possible to vary the number of rounds. The round transform consists of implementation of gamma, substitution or permutation procedures. The basic version of the encryption algorithm includes five rounds, the first round consists of the gamma and permutations procedures, while other rounds include substitution and gamma procedures. The scheme of the proposed encryption algorithm is shown in Figure 1.

Alphabet: $A = \{0, 1, ..., q-1\}$, $q > 2$. In this paper, we consider in details the case $q = 3$.

Input text: $\{x_i\}$, $i = 1, 2, ..., N$. In this paper, we consider $N = 240$.

The key is: $K = \{g_i, Q_l, E, a_i\}$, where a_i - variables of splitting which are chosen as different values for each procedure, that are the parts of the encryption round, Q_l - substitution sequences, E - permutation sequence and $\{g_i\}$ - gamma sequence.

Output text: $\{y_i\}$, $i = 1, 2, ..., N$.

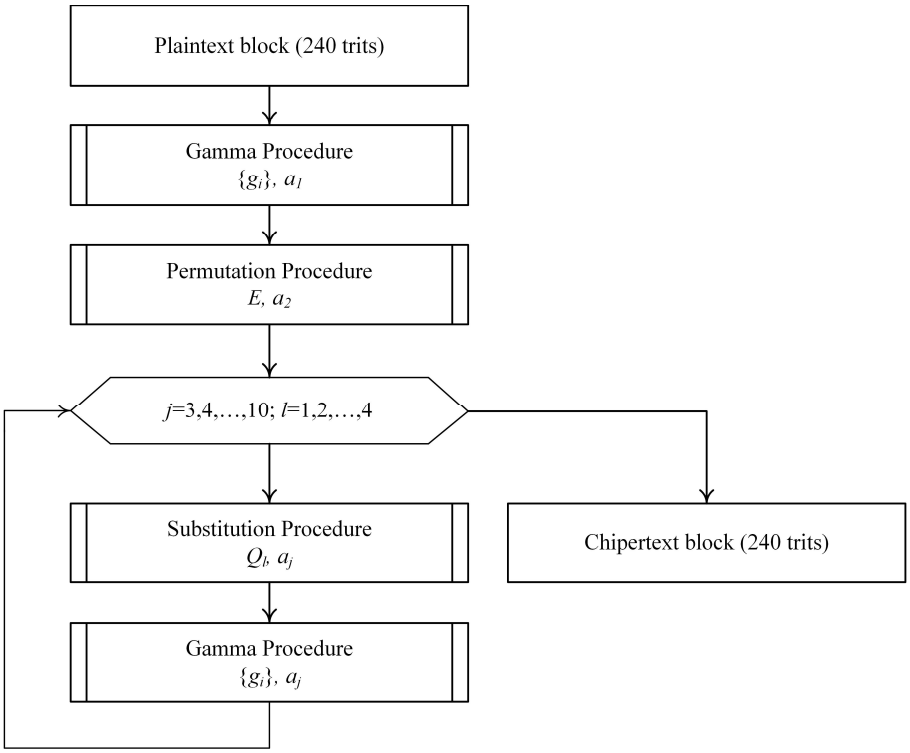

Figure 1. Proposed encryption algorithm.

For the organization of the encryption rounds, the values $\{a_i\}$ must be selected from the divisors of the number N, as well as the substitution sequences Q_1, Q_2, Q_3, Q_4 and permutation sequence E must be determined.

The decryption algorithm is the reverse order of the procedures included in the encryption algorithm. The splitting variables and the sequences of substitution and permutation are followed in reverse order. The sequences Q and a structure E, as well as gamma must be inversed as shown in Sections 3, 4 and 5. The scheme of proposed decryption algorithm is shown in Figure 2.

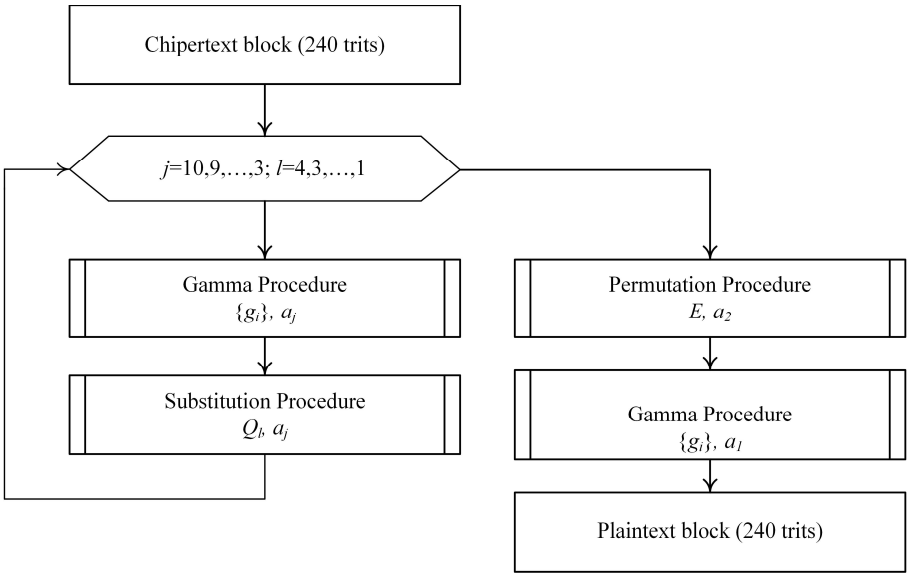

Figure 2. Proposed decryption algorithm.

3. The Procedure of Substitution

Input: A block of input code $\{x_i\}$, $i = 1, 2, ..., N$ over the alphabet $x_i \in \{0, 1, ..., q-1\}$, variable of splitting a, structure Q of the S-box of length q^a.

Output: Ciphertext block $\{y_i\}$, $i = 1, 2, ..., N$.

Denotation: For convenience, we will denote the procedure as a $y_i = S(x_i, Q, a)$, $i = 1, 2, ..., N$.

The procedure of substitution is designed to implement the concept of confusion in the cryptographic algorithm and is based on the principle of variable block length. We represent the process as a sequence of steps that will be accompanied by a particular example.

For example, let the block of ternary input text of length $N = 240$ to be

$x_i = \{220210112202212012221022121110101002112011220111121110 0$

121102111221001212021011112112122110012201101021101111 1

100202120102022201121201002110211110000010221112102101 0

022100120001111111010112022111111221121122111011020000 1

121022101111110011 11} (1)

and the value of the variable of splitting $a = 4$, as well as the optimal Q-sequence, proposed in [4] of length 3^a:

S = {0 80 37 77 43 6 40 3 74 8 73 36 79 42 5 39 2 76 1 72 44 78 41 7 38

4 75 27 26 64 23 70 33 67 30 20 35 19 63 25 69 32 66 29 22 28 18

71 24 68 34 65 31 21 54 53 10 50 16 60 13 57 47 62 46 9 52 15 59

12 56 49 55 45 17 51 14 61 11 58 48}. (2)

Step 1. The block of input text is split into $\lambda = N/a$ segments, each of length a. Thus, we obtain a two-dimensional array Λ, which is conveniently represented in the form of a matrix of size $\lambda \times a$, each line of which represents a segment of the input text of the length a.

For our example, we split the input data block to $\lambda = N/a = 240/4 = 60$ segments, which can be represented as a matrix of size 60×4, in which every row for clarity is taken in braces, and the numbers of segments are placed beside (rows of the matrix Λ)

1. {2202}	11. {1122}	21. {1211}	31. {0202}	41. {2101}	51. {1121}
2. {1011}	12. {0111}	22. {2122}	32. {2201}	42. {0022}	52. {1221}
3. {2202}	13. {1211}	23. {1100}	33. {1212}	43. {1001}	53. {1101}
4. {2120}	14. {1001}	24. {1220}	34. {0100}	44. {2000}	54. {1020}
5. {1222}	15. {2110}	25. {1101}	35. {2110}	45. {1111}	55. {0001}
6. {1022}	16. {2111}	26. {0211}	36. {2111}	46. {1110}	56. {1210}
7. {1211}	17. {2210}	27. {0111}	37. {1000}	47. {1011}	57. {2210}
8. {1010}	18. {0121}	28. {1110}	38. {0010}	48. {2022}	58. {1111}
9. {1002}	19. {2021}	29. {0202}	39. {2211}	49. {1111}	59. {1100}
10. {1120}	20. {0111}	30. {1201}	40. {1210}	50. {1122}	60. {1111}

(3)

Step 2. Perform substitution in each segment of length a, according to a rule determined by Q-sequence, whereby we obtain a new matrix $\Omega_j = Q(\Lambda_j)$ of size $\lambda \times a$, wherein $j = 1, 2, ..., \lambda$ is the row index.

For our example, we obtain a new matrix of size 60×4:

1. {0122}	11. {0211}	21. {2112}	31. {1122}	41. {1201}	51. {1002}
2. {2121}	12. {1120}	22. {1211}	32. {1200}	42. {2202}	52. {1011}
3. {0122}	13. {2112}	23. {1022}	33. {1021}	43. {0222}	53. {0201}
4. {0110}	14. {0222}	24. {2102}	34. {0022}	44. {2000}	54. {2111}
5. {0210}	15. {1221}	25. {0201}	35. {1221}	45. {2120}	55. {2222}
6. {0202}	16. {0120}	26. {1112}	36. {0120}	46. {0221}	56. {0220}
7. {2112}	17. {1220}	27. {1120}	37. {1000}	47. {2121}	57. {1220}
8. {0212}	18. {0002}	28. {0221}	38. {2212}	48. {1202}	58. {2120}
9. {2101}	19. {2010}	29. {1122}	39. {0112}	49. {2120}	59. {1022}
10. {2110}	20. {1120}	30. {0200}	40. {0220}	50. {0211}	60. {2120}

(4)

Step 3. Concatenating rows of matrix Ω, we get a sequence of output elements $\{y_i\}$, $i = 1, 2, ..., N$ in result

$$y_i = \{012221210122011002100202211202122101211002111202112022212210120122000022010112021121211102221020201111211200221112202001122120010210022122101201000221201120220120122020222200021200221212112022120021110021011020121112222022012022021201022212 0\}. \quad (5)$$

It should be noted that the statistics of the input text as a result of the substitution procedure has changed. For example, the input text contains 63 characters “0”, 114 characters “1” and 63 characters “2”, while after all the changes the output text contains 67, 77, 96 characters “0”, “1” and “2”, respectively. Thus, we can note the tendency for the approach of the text to uniform statistics.

Note also that the Q-sequences may be considered as an element of (long-term) key, and to be kept secret.

The reverse substitution procedure is identical to the substitution procedure, except that if the sequence is not an involution for the inverse permutation sequence, it has to be found as an inverse permutation Q^{-1} of the permutation sequence Q.

4. The Procedure of Permutation

Input: A block of input code $\{x_i\}$, $i = 1, 2, ..., N$ over the alphabet $x_i \in \{0, 1, ..., q-1\}$, variable of splitting a, structure of permutation E of length a.

Output: Ciphertext block $\{y_i\}$, $i = 1, 2, ..., N$.

Denotation: For convenience, we will denote the procedure as a $y_i = P(x_i, E, a)$, $i = 1, 2, ..., N$.

The permutation procedure is designed to implement the concept of diffusion in encryption. In this algorithm, we propose to make a random choice of structure of E-sequence, which determines the P-box, then send it to the receiving side and use it as a long-term element of key information.

Let us describe the process of permutation in a sequence of steps for clarity followed by a specific example. We will specify the block of input data (1), the value of the splitting variable $a = 30$, as well as a random permutation

$$E = \{5\ 8\ 21\ 11\ 10\ 19\ 20\ 24\ 13\ 29\ 30\ 22\ 14$$
$$17\ 15\ 16\ 7\ 2\ 3\ 12\ 18\ 26\ 25\ 9\ 23\ 27\ 28\ 4\ 6\ 1\}. \quad (6)$$

Step 1. The block of input text is split into $\lambda = N/a$ segments, each of length a. Thus, we obtain a two-dimensional array Λ, of size $\lambda \times a$, each row of which represents a segment of the input text of length a.

For our example, we divide the input data block to $\lambda = N/30 = 240/30 = 8$ segments, which can be represented as a matrix of size 8×30:

$$\Lambda = \begin{bmatrix} 22021011220221201222102212111 0 \\ 10100211201122011112111001211 0 \\ 21112210012120210111121121221 1 \\ 00122011010211011111100202120 1 \\ 02022201121201002110211110000 0 \\ 10221112102101002210012000111 1 \\ 11101011202211111122112112211 1 \\ 01102000011210221011111100111 1 \end{bmatrix}. \quad (7)$$

Step 2. Make a permutation of the columns of the matrix Λ in accordance with the rules defined by the permutation E, resulting in a new matrix $\Omega_{i,\,j} = \Lambda_{i,\,E_j}$, $i = 1, 2, ..., \lambda$, $j = 1, 2, ..., a$.

For our example, we obtain a new matrix

$$\Omega = \begin{bmatrix} 11102222210011201202221221120 2 \\ 01110120210121011011102121021 \\ 20121111211200211111112012212 2 \\ 21101112101011011012120001220 0 \\ 21212101000112000202101110022 0 \\ 12020100011112001021200121121 1 \\ 11120221111111111112121222100 1 \\ 20111111111101220112000011100 0 \end{bmatrix}. \quad (8)$$

Step 3. Perform concatenation of the rows of matrix, in result of which we obtain the sequence of elements of the output text $\{y_i\}$, $i = 1, 2, ..., N$.

For our example, we obtain the output sequence

$$\begin{aligned} \{y_i\} = \{&11102222210011201202221221120201110120210121011011102 \\ &12102120121111211200211111112012212222110111210101101 10 \\ &12120001220021212101000112000202101110022012020100011 1 \\ &12001021200121121111120221111111111112121222100120111 1 \\ &11111101220112000011100 0\}, \end{aligned} \quad (9)$$

the statistics of which coincides with the statistics of the input text.

We note that the random selection of a permutation E assumes use of weak permutations. However, for sufficiently large values of the variable of splitting, this probability becomes negligible.

The procedure of reverse permutation is identical to the procedure of permutation, except the fact that if the E-sequence is not an involution, then for the procedure of inverse permutation, the inverse sequence E^{-1} for the sequence E has to be found.

5. The Gamma Procedure

Input: A block of input code $\{x_i\}$, $i = 1, 2, ..., N$ over the alphabet $x_i \in \{0, 1, ..., q-1\}$, gamma $\{g_i\}$ over the alphabet $g_i \in \{0, 1, ..., q-1\}$, splitting variable a.

Output: Ciphertext block $\{y_i\}$, $i = 1, 2, ..., N$.

Denotation: For the convenience, we will denote the procedure as $y_i = \Gamma(x_i, g_i, a)$, $i = 1, 2, ..., N$.

We will describe the gamma procedure in the form of steps for clarity, followed by an example. We will specify the block of input data (1), the value of the splitting variable $a = 40$, and gamma

$$\begin{aligned}\{g_i\} = \{&2102220111011110022200010200121010121010112212012211112\\&0112112112111121112121201110111101101211020112112112 02\\&0111112002102102220111011110022200010200121010121010 11\\&2212012211120112112112111121112121201110111101101211 02\\&011211211202011111200210\}.\end{aligned} \quad (10)$$

Step 1. The block of input text is split into $\lambda = N/a$ segments, each of length a. Thus, we obtain a two-dimensional array Λ of size $\lambda \times a$ each row of which represents the segment of input text of size a. For our example, we split the source data block to $\lambda = 240/40 = 6$ segments, which can be represented as a matrix Λ of size 6×40:

$$\Lambda = \begin{bmatrix} 220210112202212012221022121110101002112 0 \\ 112201111211100121102111221001212021011 1 \\ 121121221100122011010211011111100202120 1 \\ 020222011212010021102111100000102211121 0 \\ 210100221001200011111110101120221111112 2 \\ 112112211101102000011210221011111100111 1 \end{bmatrix}. \quad (11)$$

Step 2. Similar to Step 1 (splitting scheme), we present the gamma sequence in the form of a matrix G of size $\lambda \times a$. For our example,

$$G = \begin{bmatrix} 210222011101111002220001020012101012101 0 \\ 112212012211120112112112111121112121201 1 \\ 101111011012110201121121120201111120021 0 \\ 210222011101111002220001020012101012101 0 \\ 112212012211120112112112111121112121201 1 \\ 101111011012110201121121120201111120021 0 \end{bmatrix}. \quad (12)$$

Step 3. Considering each row of the matrices Λ and G as a-digit number represented in q-ary notation, we perform line-addition of matrices Λ and G modulo q^a. Thus, during adding of the elements of the rows of matrices Λ and G, all the carry-overs, excluding the last one, must be considered, whilst the last one must be discarded.

In our example, the rows are considered as 40-trit numbers in a ternary notation (modulo 3^{40}). Thus, we can calculate the new matrix as the sum of the presented matrices Λ and G:

$$\begin{bmatrix} 201202201011100022211100211122202021220 0 \\ 002120201122221010222001102200101212212 2 \\ 000010002120002212202102202012212020222 1 1 \\ 001221100020121101022112120012211000222 0 \\ 100020010220020201000222220012111010021 0 \\ 221000222120212201210102111212222220202 1 \end{bmatrix}. \quad (13)$$

Step 4. Concatenating rows of the matrix calculated in Step 3, we obtain a sequence of elements of the output text $\{y_i\}$, $i = 1, 2, ..., N$. For our example, we obtain the output sequence

$$y_i = \{20120220101110002221110021112220202122000021202011222210102220011022001012122122000010002120002212202102202012221202221110012211000201211010221121200122110002220100200102200202010002222200121110100210221000222120212201212101021112122222202021\}. \quad (14)$$

Note that for the decryption, the reverse gamma sequence $\{g_i\}^{-1}$ must be used in the gamma procedure. Reverse gamma sequence $\{g_i\}^{-1}$ will have a different structure for different values of a splitting variable a. It is convenient to calculate the inverse gamma sequence for the matrix G. Under the rules of many-valued arithmetic for each row of the matrix must be found a q-ary complement [7], for which it is necessary to perform the subtraction of each row from number $\overbrace{2222\ldots2}^{a}$ and then add the 1.

For example, we find the first row of the matrix G^{-1}:

$$\begin{array}{l} - \underline{22222222222222222222222222222222222222} \\ \underline{21022201110111100222000102001210101210 10} \\ + \underline{01200021112111122000222120221010121012121 2} \\ \underline{00000000000000000000000000000000000000000001} \\ 01200021112111122000222120221010121012201 20 \end{array}. \quad (15)$$

Similarly, we can find the inverse gamma sequence $\{g_i\}^{-1}$.

6. Data Formats

The developed encryption algorithm works for an arbitrary alphabet $\{0, 1, ..., q-1\}$, for any prime q. For example, it is shown to work for the parameter $q = 3$. Excluding from consideration the special

telecommunication and data processing systems, operating on the principles of many-valued logic, it should be noted that most of the information that has been accumulated and processed by human, are represented in binary form.

Nevertheless, the advantages of many-valued logic encryption can be used and in the case of processing of binary information. In this case, our researches showed that it is possible to achieve high quality even in the encryption in "Electronic Codebook" mode, while the proposed algorithm in other cryptographic modes can achieve the highest level of protection.

To process the binary information with the proposed algorithm, for example, with a base $q = 3$ before encrypting, it must be represented as a ternary vector. Unfortunately, the single concept of a transfer of binary vectors to ternary has not yet been developed. However, there are several effective approaches.

The ternary computer "Setun" used an approach of the representation of data in trytes. In this way, each binary byte of data consisting of 8 bits is converted to ternary tryte, consisting of five trits. For example, byte $\{01011001\}_2 = \{12201\}_3$ [8].

It is clear that the inverse mapping from tryte to byte in the general case does not exist, since to represent a tryte in a binary system, we must have 10 binary bits or 1.25 of bytes.

Another disadvantage of this approach is that the probability of occurrence of characters of alphabet in the cells of tryte after conversion from byte is different and not equal to $1/3$. However, experiments show that the developed encryption algorithm even at the second round performs complete destruction of the statistics of the input text, which has redundancy.

7. An Example of Encryption and Cryptographic Strength of the Developed Algorithm

Experiments have shown that the proposed encryption algorithm can be successfully implemented in the modern high-level programming languages,

such as Matlab, which allows the encryption/decryption of data on a binary computer.

The most obvious and revealing (though not exhaustive) test of block cryptographic algorithm is the encryption of graphic information. For this experiment, the image displayed in grayscale was chosen (Figure 3), so every pixel is coded by a byte of data. The image size is 450 × 300 pixels, so the volume of the image in memory is 135000 bytes (131.8KB).

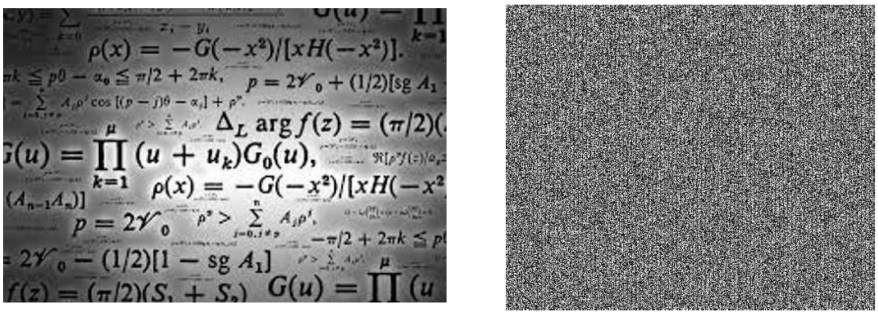

Figure 3. Initial and encrypted images.

Bytes of the image were transferred to trytes and further encrypted by the proposed encryption algorithm (Section 2), wherein we used specified split variables to encrypt

$$\begin{array}{cccccccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \\ 16 & 30 & 5 & 20 & 4 & 12 & 6 & 15 & 3 & 16 \end{array}, \tag{16}$$

substitutions from [4], as well as randomly selected permutation of length 30.

For clarity, resultant trytes were transferred again to binary representation, resulting in a new image, which is equivalent to the original encrypted, of the size 450 × 375 pixels, respectively, 168750 bytes (164.8KB).

We note in particular that the encryption was done in a mode of "Electronic Codebook", nonetheless the structure of the original image was completely destroyed, which is a very strong result for a symmetric block cryptographic algorithm.

The important observation is the change of the size of image. The fact that the construction of any modern cryptographic algorithm is based on strict observance of Kerckhoffs' principles, one of which requires that the length of the data after encryption must be equal to the length of the data before encryption. In our example, the number of trytes after encryption is strictly equal to the number of trytes before encryption and image resizing is caused only by the acquisition of redundancy by the transition from binary to ternary, which occurred before the encryption and is associated with the algorithm of transition between data formats and is not related to the encryption algorithm itself. Moreover, after the transition to ternary format and prior to the encrypting, or after encrypting to the image can be applied many-valued correcting codes, and then encoded and encrypted messages may be transmitted with the help of many-valued signal constructions without transition to binary data representation. Thus, Kerckhoffs' principles are fully implemented for the developed encryption algorithm.

Thus, even if we put splitting variables, *Q*-sequences structure and *E*-sequence as unclassified information, and all the privacy is focused in the key, the number of levels of protection of the designed cryptographic algorithm reaches astronomical magnitude $\Psi = 3^{240} = 3{,}2292 \cdot 10^{114}$, and the internal complexity and nonlinearity of the nodes of cryptographic algorithm will make the task of the cryptanalysis extremely unattractive.

8. Conclusion

A many-valued logic data encryption algorithm based on the concept of a variable block length is proposed. The algorithm is based on three basic procedures: substitution, permutation and gamma, which determine its simple and clear structure and ease of implementation. Proposed cryptographic algorithm reaches astronomical magnitude of levels of protection $\Psi = 3.2292 \cdot 10^{114}$ and has the ability to achieve good results in the hiding of encrypted information, even in an "Electronic Codebook" mode, which is confirmed by experiments.

Acknowledgement

The authors thank the anonymous referees for their valuable suggestions which led to the improvement of the manuscript.

References

[1] C. E. Shannon, A Mathematical Theory of Cryptography, Bell System Technical Memo, 1 September 1945, p. 114.

[2] Fang Zhenxian and Liu Ying, Ternary error correcting codes, Journal of Electronics and Information Technology 17 (1995), 182-186.

[3] B. J. Falkowski and S. Yan, Application of sign Hadamard-Haar transform in ternary communication system, International Journal of Electronics 79(5) (1995), 551-559.

[4] O. N. Zhdanov and A. V. Sokolov, Algorithm of construction of optimal according to criterion of zero correlation nonbinary *S*-boxes (original text in Russian), Problems of Physics, Mathematics and Technics 3(24) (2015), 94-97.

[5] O. N. Zhdanov and A. V. Sokolov, The encryption algorithm with variable block fragmentation (original text in Russian), Collection of Scientific Papers on the Results of International Scientific-practical Conference, Problems and Achievements in Science and Technology, Omsk, Russia, 2 (2015), pp. 153-159.

[6] Michael I. Mazurkov, Composite matrix cipher based on perfect binary arrays, Radioelectronics and Communications Systems 56(3) (2013), 133-140.

[7] Behrooz Parhami, Computer Arithmetic: Algorithms and Hardware Designs, Oxford University Press, New York, 2000, 510 pp.

[8] A. Stakhov, Brousentsov's ternary principle, Bergman's number system and ternary mirror-symmetrical arithmetic, The Computer Journal 45(2) (2002), 221-236.