Multi-Threaded Data Processing System Based on Cellular Automata*

Elena Kuleshova[1[0000-0002-8270-564X]], Anatoly Marukhlenko[2[0000-0002-3575-924X]], Vyacheslav Dobritsa[3[0000-0001-7533-3684]], and Maxim Tanygin[4[0000-0002-4099-1414]]

[1] Southwest State University, 305040, Kursk, Russia
lena.kuleshova.94@mail.ru
[2] Southwest State University, 305040, Kursk, Russia
proxy33@mail.ru
[3] Southwest State University, 305040, Kursk, Russia
dobritsa@mail.ru
[4] Southwest State University, 305040, Kursk, Russia
tanygin@yandex.ru

Abstract. The purpose of the work is to use an encryption algorithm based on cellular automata to develop a multi-threaded data processing system and study statistical performance indicators depending on the hardware component and input unit size, as well as develop recommendations for increasing the cryptographic strength of the method. A mathematical model of the encryption method using a floating window based on cellular automata is considered [1]. To study the speed of the processing of sensitive data, a variant of organizing the structure of the software module with an extended block of tuning parameters that determine the dimension of the matrix, the activation string of the bit neighborhood of the processed elements, the number of parallel calculations (threads) and the rule for expanding the boundary elements of the matrix has been developed. A method is proposed for generating a graphical dependence of the processing time on the initial parameters, the scope of which is possible both for processing individual files and continuous data streams of subscribers of a computer network. A cryptographic module has been developed that implements an encryption method based on cellular automata, a feature of which is a multithreaded mode of operation and dynamic control of the block of initial parameters. Recommendations on the installation of the neighborhood of the active elements of the matrix and the number of threads taking into account the architecture of the CPU are formulated. Experimental studies confirming the completeness and correctness of the proposed solutions were carried out.

Keywords: Cellular automaton · Data encryption · Parallel computing · Cryptography · System analysis · Information security.

* The reported study was funded by RFBR, project number 19-31-90069.

1 Introduction

The rapid development of information technology involves the continuous improvement of tools to ensure the information security of confidential data [2]. Special attention is paid to protecting information from unauthorized access using modern cryptographic methods and system analysis in distributed systems operating in real time. As a rule, software and hardware solutions for providing integrated cryptographic protection are distinguished by the complexity of integration into the local computer network, and also require support with the involvement of experts in the field of information security [3].

In this paper, a new block processing method is proposed, based on a cellular encryption algorithm with a floating window. The main difference between a cellular automaton with a floating window and a cellular automaton on a partition [4] is that there is no division into encryption blocks by even and odd lattices. Another feature is that when encrypting with a floating window, the encryption block moves sequentially throughout the text, moving one column forward, thereby each element of the text is encrypted a certain number of times (depending on the size of the encryption block), which increases the strength of cellular encryption. The practical significance of the method lies in the fact that the results can be used for research purposes when studying the methods of organizing multi-threaded calculations and ensuring information security when working with large data arrays. This method has the prospects of increasing durability while maintaining a high data processing speed through the use of parallel computing.

2 Problem Statement

Within the framework of the method under consideration, the task of protection against unauthorized access can be presented in the form of a sequence consisting of the following steps: confidential data is presented in the form of a binary matrix, the encryption block represents a floating window, the movement of which is determined by the input parameters and processing mode. As a rule, during processing (encryption), it is located in the upper left corner of the original matrix, the contents of this block are written out in a line in accordance with the route to bypass the neighborhood (neighboring bits). This sequence is replaced according to the transition function. The resulting cipher sequence is collapsed into a block in accordance with a given route and overwritten over the elements of the original matrix. Next, a shift occurs and the iterative process is repeated until the entire matrix is processed. When developing a software module, it is necessary to ensure that the file is read from the media or network interface, taking into account the hardware features of the automated system, to generate and process the initial matrix of a given width in accordance with the size of the floating window and the rules for expanding the matrix for boundary elements.

3 System Building

A feature of the proposed scheme for encrypting data streams using parallel computing based on cellular automata is the use of independent threads. With their help, there is a rational use of the computing resource of the central processor during processing on one computer or distribution of the processing process using network terminals [5–7]. Fig. 1 shows a basic variant of the interaction of the elements of the designed system. The information storage device stores both the source file and the encrypted file generated during the encryption process.

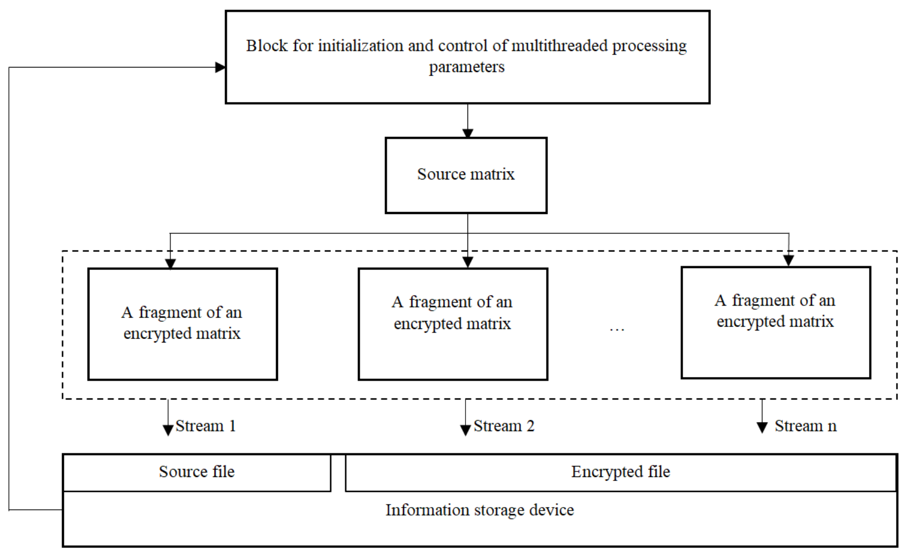

Fig. 1. The scheme of multithreaded processing.

The encrypted matrix is formed in blocks, in the process of moving a floating window and in the case of segmentation can be represented by a set of files. This means that the finished blocks of the encrypted file are stored in random access memory and wait for the end of the encryption process [8]. Depending on the storage mode of the resulting matrix, it makes sense to use several data storage devices, as while recording by several streams simultaneously, the information storage device becomes a weak link in the performance chain [9].

4 Mathematical Model

To clearly describe the operation of the method, we consider working with a two-dimensional matrix. The size of the encryption block (m_1, m_2) can be set arbitrarily, and the number of matrix columns is determined by the number of blocks written in the alphabet $A = \{0, 1\}$. The number of matrix rows is determined by the size of the initial data, and in the case of a network stream, it depends on the interaction session of the computing subscribers network [10]. The number of columns N_2 depends on the length of the source information by the formula 1:

$$N_2 = qm_2 + 1, \tag{1}$$

where q is the partial quotient in the equality $T = kq + r, 0 \leq r < k, k = m_1 m_2$ is a block area, T is the length of the source text, and r is the remainder of the division.

A cellular automaton with a floating window is called a combination (formula 2):

$$CA_o =< Z^n, (N_1, \ldots, N_n), A, (m_1, \ldots, m_n), \psi, L >, \tag{2}$$

where: Z^n is a dimension of a cellular automaton($n = 1, 2, 3$); $(N_1, \ldots, N_n)$ is a table size; $(m_1, \ldots, m_n)$ is an encryption block size; ψ is a transition function table; L is a bypass route of the encryption block of a cellular automaton with a floating window, moreover, the equalities $N_1 = m_1, \ldots, N_{n-1} = m_{n-1}$ and $N_n = qm_n + 1$, where q is a partial quotient in equality $T = kq + r, 0 \leq r < k, k = m_1 \ldots m_n$ is a number of cells in the encryption block, is a length of the source text, r is a remainder of the division.

The source text is written sequentially in layers into the source text table. In the last layer, only r cells will be filled. The remaining cells are filled with either zeros or ones. The encryption process is as follows: the encryption block is located at the beginning of the table with the source text, the contents of this block are written out in a line in accordance with the traversal route L. This sequence is replaced in accordance with the transition function ψ . The resulting cipher sequence is collapsed into a block in accordance with route L. The original block is replaced with the received one. The encryption block is shifted by one position in the data table and the process is repeated. The encryption process ends when the encryption unit is not able to move to a new position. During decryption, the floating window moves in the opposite direction, starting from the last column (the columns in the transition function are swapped).

5 Cryptographic Strength

In contrast to the cellular automaton on the partition, in which it is proposed to bypass the encryption block from the first element and go in the order of rows and columns respectively, in the cellular automaton with a floating window, it is possible to bypass the encryption block in any sequence. Let the encryption block size be $M * N$, then the number of bypass options for this block will be $(M * N)!$. The table of transition functions depends on the size of the encryption block:as the block size increases, the number of options for filling the right side of the table of transition functions increases exponentially, and therefore, the task of breaking the cipher is complicated. But it must be taken into account that an increase in the encryption block is possible only when encrypting large messages. A 5×5 block size is recommended, but the use of 6×6 and 7×7 blocks will also be successful in improving the hardware component.

Consider the rule space in more detail. In the proposed program module, a neighborhood option is used that includes a central (processed) cell and 8 elements from the environment. From the point of view of evaluating cryptographic stability, only environmental elements of the treated cell are considered, since the central cell does not affect the resistance. Thus, a neighborhood with four neighbors there are $2^8 = 256$ positions and two options for filling each cell (zero or one), then the rules table can be filled in 2^{256} various ways.

To increase the cryptographic strength, we can consider neighborhoods of the Jth order, where $J \in N$ [11]. We denote by R the number of cells in the neighborhood of Moore of the Jth order, and $R = (2J+1)^2 - 1$ then there exists $R!$ various workarounds. Similarly, denoting by S the number of cells in the Von Neumann neighborhood of the Jth order, and $S = ((2J-1)^2 - 1) + 4$ we will have $S!$ various workarounds. Obviously, the key length will vary and equal to R characters for the Jth order Moore neighborhood and S characters for the Jth order Von Neumann neighborhood. Thus, to increase the strength of cellular encryption, it is advisable to increase the size of the neighborhood and and use a function that defines the set of processed elements. It is important to consider that increasing the neighborhood leads to an increase in file size, which is especially evident in small amounts of data.

It is also worth noting that in conventional cellular encryption, the key is applied once. However, when encrypting with a floating window, the encryption block moves sequentially through the text, moving one column forward, thereby each element of the text is encrypted x times (depending on the size of the encryption block). On this basis, for the neighborhoods of Moore and Von Neumann of the Jth order, the number of different bypass options will be $(R!)^x$ and $(S!)^x$, respectively.

In the course of cryptanalysis, it was found that in order to increase the stability of the method, the matrix neighborhood should be expanded along with the rule of supplementing it based on a function that determines the state of additional cells for boundary elements [12]. It is also advisable to use a hash function that determines the sequence of processing blocks. These functions are key and should be known to the recipient of confidential data [13].

6 Simulation

The developed software module is shown in Fig. 2. Here, in the upper part, a block of initial parameters is shown - the rule of matrix addition, optional log file maintenance, the bypass rule and block size, matrix dimension (by the number of blocks in a row). The left column shows the status of the encrypted blocks in accordance with the hash function. In addition to using a conversion table based on a hash function, it is possible to load a user-specific transition guide or generate it. In order to eliminate the error of using an incorrect directory, the size of the installed block and the power of the number of transitions are compared. Experimental studies have confirmed the absence of a dependence of the processing speed on the number of blocks in the matrix row.

A search of the size options for the source blocks showed that the processing speed of the matrix is inversely proportional to the size of the block. The obtained processing statistics are shown in Table 1, here 100 percent corresponds to the longest encryption time of a file of 5 MB in size, which amounted to 23.2 seconds. The studies were carried out on a personal computer with a hardware configuration: CPU Intel i3 8100, HDD ST2000DL003, RAM 32 GB.

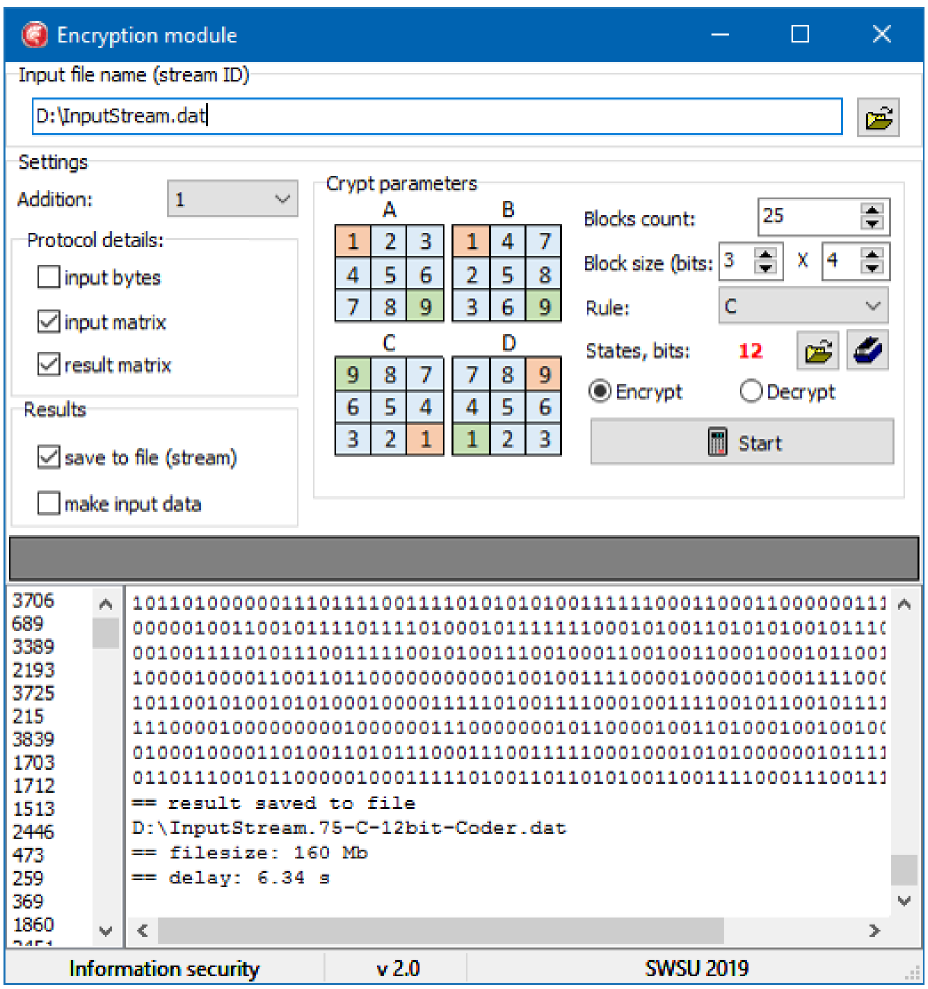

Fig. 2. The interface of the software encryption module.

The performance growth dynamics is justified by the fact that as the size of the blocks increases, their number in the file matrix decreases, and accordingly, the time for capturing, substituting, and rewriting elements decreases. It is important to take into account the fact that when using the lookup directory, processing time gains are achieved if a conversion table is available, otherwise its generation time may exceed the processing time since the time delays in its formation grow exponentially with increasing block area. To assess the dependence of the developed system performance on the number of parallel computing, the mode of enumerating the number of threads on files of various sizes was activated. In the formation of the initial matrix, blocks of 20 bits (block 4×5) were used (see Fig. 3). For clarity, the relative delay is indicated on the abscissa axis, 100 percent corresponds to the maximum processing time for a file of the specified size. Note that matching relative delay values for different data does not mean matching processing time.

Table 1. Dependence of encryption time on the size of the encryption block.

Block Dimension (bit)	2×2	3×3	4×4	5×7	7×9
Relative Delay (percent)	100	72.6	61.15	39.01	22.12

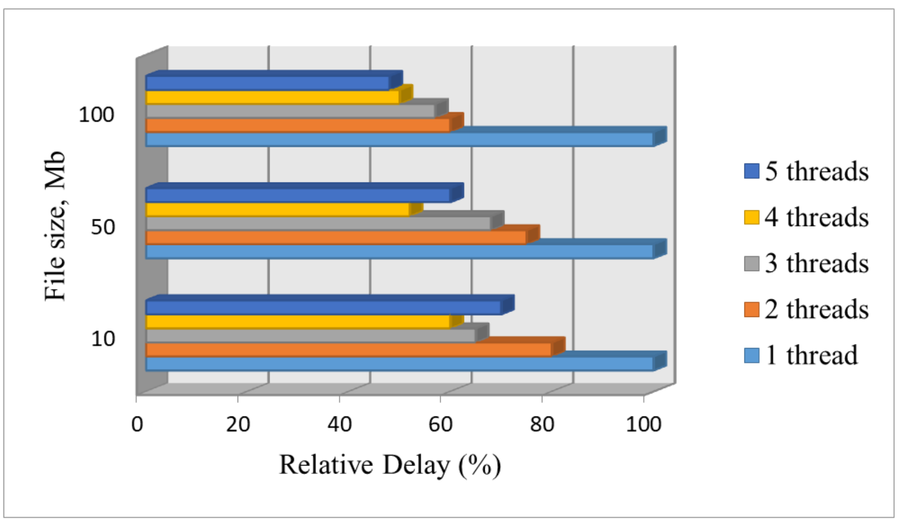

Fig. 3. Research results in multithreaded mode.

From the graphs presented it follows that on the basis of experimental equipment it is advisable to use no more than four threads (the processor has four physical cores), since the vast majority of tests showed the highest system performance. Using more than five threads is not of interest since showed a decrease in encryption speed due to the fact that there is a combination of tasks within one physical (logical) processor and its work slows down. Thus, the dynamics of performance depends on the architecture of the processor and the workload of the system as a whole. It also follows from the graph that the dependence of the increase in the speed of multi-threaded conversion depends on the file size. This is due to the fact that the initialization and start-up of threads can take up a significant share of the total processing time, which is impractical when processing files less than 10 MB. Also note the block size as it determines the number of rewrites of the elements of the original matrix, and in the case of a sequential algorithm, the time of "downtime" [14]. Analysis of computer resource loading showed that the data link is the weak link in the speed chain, because it is actively used at the time of downloading the encryption results from the random access memory. An option to solve this problem is asynchronous recording of processed segments of the original matrix of confidential data to independent information storage devices [15].

7 Results Analysis

To analyze the distribution of bits changed as a result of encryption, we form a matrix, which is the difference between the original and encrypted matrix. Figure 4 shows a fragment of the resulting table containing 100 columns. The value of the matrix element will be the values $\{-1, 0, 1\}$. White cells correspond to 0 (the value has not changed), horizontal hatching corresponds to -1, vertical hatching corresponds to 1.

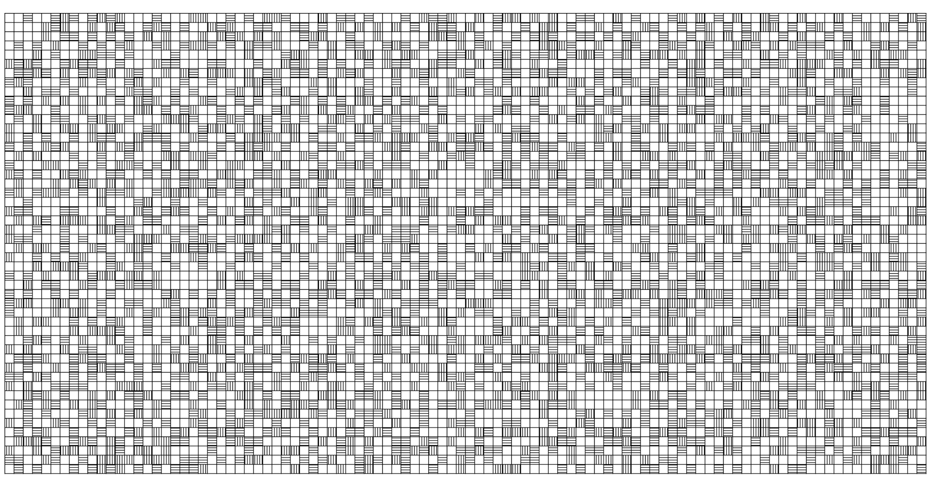

Fig. 4. Fragment of a superposition of the original and processed matrices.

A simplified visualization in the form of a surface is shown in Figure 5. For clarity, we took the distortion value modulo. Here, the difference points rise from the main level and demonstrate the distribution of the changes made. The diagram shows fifty rows corresponding to the rows of the matrix, the axis of a hundred divisions corresponds to the number of columns of the matrix (table width). Inclined faces show a uniform transition between states.

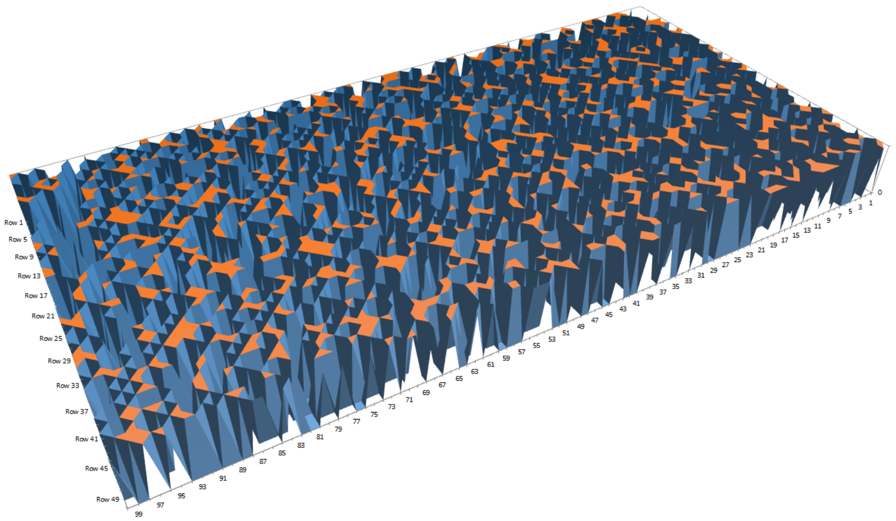

Fig. 5. Fragment of a superposition of matrices in the form of a surface.

A look at the graph “from below” (horizontal projection) coincides with Figure 4. From the results it follows that the processed matrix contains 50 percent of the changes at the bit level and 100 percent of the changes at the byte level, i.e. the result of processing is not similar to inversion and cannot be understood by an attacker without a reverse transformation, which involves knowledge or selection of key parameters. The obtained indicators confirm the high level of cryptographic strength of the encryption method under consideration based on cellular automata. In the course of further studies, it is planned to consider the changes made not only with the establishment of the fact of a change in the bit value, but also taking into account the boundary values.

In the initial definition of a cellular automaton with a floating window, processing (encryption) begins with the first block (depends on the input parameters and processing mode), then the iterative process is repeated in order until all blocks are processed. The introduction of a local rule for processing blocks will reduce the encryption time without losing cryptographic strength due to the fact that not all blocks will be processed, but only those with neighbors that correspond to certain markers. To do this, in the above definition of a cellular automaton with a floating window, we introduce a new element: M - a set of markers (boundary values).

8 Cellular Automaton with a Floating Window and a Set of Markers

The cell of the matrix n will be considered as a block. A rule is introduced for the local processing of blocks (matrix cells) based on markers - M. The marker consists of a finite set of patterns P. Each pattern, in fact, is a set of values that must be present around the cell so that its state is updated in accordance with the ψ, i.e. the update function works with a cell if and only if there is a correspondence between the states of its neighbors and the pattern in M. However, since there is no order in which each pattern P is compared with the states of the cell's neighbors, no pattern should be a subpattern of the other. Based on the studies presented in [16], we introduce two serious limitations: all markers in the composition have the same neighborhood as the final result of their composition, and all templates have the same shape. These simple constraints allow you to create a fairly simple and efficient implementation.

A cellular automaton with a floating window and a set of markers is called a combination (formula 3):

$$CA_{OM} =< Z^n, (N_1, \dots, N_n), A, (m_1, \dots, m_n), \psi, L, M >, \tag{3}$$

where: Z^n is a dimension of a cellular automaton($n = 1, 2, 3$); $(N_1, \dots, N_n)$ is a table size; $(m_1, \dots, m_n)$ is an encryption block size; ψ is a transition function table; L is a bypass route of the encryption block of a cellular automaton with a floating window. M is a set of markers consisting of patterns P, all patterns $P \in M$ have the same shape, which is determined by the neighborhood of the marker. The update function is applied to cell n if and only if the state of its neighboring cells corresponds to to some element $P \in M$.

9 Conclusion

In the course of the work, a mathematical model of the encryption method using a floating window based on cellular automata is considered. To study the speed of the processing of confidential data, a variant of organizing the structure of the software module was developed. A method is proposed for generating a graphical dependence of the processing time on the initial parameters, the scope of which is possible both for processing individual files and for continuous data streams of subscribers of a computer network, as well as at the level of a computing cluster that provides end-to-end encryption at the level of an external service.

A cryptographic module has been developed that implements the encryption method based on cellular automata, a feature of which is a multi-threaded mode of operation and dynamic control of the block of initial parameters. Recommendations are formulated for setting the neighborhood of the active elements of the matrix and the number of threads, taking into account the architecture of the central processor. Experimental studies have been carried out confirming the completeness and correctness of the proposed solutions. The expediency of using high-speed hard disk drives and saving the encryption results in asynchronous segmented mode with linking the result to the working thread is shown.

The proposed version of organizing a confidential information processing system in the form of a software module, taking into account the hardware features, allows optimizing the processing speed, and compliance with the recommendations for expanding the neighborhood during block conversion makes it possible to increase the cryptographic strength of the encryption algorithm based on cellular automata with a floating window. The totality of the results confirm the completeness and correctness of the proposed solutions.

References

1. Dr. Allen Harper, Daniel Regalado, Ryan Linn, Stephen Sims, Branko Spasojevic, Linda Martinez, Michael Baucom, Chris Eagle, The late Shon Harris.: Gray Hat Hacking: The Ethical Hacker's Handbook. 1nd edn. McGraw-Hill Education (2018)
2. Marukhlenko, A.L., Mirzakhanov, P.S.: A software package for modeling the process of transmitting and processing network data streams. Bulletin of the South-West State University. Series: Management, Computing, Informatics. Medical instrumentation **2**(3), 175–180 (2012)
3. Marukhlenko, A.L., Plugatarev, A.V., Marukhlenko, L.O., Efremov, M.A.: A comprehensive assessment of the information security of an object using a mathematical model for calculating risk indicators. Bulletin of the South-West State University. Series: Management, Computing, Informatics. Medical instrumentation **4**(29), 34–40 (2018)
4. Asyutikov A. A., Dobritsa V. P., Efremov M. A., Zarubin D. M.: A cellular automaton on a partition in encryption. Information Security of Sociotechnical Systems. **1**(1), 72–79 (2017)
5. Borzov, D.B, Chesnokova, E.O., Marukhlenko, A.L., Al-Ashval, M.M.Ya.: Search device for lower estimation of placement in fully connected matrix systems with bi-directional transmission of information. Patent for invention RUS 2421805 (11.24.2008)
6. Dobritsa, V.P., Marukhlenko, A.L., Marukhlenko, L.O., Plugatarev, A.V.: A software module for assessing the cryptographic strength of symmetric encryption methods using parallel computing In: Infocommunications and space technologies: state, problems and solutions. The collection of scientific articles based on the materials of the II All-Russian scientific and practical conferenceg, pp. 33–38. SWSU, Kursk (2018)
7. Tanygin, M.O., Marukhlenko, A.L., Marukhlenko, L.O., Konoreva, E.E.: Analysis of potential vulnerabilities and modern methods of protecting multi-user resources In: Infocommunications and space technologies: state, problems and solutions. The collection of scientific articles based on the materials of the II All-Russian scientific and practical conferenceg, pp. 136–140. SWSU, Kursk (2018)

8. Tanygin, M.O., Marukhlenko, A.L., Marukhlenko, L.O., Romanov, A.N.: ATechnology and software implementation of a software module for localizing potentially dangerous objects on a graphic substrate using neural networks. In: Infocommunications and space technologies: state, problems and solutions. The collection of scientific articles based on the materials of the II All-Russian scientific and practical conferenceg, pp. 23–28. SWSU, Kursk (2018)
9. Bobyntsev, D.O., Lisitsin, L.A., Marukhlenko, A.L., Kuzheleva, S.A.: Administration of information systems: study guide. Southwest state un-t. Kursk (2019)
10. Asyutikov, A.A., Dobritsa, V.P.: Encryption with a cellular machine on a partition by the principle of a floating window In: Infocommunications and space technologies: state, problems and solutions The collection of scientific articles based on the materials of the II All-Russian scientific and practical conference, pp. 45–50. SWSU, Kursk (2018)
11. Asyutikov, A.A., Dobritsa, V.P., Zarubin, D.M., Efremov, M.A.: Improvement of a cellular automaton on the decomposition to increase the resistance In: Infocommunications and space technologies: state, problems and solutions The collection of scientific articles based on the materials of the I All-Russian scientific and practical conference, pp. 219–221. SWSU, Kursk (2017)
12. Marukhlenko, A.L., Tanygin, M.O., Efremov, M.A., Spevakov, A.G.: Security of information systems: study guide. Southwest state un-t. Kursk (2019)
13. Efremov, M.A., Khalin, Yu.A., Marukhlenko, A.L., Marukhlenko, L.O.: Development of secure enterprise systems based on client-server technology: study guide. Southwest state un-t. Kursk (2018)
14. Tanygin, M.O., Alshaya, Kh.Ya., Altukhova, V.A., Marukhlenko, A.L.: Establishing a confidence channel for exchanging data between a source and a receiver of information using the modified one-time password method. Bulletin of the South-West State University. Series: Management, Computing, Informatics. Medical instrumentation **4**(29), 63–71 (2018)
15. Marukhlenko, A.L., Seleznev, K.D., Tanygin, M.O., Marukhlenko, L.O.: Organization of a network monitoring system and an assessment of the state of information security of an object. News of Southwestern State University **23**(1), 118–129 (2019)
16. Clarridge, A., Salomaa, K.: A cryptosystem based on the composition of reversible cellular automata. In: Ionescu, A., Martın-Vide, C. (eds.) Language and Automata Theory and Applications, LNCS, vol. 5457, pp. 314–325. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00982-2-27