Information-Driven Permutation Operations for Cryptographic Transformation

Vira Babenko[1], Tetiana Myroniuk[1], Artem Lavdanskyi[1], Yaroslav Tarasenko[1], and Oleg Myroniuk[1]

[1] *Cherkasy State Technological University, 460 Shevchenko str. Cherkasy, 18006, Ukraine*

Abstract

In the work, the authors proposed one of the techniques of using information-driven permutation operations for the implementation of cryptographic data transformation. An algorithm for implementing the proposed method of cryptographic data transformation based on the use of a basic group of information-driven permutation operations has been developed. The process of cryptographic transformation of three bytes of data based on the proposed algorithm is shown by a flowchart containing information-driven permutations, a Feistel network, shift and XOR operations, and addition modulo 2. The software implementation of the developed algorithm in the high-level object-oriented programming language Python is carried out. The obtained results of the work of the created software made it possible to conduct further research and carry out a qualitative assessment of the results of cryptographic data transformation according to the proposed method of using previously synthesized information-driven permutation operations. The effectiveness of this algorithm was evaluated based on statistical testing by the NIST STS package, as well as its suitability for implementing data encryption by hardware and software based on a comparison of test results with the results of using standard encryption algorithms DES, AES, Blowfish, Kalyna, Strumok, and Linear Feedback Shift Register.

Keywords

Technique, information-driven permutation operations, basic operations, algorithm, cryptographic transformation, key, round, statistical testing.

1. Introduction

The development of new and improvement of common encryption methods that would be simple in hardware and software implementations, and at the same time provide a sufficiently high level of cryptographic strength by expanding the range of cryptographic transformation operations used, obtained by modifying the basic operations, is one of the urgent tasks of information security. The search and synthesis of modified operations for cryptographic transformation will make it possible to build algorithms using them with the best cryptographic properties, which makes this study relevant.

Software cryptographic protection tools are flexible, which gives them a special advantage over hardware ones. Mobility and ease of use explain their modern popularity and prevalence. Therefore, among the ways to improve the stability indicators of cryptographic algorithms, there are several approaches to the construction of software ciphers. The most promising for software implementation are flexible ciphers based on the use of several modifications of the encryption algorithm, ciphers with pseudo-probable key selection, and ciphers with permutation of fixed procedures and customization of transformation operations. In addition, one of the well-known ways to increase cryptographic strength is the multi-pass mode of applying the encryption algorithm.

CPITS-2024: Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2024, Kyiv, Ukraine

EMAIL: v.babenko@chdtu.edu.ua (V. Babenko); t.myroniuk@chdtu.edu.ua (T. Myroniuk); a.lavdanskyi@chdtu.edu.ua (A. Lavdanskyi); o.m.myroniuk.asp23@chdtu.edu.ua (O. Myroniuk)
ORCID: 0000-0003-2039-2841 (V. Babenko); 0000-0002-7588-1055 (T. Myroniuk); 0000-0002-1596-4123 (A. Lavdanskyi); 0000-0002-5902-8628 (Y. Tarasenko); 0009-0007-7572-6972 (O. Myroniuk)

2. Background Analysis

In our day to be secure, modern information and communication technology needs reliable encryption. That is why a large number of modern scientific publications are devoted to the development of new and improvement of existing common cryptographic algorithms. Such works include scientific publications [1–3, 12–14, 22].

In this paper [1] to improve the security effectiveness of electronic information resources, two encryption algorithms (Luna and Neptun) have been developed based on fixed lookup tables with extended bit depth and dynamic key-dependent lookup tables. Properties of random sequences formed using the proposed algorithms' encryption [1] were explored in the environment of NIST STS statistical tests.

The paper [2] proposes a method that uses substitution tables with increased capacity and randomized linear and non-linear operations. Based on this method, a new advanced block cipher was proposed and its specifications were given [2].

In this work [3] the Linear Feedback Shift Register (LFSR) is used to produce nonbinary pseudo-random key sequences. The length of the sequence has been enhanced by designing a hybrid model using LFSR and Genetic Algorithm. Achieving a length more than the maximum length of LFSR is the primary intention of this work.

It should be noted that the main block and stream symmetric encryption algorithms used in Ukraine are the Kalyna and Strumok cryptographic algorithms. For the cryptographic transformation in the Kalyna standard [4–8], the SPN structure was chosen as a high-level design of the cipher based on analytical comparison, the layer of nonlinear transformation of the cyclic function was implemented based on S-blocks, and for the implementation of the linear scattering block, multiplication by MDV was chosen matrix. This method of constructing a crypto algorithm ensures compliance with modern requirements for the level of cryptographic stability and speed compared to other international standards.

In turn, it should be noted the high statistical security of the stream symmetric encryption algorithm Strumok [9–11], which revealed certain properties of the random bit generator. The results of experimental studies of the statistical security and speed characteristics of stream ciphers show that the Strumok algorithm is the most balanced solution, it can provide the properties of a random sequence generator and give huge indicators in terms of encryption speed. It has been practically proven that the encryption speed of the Strumok algorithm on modern computer systems can reach 10–15 Gbit/s.

In works [9, 11] the results of experimental studies of statistical properties of common and modern cryptographic encryption algorithms are given. In the article [11], experimental studies of the cryptographic properties of the Strumok stream cipher were carried out by the NIST STS statistical testing methodology. In articles [9–11], a comparative analysis of the statistical security indicators of the world-famous and most widespread cryptographic algorithms (Enocoro, HC-128, HC-256, Grain, MICKEY 2, MUGI, Rabbit, Salsa20, SNOW 2.0, Sosemanuk, Trivium, AES, Strumok, Kalyna, etc.), which are standardized at the international or national level, is carried out.

The article [12] deals with an actual task for increasing the reliability of information protection systems by the creation and use of new four-bit cryptographic transformations with nonlinear Boolean functions that have the property of strict avalanche criterion. In the article [12], for the first time, there was proposed a method for obtaining inverse four-bit cryptographic transformations with the strict avalanche criterion property for balanced Boolean functions containing two logical operations (inversion and addition modulo two). This method [12] is a method of selecting the already existing basic Boolean functions from a predetermined set of balanced basic Boolean functions for direct and inverse cryptographic transformations, whereas the existing methods of searching for inverse cryptographic transformation are methods for calculating each element of the Boolean functions for the inverse cryptographic transformation.

The article [13] proposes a modification of the algorithm for calculating the reciprocal of a number presented in a finite ring, which makes it possible to reduce the number of elementary operations and thereby increase the performance of encryption algorithms.

However, the problem task of using a group of information-driven permutation operations for the implementation of cryptographic transformation has not been given attention in the development of security software. Thus, the development of methods for applying this group of operations for data encryption and decryption and the algorithmization of these processes is certainly an urgent task of research.

3. Formal Problem Statement

The usage of controlled operations opens up great opportunities for achieving the required level of cryptographic protection. Considering that the efficiency of using controlled operations increases with the number of potentially implemented modifications, since in this case the sub-block of the data being converted expands, the use of the permutation operation becomes relevant, since it has a very large number of modifications. Thus, the development of cryptographic tools based on controlled permutations is a promising direction in modern cryptography [15].

Therefore, the main objective of this work is to improve the technique of using basic information-driven permutation operations through the use of a modified algorithm of cryptographic information transformation that allows using the full set of permutation modifications.

Purpose of the work. Develop a way to apply a basic group of information-driven permutation operations and propose an algorithm for implementing cryptographic data transformation based on the use of these operations. Investigate and evaluate the suitability of using the proposed algorithm for encryption by software and hardware.

4. Materials and Methods

Let us consider in more detail the synthesis of cryptographic transformation operations based on elementary information-driven permutation functions.

In [16] it was determined that the sets of groups of 3-bit elementary cryptographic transformation functions are functions consisting of three elementary functions, the formal model of which is similar to the complexity of the addition modulo 2. This means that these sets are simple and do not require significant resources for implementation by hardware. The results obtained are presented in Table 1 [16].

To obtain the elementary functions of the cryptographic transformations given in Table 1, the bit permutation method should be used, the main task of which is to replace one bit of the elementary function with two others.

Based on the researched results, the obtained elementary functions will be called elementary functions of information-driven permutations.

In the work [16], a model of elementary functions for cryptographic transformation is built, which has the following form:

$$Y = \tilde{x}_i\tilde{x}_j \vee \bar{\tilde{x}}_i\tilde{x}_k, \qquad (1)$$

where Y is the value of the corresponding digit of the output signal of the result of the elementary functions of the cryptographic transformation; $\tilde{x}_i, \tilde{x}_j, \tilde{x}_k$ are the values of the corresponding digits of the input signal.

The following are the properties of the defined model [16]:

1. i, j, k take the values 1, 2, 3, and $i \neq j \neq k$.
2. x_i can take a direct or inverse value.
3. x_j, x_k can be both in direct and inverse meaning and $j \neq k$.

Table 1
Elementary functions of low complexity

Number of function	Execution result	Discrete model
83	01010011	$x_1 \cdot x_2 \vee \bar{x}_1 \cdot x_3$
163	10100011	$x_1 \cdot x_2 \vee \bar{x}_1 \cdot \bar{x}_3$
46	00101110	$x_1 \cdot \bar{x}_2 \vee x_2 \cdot \bar{x}_3$
71	01000111	$x_1 \cdot x_2 . \vee \bar{x}_2 \cdot x_3$
139	10001011	$x_1 \cdot x_2 \vee \bar{x}_2 \cdot \bar{x}_3$
53	00110101	$\bar{x}_1 \cdot x_2 . \vee x_1 \cdot x_3$
58	00111010	$\bar{x}_1 \cdot x_2 . \vee x_1 \cdot \bar{x}_3$
184	10111000	$\bar{x}_1 \cdot x_2 \vee \bar{x}_2 \cdot \bar{x}_3$
116	01110100	$\bar{x}_1 \cdot x_2 \vee \bar{x}_2 \cdot x_3$
92	01011100	$x_1 \cdot \bar{x}_2 \vee \bar{x}_1 \cdot x_3$
172	10101100	$x_1 \cdot \bar{x}_2 \vee \bar{x}_1 \cdot \bar{x}_3$
29	00011101	$x_1 \cdot \bar{x}_2 \vee x_2 \cdot x_3$
197	11000101	$\bar{x}_1 \cdot \bar{x}_2 \vee x_1 \cdot x_3$
202	11001010	$\bar{x}_1 \cdot \bar{x}_2 \vee x_1 \cdot \bar{x}_3$
209	11010001	$\bar{x}_1 \cdot \bar{x}_2 \vee x_2 \cdot x_3$
226	11100010	$\bar{x}_1 \cdot \bar{x}_2 \vee x_2 \cdot \bar{x}_3$
39	00100111	$x_1 \cdot x_3 \vee x_2 \cdot \bar{x}_3$
141	10001101	$x_1 \cdot x_3 \vee \bar{x}_2 \cdot \bar{x}_3$
114	01110010	$\bar{x}_1 \cdot x_3 \vee x_2 \cdot \bar{x}_3$
27	00011011	$x_1 \cdot \bar{x}_3 \vee x_2 \cdot x_3$
78	01001110	$x_1 \cdot \bar{x}_3 \vee \bar{x}_2 \cdot x_3$
177	10110001	$\bar{x}_1 \cdot \bar{x}_3 \vee x_2 \cdot x_3$
228	11100100	$\bar{x}_1 \cdot \bar{x}_3 \vee \bar{x}_2 \cdot x_3$
216	11011000	$\bar{x}_1 \cdot x_3 \vee \bar{x}_2 \cdot \bar{x}_3$

To construct a method of synthesis of information-driven elementary functions, we will introduce the following definitions [16].

Definition 1. The main element of the elementary function of information-driven permutations is the repeating element in the right and left parts of the elementary function in direct and inverse meanings.

Definition 2. An additional element or an elementary function of information-driven permutations is an element that occurs once either on the left or on the right side of the elementary function.

The method for synthesizing elementary information-driven permutation functions is as follows [16]:

1. Determine the indices of the main and additional elements of elementary functions of information-driven permutation.
2. Determine direct and inverse values of elements of elementary functions of information-driven permutation.
3. Substitute certain values into expression (1) to obtain elementary information-driven permutation functions.
4. Applying items 1–3 on this set of elementary indices for direct and inverse values of elementary functions of information-driven permutation, we obtain the complete set of elementary functions of information-driven permutation.

Subsequently, the analysis of cryptographic transformation operations [16] was carried out based on elementary permutation functions controlled by information obtained based on the experiment.

As a result of the computational experiment [16], it was determined that the total number of three-bit information-driven permutation operations for cryptographic transformation will be equal to the product of the multiplication of the number of operations in each group, which is 764 information-driven operations.

When researching cryptographic transformation operations, it was found that the total number of these operations is formed by a combination of basic operations, permutation operations, and inversion operations [16]: $N = N_b \cdot N_{\text{п}} \cdot N_{\text{і}} = N_b \cdot 3! \cdot 2^3 = 384$, where N is the total number of operations; N_b is the number of basic operations; N_n is the number of operations based on the replacement of 1 or 2 elementary group functions and N_i is the number of inversion transactions. Therefore, to determine the number of three-bit basic cryptographic transformation operations, it is necessary to calculate the number of basic three-bit operations from the total number of cryptographic transformation operations.

For further research, elementary information-driven functions were divided into direct and reverse elementary information-driven functions [16].

When studying certain groups of information-driven permutation operations, it was found that the base group can be considered a group where elementary functions are located in the following way [16]: the first elementary function of the operation is the function f_1, the value of the main element of which is x_1; the second elementary function of the operation is the function f_2, the main element of which is x_2; the third elementary function of the operation is the function f_3, where the value of the main element is x_3.

Based on this, it can be concluded that the main elements of the basic operation, which can form the basic group of cryptographic transformation operations, should be placed diagonally [16].

Thus, eight operations for two groups were defined, which are the basic operations of forward (F^f) and reverse (F^r) cryptographic transformation. They are represented by expressions (2)–(17) [16].

$$F^{f}_{92,46,27} = \begin{bmatrix} x_1 \cdot \bar{x}_2 \vee \bar{x}_1 \cdot x_3 \\ x_1 \cdot \bar{x}_2 \vee x_2 \cdot \bar{x}_3 \\ x_1 \cdot \bar{x}_3 \vee x_2 \cdot x_3 \end{bmatrix} = \\ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Leftrightarrow \begin{bmatrix} - & 0 & 1 \\ 1 & - & 0 \\ 1 & 1 & - \end{bmatrix}. \tag{2}$$

$$F^{r}_{83,116,78} = \begin{bmatrix} x_1 \cdot x_2 \vee \bar{x}_1 \cdot x_3 \\ \bar{x}_1 \cdot x_2 \vee \bar{x}_2 \cdot x_3 \\ x_1 \cdot \bar{x}_3 \vee \bar{x}_2 \cdot x_3 \end{bmatrix} = \\ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Leftrightarrow \begin{bmatrix} - & 1 & 1 \\ 0 & - & 1 \\ 1 & 0 & - \end{bmatrix}. \tag{3}$$

$$F^{f}_{53,71,27} = \begin{bmatrix} \bar{x}_1 \cdot x_2 \vee x_1 \cdot x_3 \\ x_1 \cdot x_2 \vee \bar{x}_2 \cdot x_3 \\ x_1 \cdot \bar{x}_3 \vee x_2 \cdot x_3 \end{bmatrix} = \\ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Leftrightarrow \begin{bmatrix} - & 1 & 1 \\ 1 & - & 1 \\ 1 & 1 & - \end{bmatrix}. \tag{4}$$

$$F^{r}_{83,29,39} = \begin{bmatrix} x_1 \cdot x_2 \vee \bar{x}_1 \cdot x_3 \\ x_1 \cdot \bar{x}_2 \vee x_2 \cdot x_3 \\ x_1 \cdot x_3 \vee x_2 \cdot \bar{x}_3 \end{bmatrix} = \\ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Leftrightarrow \begin{bmatrix} - & 1 & 1 \\ 1 & - & 1 \\ 1 & 1 & - \end{bmatrix}. \tag{5}$$

$$F^{f}_{83,29,39}=\begin{bmatrix}x_1\cdot x_2\vee\bar{x}_1\cdot x_3\\x_1\cdot\bar{x}_2\vee x_2\cdot x_3\\x_1\cdot x_3\vee x_2\cdot\bar{x}_3\end{bmatrix}= =\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\Leftrightarrow\begin{bmatrix}-&1&1\\1&-&1\\1&1&-\end{bmatrix}. \tag{6}$$

$$F^{r}_{53,71,27}=\begin{bmatrix}\bar{x}_1\cdot x_2\vee x_1\cdot x_3\\x_1\cdot x_2\vee\bar{x}_2\cdot x_3\\x_1\cdot\bar{x}_3\vee x_2\cdot x_3\end{bmatrix}= =\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\Leftrightarrow\begin{bmatrix}-&1&1\\1&-&1\\1&1&-\end{bmatrix}. \tag{7}$$

$$F^{f}_{58,29,78}=\begin{bmatrix}\bar{x}_1\cdot x_2\vee x_1\cdot\bar{x}_3\\x_1\cdot\bar{x}_2\vee x_2\cdot x_3\\x_1\cdot\bar{x}_3\vee\bar{x}_2\cdot x_3\end{bmatrix}= =\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\Leftrightarrow\begin{bmatrix}-&1&0\\1&-&1\\1&0&-\end{bmatrix}. \tag{8}$$

$$F^{r}_{53,46,114}=\begin{bmatrix}\bar{x}_1\cdot x_2\vee x_1\cdot x_3\\x_1\cdot\bar{x}_2\vee x_2\cdot\bar{x}_3\\\bar{x}_1\cdot x_3\vee x_2\cdot\bar{x}_3\end{bmatrix}= =\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\Leftrightarrow\begin{bmatrix}-&1&1\\1&-&0\\0&1&-\end{bmatrix}. \tag{9}$$

$$F^{f}_{58,116,39}=\begin{bmatrix}\bar{x}_1\cdot x_2\vee x_1\cdot\bar{x}_3\\\bar{x}_1\cdot x_2\vee\bar{x}_2\cdot x_3\\x_1\cdot x_3\vee x_2\cdot\bar{x}_3\end{bmatrix}= =\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\Leftrightarrow\begin{bmatrix}-&1&0\\0&-&1\\1&1&-\end{bmatrix}. \tag{10}$$

$$F^{r}_{92,71,114}=\begin{bmatrix}x_1\cdot\bar{x}_2\vee\bar{x}_1\cdot x_3\\x_1\cdot x_2\vee\bar{x}_2\cdot x_3\\\bar{x}_1\cdot x_3\vee x_2\cdot\bar{x}_3\end{bmatrix}= =\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\Leftrightarrow\begin{bmatrix}-&0&1\\1&-&1\\0&1&-\end{bmatrix}. \tag{11}$$

$$F^{f}_{53,46,114}=\begin{bmatrix}\bar{x}_1\cdot x_2\vee x_1\cdot x_3\\x_1\cdot\bar{x}_2\vee x_2\cdot\bar{x}_3\\\bar{x}_1\cdot x_3\vee x_2\cdot\bar{x}_3\end{bmatrix}= =\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\Leftrightarrow\begin{bmatrix}-&1&1\\1&-&0\\0&1&-\end{bmatrix}. \tag{12}$$

$$F^{r}_{58,29,78}=\begin{bmatrix}\bar{x}_1\cdot x_2\vee x_1\cdot\bar{x}_3\\x_1\cdot\bar{x}_2\vee x_2\cdot x_3\\x_1\cdot\bar{x}_3\vee\bar{x}_2\cdot x_3\end{bmatrix}= =\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\Leftrightarrow\begin{bmatrix}-&1&0\\1&-&1\\1&0&-\end{bmatrix}. \tag{13}$$

$$F^{f}_{92,71,114}=\begin{bmatrix}x_1\cdot\bar{x}_2\vee\bar{x}_1\cdot x_3\\x_1\cdot x_2\vee\bar{x}_2\cdot x_3\\\bar{x}_1\cdot x_3\vee x_2\cdot\bar{x}_3\end{bmatrix}= =\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\Leftrightarrow\begin{bmatrix}-&0&1\\1&-&1\\0&1&-\end{bmatrix}. \tag{14}$$

$$F^{r}_{58,116,39}=\begin{bmatrix}\bar{x}_1\cdot x_2\vee x_1\cdot\bar{x}_3\\\bar{x}_1\cdot x_2\vee\bar{x}_2\cdot x_3\\x_1\cdot x_3\vee x_2\cdot\bar{x}_3\end{bmatrix}= =\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\Leftrightarrow\begin{bmatrix}-&1&0\\0&-&1\\1&1&-\end{bmatrix}. \tag{15}$$

$$F^{f}_{83,116,78}=\begin{bmatrix}x_1\cdot x_2\vee\bar{x}_1\cdot x_3\\\bar{x}_1\cdot x_2\vee\bar{x}_2\cdot x_3\\x_1\cdot\bar{x}_3\vee\bar{x}_2\cdot x_3\end{bmatrix}= =\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\Leftrightarrow\begin{bmatrix}-&1&1\\0&-&1\\1&0&-\end{bmatrix}. \tag{16}$$

$$F^{r}_{92,46,27}=\begin{bmatrix}x_1\cdot\bar{x}_2\vee\bar{x}_1\cdot x_3\\x_1\cdot\bar{x}_2\vee x_2\cdot\bar{x}_3\\x_1\cdot\bar{x}_3\vee x_2\cdot x_3\end{bmatrix}= =\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\Leftrightarrow\begin{bmatrix}-&0&1\\1&-&0\\1&1&-\end{bmatrix}. \tag{17}$$

Having studied the basic group of cryptographic transformation operations, presented above in the form of discrete models, we have determined the following dependencies for the main elements of operations [16]:

$$\begin{gathered}F^{f}_{92,46,27}=x_1=1,x_2=0,x_3=0.\Rightarrow\\\Rightarrow F^{r}_{83,116,78}=x_1=1,x_2=1,x_3=0.\\F^{f}_{53,71,27}=x_1=0,x_2=1,x_3=0.\Rightarrow\\\Rightarrow F^{r}_{83,29,39}=x_1=1,x_2=0,x_3=1.\\F^{f}_{83,29,39}=x_1=1,x_2=0,x_3=1.\Rightarrow\\\Rightarrow F^{r}_{53,71,27}=x_1=0,x_2=1,x_3=0.\\F^{f}_{58,29,78}=x_1=0,x_2=0,x_3=0.\Rightarrow\\\Rightarrow F^{r}_{53,46,114}=x_1=0,x_2=0,x_3=1.\\F^{f}_{58,116,39}=x_1=0,x_2=1,x_3=1.\Rightarrow\\\Rightarrow F^{r}_{92,71,114}=x_1=1,x_2=1,x_3=1.\\F^{f}_{53,46,114}=x_1=0,x_2=0,x_3=1.\Rightarrow\\\Rightarrow F^{r}_{58,29,78}=x_1=0,x_2=0,x_3=0.\\F^{f}_{92,71,114}=x_1=1,x_2=1,x_3=1.\Rightarrow\\\Rightarrow F^{r}_{58,116,39}=x_1=0,x_2=1,x_3=1.\\F^{f}_{83,116,78}=x_1=1,x_2=1,x_3=0.\Rightarrow\\\Rightarrow F^{r}_{92,46,27}=x_1=1,x_2=0,x_3=0.\end{gathered}$$

Having examined the obtained dependencies, we can conclude that the diagonal values of the main elements of the cryptographic transformation operations included in the group under study form eight variants of operations, that is, 2^3 variants. Hence, it can be assumed that the basic group of cryptographic transformation operations can be formed only by those elementary operations for cryptographic transformation, in which the value of the main elements along the diagonal is equal to 2^3 options [16].

The modified matrix discrete model of the combination of the permutation matrix and the complement matrix is described as [16]:

$$F^f = \\ = \begin{bmatrix} x_{11} & \bar{x}_{11} \vee (x_{22} \oplus x_{33}) & x_{11} \vee (x_{22} \oplus x_{33}) \\ \bar{x}_{22} \vee (x_{11} \equiv x_{33}) & x_{22} & x_{22} \vee (x_{11} \equiv x_{33}) \\ \bar{x}_{33} \vee (x_{11} \oplus x_{22}) & x_{33} \vee (x_{11} \oplus x_{22}) & x_{33} \end{bmatrix} \quad (18)$$

where x_{ij} are the elementary functions of the cryptographic transformation of forward (F^f) cryptographic transformation; $i, j \in \{1,2,3\}$.

From here, a generalized discrete model of basic groups of encoding operations for cryptographic transformation is obtained, which is presented in the following form [16]:

$$F^f = \\ = \begin{bmatrix} \big(x_1 \equiv (x_{11})\big) \cdot \big(x_2 \equiv (\bar{x}_{11} \vee (x_{22} \oplus x_{33}))\big) \vee \\ \vee \big(x_1 \equiv (x_{11} \oplus 1)\big) \cdot \big(x_3 \equiv (x_{11} \vee (x_{22} \oplus x_{33}))\big) \\ \big(x_2 \equiv (x_{22})\big) \cdot \big(x_1 \equiv (\bar{x}_{22} \vee (x_{11} \equiv x_{33}))\big) \vee \\ \vee \big(x_2 \equiv (x_{22} \oplus 1)\big) \cdot \big(x_3 \equiv (x_{22} \vee (x_{11} \equiv x_{33}))\big) \\ \big(x_3 \equiv (x_{33})\big) \cdot \big(x_1 \equiv (\bar{x}_{33} \vee (x_{11} \oplus x_{22}))\big) \vee \\ \vee \big(x_3 \equiv (x_{33} \oplus 1)\big) \cdot \big(x_2 \equiv (x_{33} \vee (x_{11} \oplus x_{22}))\big) \end{bmatrix} \quad (19)$$

where x_{ij} are the elementary functions of the cryptographic transformation of reverse (F^r) cryptographic transformation; $i, j \in \{1,2,3\}$.

Having studied the obtained models of encoding and decoding functions, represented by expressions (18) and (19), we can conclude that the essence of the method of synthesis of basic cryptographic transformation operations is to change the values, which allows obtaining eight basic cryptographic transformation operations for encoding and decoding functions.

Having determined the essence of the method for synthesizing the basic operations of cryptographic transformation, we can conclude that the synthesis of cryptographic transformation operations based on the obtained discrete models is as follows [16]:

1. Synthesis of all basic operations of cryptographic transformation.
2. For each received operation, it is necessary to perform a permutation, which will increase their number by six times.
3. To increase the number of operations, it is necessary to use inversion operations, which will increase the number of transformation operations by another eight times.

The result of the computational experiment is 384 cryptographic transformation operations for three-digit elementary functions.

5. Experiments

An algorithm for a 5-round data transformation process, shown in Fig. 1, was developed to experiment to implement the proposed method of using the synthesized basic group of information-driven permutation operations to perform cryptographic data transformation.

This algorithm was implemented by us programmatically for further research and analysis of the results of its work. The development of this software tool was carried out using the high-level object-oriented programming language Python.

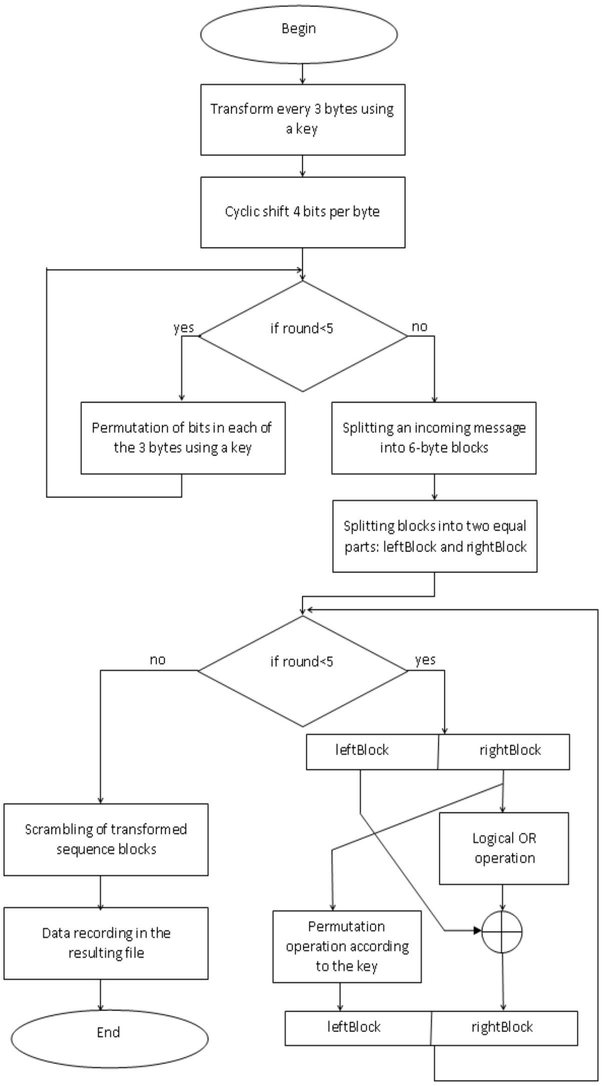
Begin
Transform every 3 bytes using a key
Cyclic shift 4 bits per byte
yes
if round<5
no
Permutation of bits in each of the 3 bytes using a key
Splitting an incoming message into 6-byte blocks
Splitting blocks into two equal parts: leftBlock and rightBlock
no
if round<5
yes
leftBlock
rightBlock
Logical OR operation
Scrambling of transformed sequence blocks
Data recording in the resulting file
Permutation operation according to the key
leftBlock
rightBlock
End

Figure 1: Algorithm of the developed software

The essence of this algorithm consists of the following. The input data stream is divided into 3-byte blocks, on which the following operations are performed: addition modulo 2, cyclic shift, and permutations.

The algorithm of the software, which was developed by us for conducting an experimental study, consists of several steps.

Step 1. For each byte of the output message sequence, we perform an XOR operation with each byte of the key. The keys are the basic operations determined as a result of the computational experiment. A new key is used for each group of three bytes.

Step 2. The next step is to apply the cyclic shift operation. At the same time, the message from the previous step is split into bytes, each of which is cyclically shifted by 4 bits.

Step 3. After performing the cyclic shift operation on the message obtained in step 2, a permutation operation by key 2 is applied to each 24 bits of information to perform dispersal of the statistical structure of the message. 5 such rounds of bit permutations in three bytes are performed. The process of implementing the permutation is shown in Fig. 2.

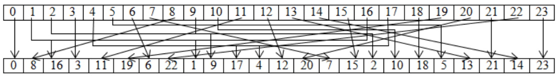

Figure 2: Bit permutation process for three bytes

Step 4. At this step, the Feistel network was applied, which is based on two main features of cryptography [15, 24–27, 30]: substitution and transposition. The sequence obtained in step 3 is split into blocks of 6 bytes. For each of the 6 bytes, the Feistel network is used, which involves splitting the message into two equal parts (three bytes each), where leftBlock is the left part and rightBlock is the right part.

An OR operation is performed for each byte of the sequence contained in rightBlock with each byte of the key. An XOR operation is performed on the result obtained after conversion with each byte contained in leftBlock according to the Feistel network algorithm.

For the value to be placed in leftBlock, the bits are permuted according to step 3.

The number of rounds of transformation using the permutation and key for step 4 is 5 rounds.

Step 5. Before data recording in the resulting file, each 24-bit block (leftBlock, rightBlock) is scrambled with a Linear Feedback Shift Register word (a 24th-degree polynomial) by performing the addition modulo 2. The feedback polynomial is $x^{24} + x^{23} + x^{21} + x^{20}$. In our case, the taps (the bit positions that affect the next state) are [20–24]. The result is recorded byte by byte to a file.

For the example shown in Fig. 2, we estimate the probability P_{crack} of cracking one round of bit permutations for step 3 of the cryptographic transformation using the formula:

$$P_{crack} = \frac{the\ number\ of\ correct\ permutations\ of\ bit\ pairs}{the\ number\ of\ all\ permutations\ of\ bit\ pairs}.$$

The number of all different permutations of n bits is $n!$. For three bytes, this number is $n! = 24! = 6.204484017 \times 10^{23}$.

For the given example in Fig. 2 probability P_{crack} of cracking one round of bit permutations for step 3 of the cryptographic transformation is

$$P_{\text{crack}} = \frac{8}{6{,}204484017 \times 10^{23}} = \\ = 1.289390057 \times 10^{-23}.$$

Therefore, the probability of cracking one round of bit permutations P_{crack} is quite low.

Further research consisted of the verification of the developed cryptographic transformation algorithm. For this, we chose the most common method of testing the statistical properties of NIST STS.

Binary sequence testing according to the NIST STS method has the following order [18-21, 28, 29]:

1. It is assumed that the binary sequence studied during testing $S = S_0, S_1, ... S_{n-1}$is random—the null hypothesis is accepted H_0.
2. Test statistics are calculated $c(S)$.
3. The value of a certain probability is determined using the test statistics function, $P = f(c(S))$, $P \in [0,1]$.
4. The probability value P is compared with the significance level $\alpha \in [0.001, 0.01]$. The null hypothesis is accepted in the case of $P \geq \alpha$, and in the opposite case, a conclusion is made that the alternative hypothesis is accepted.

The built-in tests included in the NIST STS package are shown in Table 2 [18, 20–21].

Using 16 built-in tests included in the NIST STS package [21], 189 probabilities P are calculated. Therefore, the result of testing is the construction of some vector of values of calculated probabilities $P = \{P_1, P_2, ..., P_{189}\}$. These probabilities can be considered as separate results of test calculations.

Table 2
List of tests in the NIST STS package

Number	Test Name
1	Frequency
2	Block Frequency
3	Runs
4	Long Runs of Ones
5	Binary Matrix Rank
6	Spectral Discrete Fourier Transform
7	Non-overlapping Templates Matching
8	Overlapping Templates Matching
9	Universal
10	Lempel Ziv Compression
11	Linear Complexity
12	Serial
13	Approximate Entropy
14	Cumulative Sums
15	Random Excursions
16	Random Excursions Variant

As a result of testing according to the NIST STS method, a statistical portrait is formed, the form of which is a matrix with dimension $m \times q$, where m is the number of binary sequences being tested, and q is the number of statistical tests used to test each sequence [18–21]. The elements of the matrix $P_{ij} \in [0, 1]$, where $i = \overline{(1, m)}$ and $j = \overline{(1, q)}$ are the values of the probability obtained as a result of testing the i^{th} sequence by the j^{th} test.

According to the obtained statistical portrait, a part of the sequences that passed each statistical test is determined. For this, the level of significance is set $\alpha \in [0.001, 0.01]$ and the probability values exceeding the established level of significance α are calculated for each of the q tests. That is, determine the coefficient [18–20]:

$$r_j = \frac{\#\{P_{ij} \geq \alpha | i = \overline{1,\ m}\}}{m}.$$

As a result, a vector of coefficients is formed, the elements of which characterize in percentage the passage of the sequence of all statistical tests.

The properties of pseudo-random sequences formed using the developed software tool were studied in the environment of statistical tests NIST STS. Statistical portraits of software implementations of the method of using information-driven permutation operations and DES [17, 24], AES [26–27], Blowfish [14, 29–30] algorithms, and linear feedback shift register are shown in Figs. 3–7 respectively.

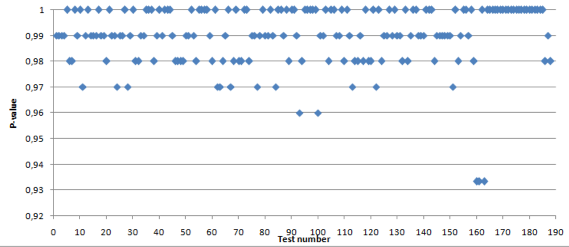

Figure 3: Statistical portrait of the sequence of the technique of using information-driven permutation operations

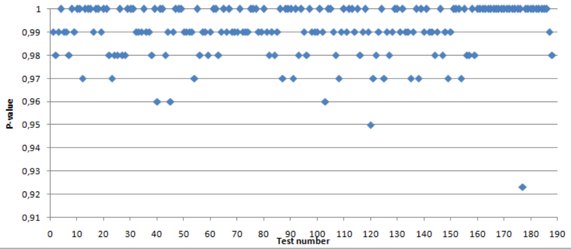

Figure 4: Statistical portrait of the sequence of the DES cryptographic algorithm

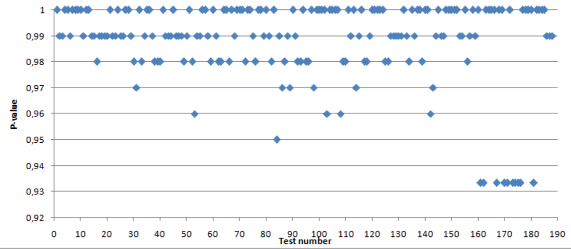

Figure 5: A statistical portrait of the sequence of the AES cryptographic algorithm

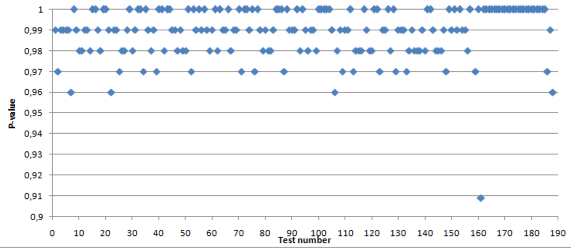

Figure 6: A statistical portrait of the sequence of the Blowfish cryptographic algorithm

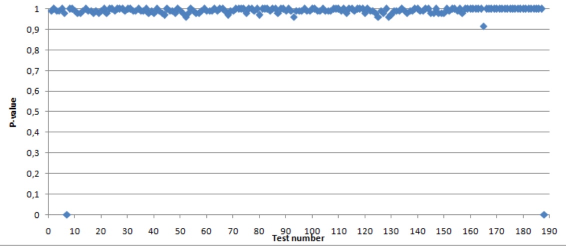

Figure 7: A statistical portrait of the sequence of scrambling with a LFSR word ($x^{24} + x^{23} + x^{21} + x^{20}$ polynomial)

To design the test, the following parameters were selected [28–29]:

1. The sequence length for testing: $n = 10^6$ bits.
2. Number of sequences that are tested: $m = 100$.
3. number of tests $q = 189$.

Thus, the tested sample size was $N = 10^6 \times 100 = 10^8$ bits, and the number of tests (q) for different lengths $q = 189$, thus, the statistical portrait of the generator contains 18900 probability values P.

In the ideal case, with $m = 100$ and $\alpha = 0.01$, only one sequence out of a hundred can be rejected during testing. The pass rate for each test must be 99%. But this is too strict a rule. Therefore, a rule based on the confidence interval is applied. The lower limit is 0.96015.

The structure and common basic transformations of the Kalyna block cipher are given in the papers [6–8]. A cipher mini-model is developed by scaling common cipher cryptographic transformations with the preservation of their algebraic structure in paper [7]. The developed mini-model [7] is intended to study the general characteristics of the cipher and was used for statistical testing by the NIST STS package.

The paper [8] is devoted to the study of statistical properties of crypto-algorithms by the Suite NIST STS and contains images of statistical profiles of algorithms, in particular, the Kalyna block cipher. We will use the experimental data obtained in works [6–8] regarding the study of the statistical properties of the Kalyna block cipher by using the Suite NIST STS for comparison with the results of the research of other algorithms obtained by us. Statistical portraits of the results of the Kalyna cipher are shown in Figs. 8–9 (Kalyna mini-version) [6–8].

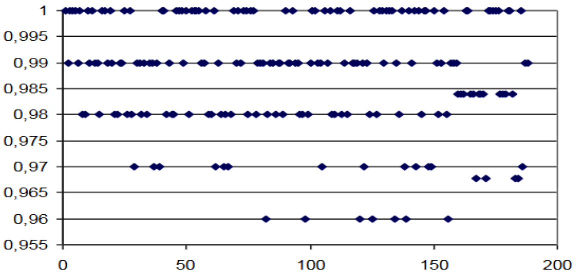

Figure 8: A statistical portrait of Kalyna cipher

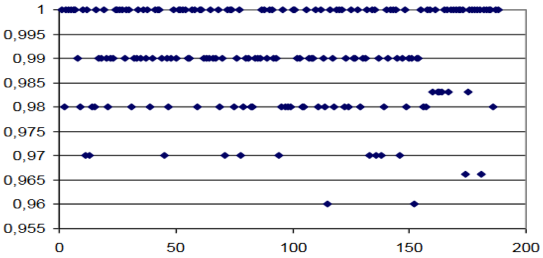

Figure 9: A statistical portrait of Kalyna (mini-version) cipher

According to the NIST STS testing methodology, the Kalyna cipher and mini-version of Kalyna cipher have the following results: the number of passed statistical tests according to the $P_i \geq 0.96$ criterion is 187 and 186 respectively, and according to the $P_j \geq 0.99$ criterion, it is 132 and 135 respectively.

Let's enter the data obtained in [6–8] regarding the statistical testing of the Kalyna cipher into Table 3 for a comparative analysis.

The work [9] presents the results of experimental studies of statistical security and speed characteristics of modern stream ciphers, in particular the Strumok cipher. In [11], a study of the statistical properties of standardized stream encryption algorithms was carried out and a detailed description and analysis of the obtained test results was given. The data obtained in the course of experimental studies in works [9, 11] can be used to conduct a comparative assessment of the statistical properties of the modern Strumok cipher and the software implementation of the method of applying in formation-driven permutation operations developed by us. Statistical portraits of the results of the Strumok cipher with a key length of 256 and 512 bits are shown in Figs. 10–11 respectively [11]. According to the NIST STS testing methodology, the Strumok cipher with 256- and 512-bit keys has the following results: the number of passed statistical tests according to the $P_i \geq 0.96$ criterion is 186 and 187 respectively, and according to the $P_j \geq 0.99$ criterion, it is 130 and 133 respectively.

Let's enter the data obtained in the paper [11] regarding the statistical testing of the Strumok cipher into Table 3 for comparative analysis.

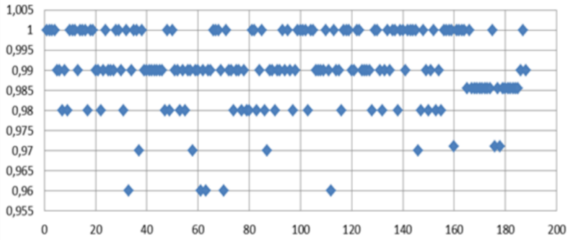

Figure 10: A statistical portrait of Strumok-256 cipher

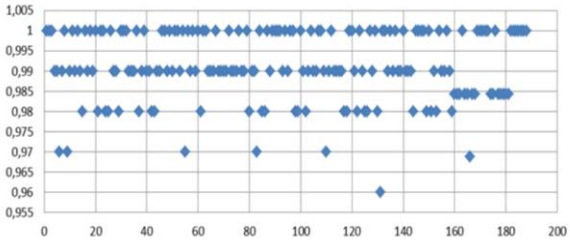

Figure 21: A statistical portrait of Strumok-512 cipher

Table 3 shows the results of testing the sequences formed based on the applied algorithms for comparison.

As can be seen from the results, the generator based on the algorithm developed by us passed the comprehensive control over the NIST STS methodology and has acceptable results compared to other generators (Table 3).

Table 3
Sequence test results

Generator	**Number of tests, in which tests have passed**	
	99% sequences	**96% sequences**
An algorithm that implements a technique of using information-driven permutation operations	138 (73.4%)	185 (98.4%)
DES	145 (77.1%)	186 (98.9%)
AES	138 (73.4%)	177 (94.1%)
Blowfish	129 (68.6%)	186 (98.9%)
Scrambling with a linear feedback shift register word ($x^{24}+x^{23}+x^{21}+x^{20}$ polynomial)	150 (79.8%)	185 (98.4%)
Kalyna	132 (70.2)	187 (99.5)
Kalyna (mini-versions)	135 (71.8)	186 (98.9)
Strumok-256	130 (69,1%)	186 (98.9%)
Strumok-512	133 (70,7)	187 (99,5%)

Thus, we can conclude that the developed algorithm for implementing the proposed technique of using information-driven permutation operations is suitable for cryptographic data transformation.

The method of researching the efficiency of implementation of cryptographic transformation by a certain algorithm presupposes the definition and analysis of encryption speed as one of the main indicators used when comparing cryptographic algorithms. The main requirement for speed measurement is the measurement of the encryption speed indicator of the same volume of open texts (in different modes) for all possible combinations of block size and key length under the same conditions within one interactive process of the user of the operating system. To ensure the same conditions for measuring encryption speed and for further qualitative assessment, it is necessary to consider that the results of speed testing of cryptographic algorithms are directly related to the technical characteristics of the selected hardware and software platform. In addition, speed indicators and their ratio for different ciphers can change significantly depending on the compiler version. More often, speed comparisons are performed for software implementations of cryptographic algorithms. Usually, specially developed optimized versions of the software implementation of the investigated cryptographic algorithms are used to evaluate the encryption speed to obtain the highest possible indicators.

It should be noted that the work carried out a software implementation of the developed algorithm of one of the methods of cryptographic data transformation based on the use of a basic group of permutation operations controlled by information containing only 25 basic synthesized groups to check the possibility of using similar operations when constructing cryptographic primitives. Since at this stage of the study, the development of an optimized software version of the implementation of the proposed method for applying information-driven permutation operations for the implementation of cryptographic data transformation was not carried out, then, accordingly, there is no possibility to evaluate the speed of the developed algorithm for comparison with other algorithms. Investigation of the implementation of this method and the effectiveness of its use in terms of encryption speed is planned to be carried out in further studies.

6. Conclusions

The paper proposes one of the techniques of using information-driven permutation operations for the implementation of cryptographic data transformation. The study of the implementation of this technique and the effectiveness of its use was carried out only on the example of 25 basic synthesized groups of 3-bit permutation operations controlled by information.

In the course of the study, we developed an algorithm for the proposed method of cryptographic data transformation based on the use of a basic group of information-driven permutation operations and implemented a software implementation using the high-level object-oriented programming language Python. Cryptographic transformation according to the developed algorithm is performed on 3 bytes of data using the following operations: information-driven permutations, Feistel network, and shift and addition modulo 2 operations.

Among the features of the implementation of the developed cryptographic data transformation algorithm is the use of a basic group of information-driven permutation operations, the multi-pass of the algorithm, in particular, there must be at least 5 transformation rounds, and also ensuring that the key value changes at each transformation round.

The obtained results of the work of the created software made it possible to conduct further research and carry out a qualitative assessment of the results of cryptographic data transformation according to the proposed method of using previously synthesized information-driven permutation operations. The effectiveness of this algorithm was evaluated based on statistical testing by the NIST STS package, as well as its suitability for implementing data encryption by hardware and software based on a comparison of test results with the results of using standard encryption algorithms (DES, AES, Blowfish, LFSR and modern Kalyna and Strumok ciphers).

Subsequent research should be directed to the study of operations of permutations controlled by information of greater capacity, as well as the use of the full set of basic groups of synthesized operations, which will provide an increase in the number of transformation operations and the possibility of processing data blocks of greater length. In addition, it is necessary to study and evaluate in more detail such parameters of the cryptographic transformation algorithm as cryptographic strength, avalanche effect, and speed.

References

[1] S. Gnatyuk, et al., High-Performance Reliable Block Encryption Algorithms Secured against Linear and Differential Cryptanalytic Attacks, CEUR Workshop Proceedings Vol. 2104 (2018) 657–668.

[2] S. Gnatyuk, et al., Studies on Cryptographic Security and Speed Analysis of New Advanced Block Cipher, CEUR Workshop Proceedings Vol. 2711 (2020) 202–213.

[3] K. Sudeepa, et al., Genetic Algorithm Based Key Sequence Generation for Cipher System, Pattern Recognit. Lett. 133 (2020) 341–348. doi: 10.1016/ j.patrec.2020.03.015.

[4] R. Oliynykov, et al., Design Principles and Main Properties of the New Ukrainian National Standard of Block Encryption, Ukrainian Inf. Secur. Res. J. 17(2) (2015) 142–157. doi: 10.18372/2410-7840.17. 8789.

[5] Y. Sovyn, et al., Effective Implementation and Performance Comparison of “KALYNA” and GOST 28147-89 Ciphers witch the Use of Vector Extensions SSE, AVX AND AVX-512, Ukrainian Ukrainian Inf. Secur. Res. J. Vol. 21(4) (2019) 207–223. doi: 10.18372/2410-7840.21.142 66.

[6] A. Kuznetsov, D. Ivanenko, E. Kolovanova, Perspective Block Cipher “Kalyna” Modelling, Appl. Radio Electron. 13(3) (2014) 201–207.

[7] S. Yevseiev, S. Ostapov, R. Korolev, Use of Mini-Versions to Evaluate the Security of Block-Symmetric Ciphers, Ukrainian Sci. J. Inf. Secur. 23(2) (2017) 100–108. doi: 10.18372/2225-5036.23.11796.

[8] V. Dolgov, A. Nastenko, Large Ciphers—Random Permutations. Verification of Statistical Properties Ciphers Submitted for Ukrainian Contecst with a Test Suite NIST STS, Inf. Proces.Syst. 7 (2012) 2–16.

[9] O. Kuznetsov, et al., Comparative Studies of Stream Cryptographic Transformation Algorithms, Radyotekhnyka 191 (2017) 52–75. doi: 10.30837/rt.2017.4.191.06.

[10] A. Oleksiichuk, S. Koniushok, M. Poremskyi, Security Justification for Strumok Stream Cipher Against Correlation Attacks Over Finite Fields of Characteristic 2, Math. Comput. Modeling 19 (2019) 114–119. doi: 10.32626/2308-5916.2019-19.114-119

[11] O. Kuznetsov, et al., Statistical Studies of Modern Stream Ciphers, Appl. Radio Electron. 15(3) (2016) 167–178.

[12] I. Fedotova-Piven, et al., The Inversion Method of Four-Bit Boolean SAC Cryptotransforms. Radio Electronics, Comput. Sci. Control. 4 (2019) 199–210. doi: 10.15588/1607-3274-2019-4-19.

[13] B. Shramchenko, Improving the Performance of Encryption Algorithms, 5th International Scientific and Practical Conference on Mechatronic Systems: Innovations and Engineering (2021) 180–181.

[14] V. Karpinets, A. Priymak, Y. Yaremchuk, Increasing the Stability of the Blowfish Cipher Based on the Optimization of Weak Keys by the Genetic Algorithm. Legal, Regulatory and Metrological Support of the Information Protection System in Ukraine 35 (2018) 106–115.

[15] O. Korchenko, V. Sidenko, Y. Dreis, Applied Cryptology: Encryption Systems, State University of Telecommunications (2014).

[16] T. Myroniuk, et al., Information Protection Based on Permutation Operations by Controlled of Information: monograph. Cherkasy State Technological University (2021).

[17] G. Mamonova, M. Mednikova, Cryptographic Analysis of the DES Algorithm, Model. Inf. Syst. Econom. Coll. Sci. Pr. 98 (2019) 146–156. doi: 10.33111/mise.98.15.

[18] P. Burciu, E. Simion, A Systematic Approach of NIST Statistical Tests Dependencies, J. Electrical Eng. Electronics Control Comput. Sci. 5(15) (2019) 1–6.

[19] R. Hegadi, A. Patil, A Statistical Analysis on In-Built Pseudo Random Number Generators Using NIST Test Suite, 5th International Conference on Computing, Communication and Security (2020). doi: 10.1109/ICCCS49678.2020.9276849.

[20] R. Kochana, et al., Statistical Tests Independence Verification Methods, Procedia Comput. Sci. 192 (2021) 2678–2688. doi: 10.1016/j.procs.2021.09.038.

[21] NIST 800-22 A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications.

[22] O. Pliushch, The Use of Cyclic Shifts of a Pseudo-Random Code Sequence to Improve the Characteristics of a Telecommunications Channel, Cybersecur. Educ. Sci. Technol. 1(9) (2020) 126–139. doi: 10.28925/2663-4023.2020.9.126139.

[23] B. Hamouda, Comparative Study of Different Cryptographic Algorithms, J. Inf. Secur. 11 (2020) 138–148. doi: 10.4236/jis.2020.113009.

[24] R. Sivakumar, B. Balakumar, V. ArivuPandeeswaran, A Study of Encryption Algorithms (DES, 3DES and AES) for Information Security, Int. Res. J. Eng. Technol. 5 (2018) 4133–4137.

[25] K. Patel, Performance Analysis of AES, DES and Blowfish Cryptographic Algorithms on Small and Large Data Files Int. J. Inf. Tecnol. 11 (2019) 813–819. doi: 10.1007/s41870-018-0271-4.

[26] Difference Between DES (Data Encryption Standard) and AES (Advanced Encryption Standard). URL: https://techdifferences.com/difference-between-des-and-17.aes.html

[27] K. Logunleko, O. Adeniji, A. Logunleko, A Comparative Study of Symmetric Cryptography Mechanism on DES, AES and EB64 for Information Security, Int. J. Sci. Res. Comput. Sci. Eng. 8(1) (2020) 45–51.

[28] NIST SP 800-22 Revision 1a, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. URL: https://nvlpubs.nist.gov/nistpubs/Lega

cy/SP/nistspecialpublication800-22r1a.pdf.

[29] Z. Mengdi, et al., Overview of Randomness Test on Cryptographic Algorithms, J. Phys. Conf. Ser. 1861 (2021). doi: 10.1088/1742-6596/1861/1/012009.

[30] H. Dibas, K. Sabri, A Comprehensive Performance Empirical Study of the Symmetric Algorithms:AES, 3DES, Blowfish and Twofish, International Conference on Information Technology (ICIT) (2021) 344–349. doi: 10.1109/ICIT52682.2021.9491644.