TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE
AUTOMATIC CONTROL | AUTOMATYKA

2-AC/2013

ANDRIY LAGUN*

CRYPTOGRAPHIC STRENGTH OF A NEW SYMMETRIC BLOCK CIPHER BASED ON FEISTEL NETWORK

KRYPTOGRAFICZNA ODPORNOŚĆ NOWEGO ALGORYTMU BLOKOWEGO SZYFROWANIA INFORMACJI OPARTEGO NA SIECI FEISTELA

Abstract

The paper summarizes research on the cryptographic strength of a new symmetric block cipher based on the Feistel network. The classification of cryptographic attacks, depending on the cryptanalyst's input data, is considered. For the purpose of testing, the linear and differential cryptanalysis as well as the Slide attack were used.

Keywords: *block cipher, the Feistel network, cryptographic strength*

Streszczenie

Niniejszy artykul pokazuje wyniki badania kryptograficznych szyfrów blokowych opartych na sieci Feistela. W artykule przedstawiono klasyfikacje ataków kryptograficznych na podstawie danych wejściowych, które ma analityk kryptograficzny. Do badań użyto ataki wykorzystujące metody kryptoanalizy liniowej i różnicowej, a także ataki typu Slide.

Słowa kluczowe: *szyfr blokowy, sieć Feistela, odporność kryptograficzna*

* Ph.D. Andriy Lagun, e-mail: a.e.lagun@gmail.com, Department of Information Security Management, Lviv State University of Life Safety.

1. Introduction

The rapid development of computer technology and open networks, modern methods of storage, processing and the transmission of information has contributed to the emergence of threats relating to the possibility of loss, disclosure and the modification of data belonging to different users. Cryptographic techniques and data protection form the basis of information security in information and telecommunication systems.

Determining the efficiency of cryptographic algorithms is usually a more difficult task than their design, because they requires a higher level of knowledge in this area and are inherently more scientific than engineering problems. This means there exists a large number of cryptographic protection methods, the reliability of which are not defined or guaranteed, because the algorithms on which they are based are insufficient or completely unexplored.

2. Review of cryptographic systems

Cryptographic systems are split into symmetric and public key systems. In symmetric cryptosystems, the same key is used for encryption and decryption. In public-key systems, public and private keys are used, which are mathematically related to each other. Information encrypted with a public key, which is available to everyone, is decrypted using the private key, known only to the recipient of the message.

Symmetric cryptosystems are split into block and stream systems.

Block ciphers are easy to use and allow for the handling of more parts of information compared to stream ciphers, in addition, they can be easy transformed into stream ciphers. Block ciphers are easily deployed. Their advantage over asymmetric ciphers is a greater performance and reliability using smaller keys [1].

Since the article deals mostly with block ciphers, we conducted a thorough investigation of them.

Block ciphers are one form of symmetric ciphers which allows for the handling of plaintext with blocks of multiple bytes per iteration. For a modern block cipher block, the size is 128 or 256 bits. The basic principles used in block ciphers are diffusion and confusion. Diffusion hides the statistical properties of the plaintext and ensures that any change of character in the plaintext or encryption key affects a large number of ciphertext characters. Confusion complicates the tracking of statistical dependencies between ciphertext and plaintext [8].

The main advantage of this class of ciphers is that in most cases, data encryption and decryption procedures differ only in the order of operations. This feature greatly simplifies the creation of software and hardware tools for encryption and enables the use of the same tools to both encrypt and decrypt data.

Block ciphers are two mutually related algorithms – an algorithm for encryption and a converse decryption algorithm which are demonstrated by the formulas (1) and (2) [1]. Input data are the blocks of plaintext (ciphertext) and the encryption key, encrypted (decrypted) data block of a similar size appear on the output. For a cipher of this type, an equation must be performed (3) that provides an unambiguous data encryption and decryption. Some of the encryption algorithms are based on transformations which are an involution. In this case, the encryption algorithm can be used for decryption without additional changes and modifications.

$$C = E(M, K) \tag{1}$$

$$M = E^{-1}(C, K) \tag{2}$$

$$M = E^{-1}(E(M, K)) \tag{3}$$

where:

M – block of plaintext,
C – block of ciphertext,
K – encryption key,
E – direct cryptographic transformation,
E^{-1} – inverse cryptographic transformation.

The block cipher consists of the simple transformation of plaintext that is performed in a certain sequence a certain number of times. These transformations with plaintext, or its component parts, or with the encryption key, provide the opportunity to achieve the main goal of encryption – to eliminate or significantly reduce the statistical information and dependence of the plaintext. In other words, it is necessary to increase the entropy of the plaintext to such a value where a relationship between the input and output of the cryptographic algorithm is not observed. In most cases, operations and transformations which are used in the cryptographic algorithm should have an inverse transformation – this is demonstrated in expression (4). In this case, the realization of the operations (which have an inverse) with plaintext will also have an inverse operation. This will be a set of inverse transformations applied in an inverse order, as shown in formula (5).

$$E^{-1}(E(M)) = M \tag{4}$$

$$E_3^{-1}(E_2^{-1}(E_1^{-1}(E_3(E_2(E_1(M)))))) = M \tag{5}$$

Certain operations can be inverse of themselves, namely, involutions. An example of this type of operation is the exclusive *OR* operation (*XOR*), which is most common in cryptographic algorithms. The two main methods used for this purpose are diffusion and confusion. The application of confusion provides a certain property, when the change in one byte of plaintext leads to changes in many bytes of ciphertext – this is the so-called ‘avalanche’ effect. The easiest way to achieve this effect is to use transposition. Mixing allows you to hide the statistical properties of the plaintext and its redundancy. The simplest variant of mixing is the alphabetical substitutions of different types. As a rule, in modern block ciphers these methods have never been used alone, but only in combination. This fact allows for achieving the best effect.

In addition to these methods, different algebraic operations are used. These operations often belong to different algebraic groups. An example of using this type of operation is the IDEA algorithm, which uses the operations of *the* addition modulo, multiplication and *XOR*. The feature of these operations is their incompatibility in the sense that no two of them satisfy the associative and distributive laws, which greatly complicates cryptanalysis.

In addition to the above, cryptographic algorithms frequently use operations such as logical shifts, multiplication in the Galois field, Hadamard pseudo transformation, and different arithmetic and logical operations.

Since the usage of one of the above operations does not provide the necessary result, these operations should be used in a specific combination and according to certain rules in order to achieve the desired effect. The most common of these combined methods are the Feistel network and the Substitution-Permutation network, as well as their modification or combination.

The Feistel network is a certain structure, which is repeated a certain number of times, for each of which, a different round key is used [2]. Encryption and decryption operations on every stage are quite simple and usually identical, but require the reordering of round keys.

The essence of the Feistel transform is determined using the following algorithm [3]:

- the block of plaintext is split into two equal parts – left and right;
- to the left part and the round key, certain functions are applied and to the execution result of this function and the right part operation, XOR is performed; the result from the previous step is assigned to the new left sub-block, whereas to the right sub-block is assigned the unmodified previous right sub-block;
- the two previous steps are repeated a certain number of times with different round keys.

3. Description of the algorithm

The algorithm is a designed block encryption algorithm based on the Feistel network (Fig. 1) which has the following characteristics and properties:

- data block size: 256 bits;
- variable key size encryption (to 256-bit) and a variable number of rounds;
- orientation to 64-bit architecture;
- uses methods and mechanisms to prevent known types of attacks.

This algorithm is based on the use of direct changes in the encrypted information (Fig. 1a) and the inverse – in the decrypted information (Fig. 1b). Data encryption and decryption in this algorithm is described by formulas (6) and (7), respectively:

$$X_i = E2(E1(X_{i-1}, rK_i), rK_i) \tag{6}$$

$$X_i = D1(D2(X_{i-1}, rK_{n+1-i}), rK_{n+1-i}) \tag{7}$$

where

X_i	–	block input data (plaintext or ciphertext),
$E1, E2, D1, D2$	–	Feistel transformations of different types,
rK_i	–	round key,
i	–	round number.

There have been different types of Feistel networks used with different properties in this algorithm. This allows for achieving different levels of dispersion and implicit use of different types of permutations in one round.

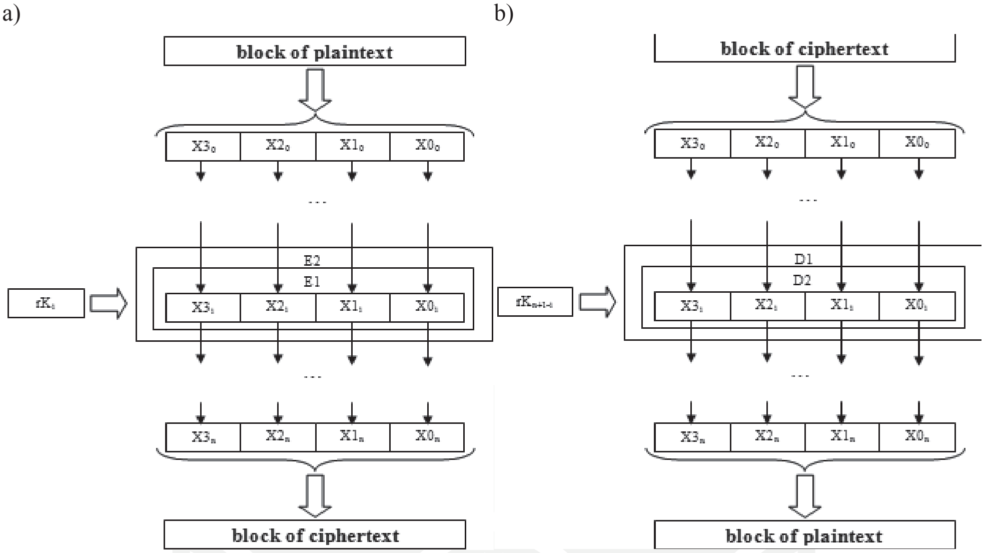

Fig. 1. The general structure of the algorithm: a – encryption scheme; b – decryption scheme

3.1. The Feistel transformation

During the encryption of the information for each pair of blocks, transformations *E*1 and *E*2 are used. These transformations are alternated between themselves. During the decryption, a pair of transformations *D*1 and *D*2, which are inverse of the previous ones, must be applied in reverse order.

Transformation *E*1 is shown in Fig. 2a. The block of information on its input is split into four equal parts $X0_{i-1} - X3_{i-1}$. Part $X3_{i-1}$ and the round key rK_i are input arguments of function *F*1, which has three outputs. These outputs form new values $X1_i - X3_i$ after summation modulo 2 with values $X0_{i-1} - X2_{i-1}$. Part $X0_i$ is formed by cyclic *i* bit shift of $X3_{i-1}$. This transformation provides a modification of all input parts, moreover, the dependence of the transformation *E*1 on the round number provides another modification of the block every time.

This transformation is described by formulas (8) and (9). To perform the inverse operations, transformation *D*1 is applied, as shown in Fig. 2b. It is performed by a similar principle and has the same properties as transformation *E*1 (formulas (10), (11)).

Transformation *E*2 is shown in Fig. 2c. The block of information on its input is split into four equal parts $X0_{i-1} - X3_{i-1}$. Parts $X0_{i-1} - X2_{i-1}$ and round key rK_i are input arguments of function *F*2 which forms the output part of the transformation $X0_i$ by performing the *XOR* operation with the output function *F*2 and part $X3_{i-1}$. Outputs $X1_i - X3_i$ are not modified parts of $X0_{i-1} - X2_{i-1}$ respectively. This transformation provides a modification of only one part of the input, but this modification depends on all parts of the input block $X0_{i-1} - X3_{i-1}$ and round key rK_i. To perform the inverse operations, transformation *D*2 is applied, as shown in Fig. 2d. It is performed by a similar principle and has the same properties as transformation *E*2. Transformations *E*2 and *D*2 are described by the formulas (12)–(15).

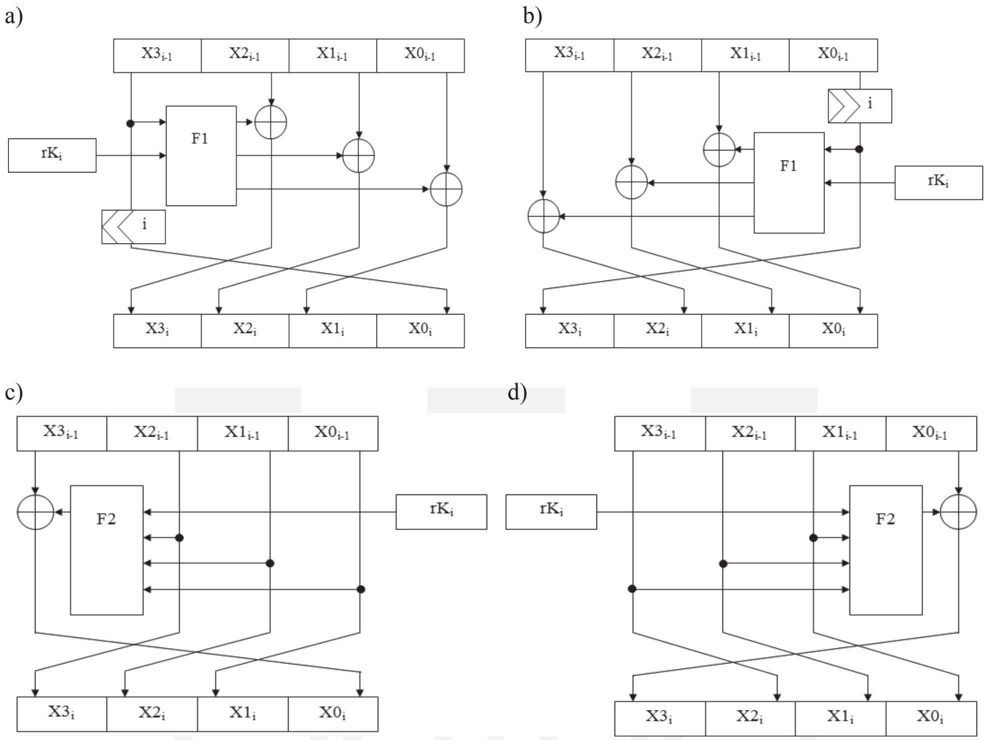

Fig. 2. Structures of used Feistel networks: a – *E*1, b – *D*1, c – *E*2, d – *D*2

$$X0_i = X3_{i-1} << i \tag{8}$$

for $t = 1..3$
$$X(t)_i = F1(i, X3_{i-1}, \mathrm{rK}_i)_1 \oplus X(t-1)_{i-1} \tag{9}$$

for $t = 0..2$
$$X(t)_i = F1(X0_{i-1} >> (n+1-i), rK_{n+1-i})_1 \oplus X(t+1)_{i-1} \tag{10}$$

$$X3_i = X0_{i-1} >> (n+1-i) \tag{11}$$

$$X0_i = F2(i, X2_{i-1}, X1_{i-1}, X0_{i-1}, rK_i) \oplus X3_{i-1} \tag{12}$$

for $t = 1..3$
$$X(t)_i = X(t-1)_{i-1} \tag{13}$$

for $t = 0..2$
$$X(t)_i = X(t+1)_{i-1} \tag{14}$$

$$X3_i = \mathrm{F2}(i, X3_{i-1}, X2_{i-1}, X1_{i-1}, rK_{n+1-i}) \oplus X0_{i-1} \tag{15}$$

where $F1(i, X, rK)_j$, $F2(i, X3, X2, X1, rK)$ are complicated modification functions.

The described transformations contain different degrees of dispersion and implicit permutations of different types through the use of cyclic shift operations and structure changes. Transformations *E*1 and *D*1 are focused on the diffusion of bits of the whole input block and permutations in one of its parts. Transformations *E*2 and *D*2 are focused on the diffusion of bits in one part of the input block and permutations in the whole block. The availability of diffusion and permutation methods, and also striping of the transformations of different types, provide rapid achievement of the 'avalanche' effect.

3.2. Complicated modification function

Function *F*1, the structure of which is shown in Fig. 3, contains three input arguments: round *i* number, data block *X* and round key *rK* sized 64 bits; and returns three output values $F1_1 - F1_3$.

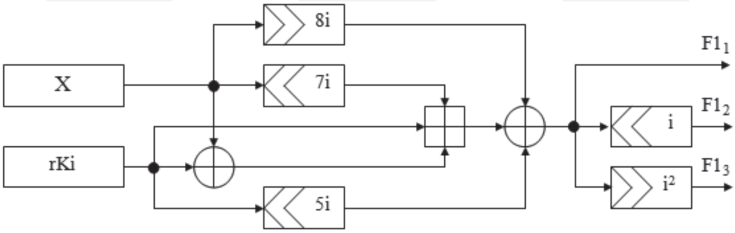

Fig. 3. Structure of complicated modification function *F*1

Function *F*1 uses three types of operations: exclusive *OR*; addition modulo 2^{64}; and the cyclic shift. Using operations of these types provides protection against linear and differential cryptanalysis provided their alternation. The main result of the transformation appears at the output $F1_1$, and the other two outputs are permutations of this result, which are achieved through the use of cyclic shifts (formulas (16)–(18)).

$$F1_1(X1, rK_i) = ((rK_i \oplus X1) + (X1 << 7i) + rK_i) \oplus (X1 >> 8i) \oplus (rK_i << 5i) \qquad (16)$$

$$F1_2(X1, rK_i) = F1_1(X1, rK_i) << i \qquad (17)$$

$$F1_3(X1, rK_i) = F1_1(X1, rK_i) << i^2 \qquad (18)$$

The use of different types of shifts in a different number of positions after a sufficient number of rounds provides permutation across the set of the input block bits. The scattering of bits of the input block is achieved by imposing the initial and modified input arguments.

Function *F*2, the structure of which is shown in Fig. 4, contains five input arguments – number of round *i*, data blocks *X*1 – *X*3 and round key *rK* sized 64 bits; it returns one output value. This function uses three additional constants const1 – const3, which provides additional scattering of bits. This function is represented by formulas (19)–(22).

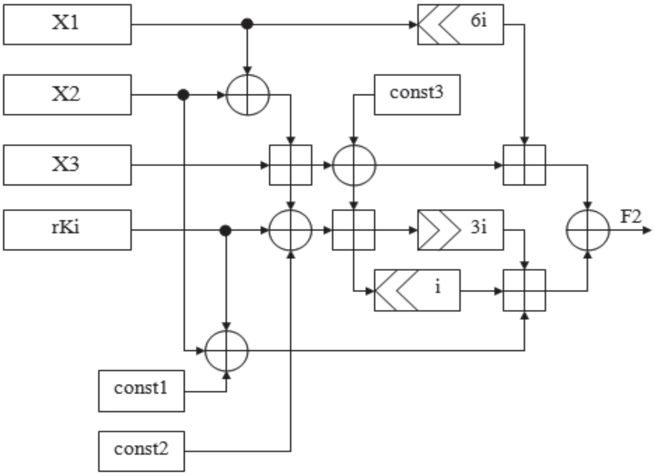

Fig. 4. Structure of complicated modification function *F*2

$$\lambda_1 = (X1 \oplus X2) + X3 \tag{19}$$

$$\lambda_2 = \lambda_1 \oplus \text{const3} \tag{20}$$

$$\lambda_3 = \lambda_2 + (\lambda_1 \oplus rK_i \oplus \text{const2}) \tag{21}$$

$$F2(X1, X2, X3, rK_i) = ((X1 << 6i) + \lambda_2) \oplus ((\lambda_3 >> 3i) + \\ + (\lambda_3 << i) + (X2 \oplus rK_i \oplus \text{const1})) \tag{22}$$

Function *F*2 is applied to modify only one part, but it allows for providing a greater effect of diffusion by the complicated internal structure.

Implementing the algorithm. Constants const1 – const3 can be initialized by the arbitrary values, or used as an additional key information It is desirable that the constants are of a random bit sequence. This would bring more sustainability than a periodic sequence filling.

The following function is applied in transformations *E*2 and *D*2, as well as to form the round keys, which allows to form strong subkeys.

3.3. Mechanism of round keys formation

In order to form the round keys, the input key, if necessary, is complemented by zero bits to the size of 256 bits. The structural diagram of the key formation for the *i*-th round is shown in Fig. 5.

The parts of the current encryption key are supplied as input arguments to the input of function *F*2 in the order determined depending on the number of the round. The round key is the output value. The resulting function is:

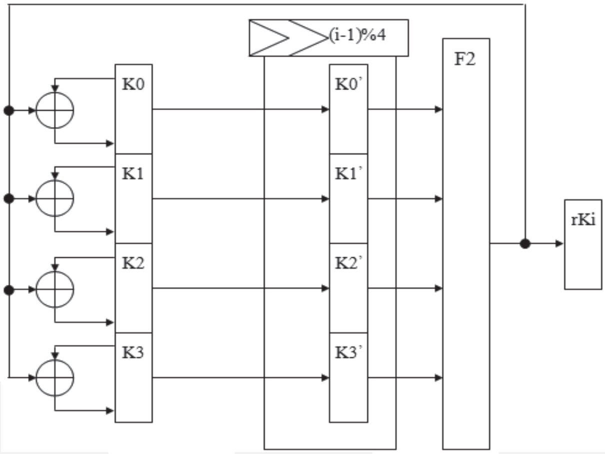

Fig. 5. Mechanism of round keys formation

If:

$$\begin{cases} (i-1)\%4=0 & \Rightarrow & rK_i = F2(K0, K1, K2, K3) \\ (i-1)\%4=1 & \Rightarrow & rK_i = F2(K1, K2, K3, K0) \\ (i-1)\%4=2 & \Rightarrow & rK_i = F2(K2, K3, K0, K1) \\ (i-1)\%4=3 & \Rightarrow & rK_i = F2(K3, K0, K1, K2) \end{cases} \tag{23}$$

To form strong round keys, the input key value is modified by operation XOR with its parts and the round key itself.

4. Overview of cryptanalysis methods

The appearance of newly encrypted algorithms leads to the development of their hacking methods. If the purpose is the disclosure of as many ciphers as possible, then the best strategy is to develop universal methods of cryptanalysis.

At this time, there are many methods of block ciphers analysis such as the brute force method, the statistical method, the method of meeting in the middle, linear and differential cryptanalysis, the boomerang method, the slide – attack and others [4, 5].

The brute force method provides the iteration of all possible variants of encryption keys. To search for a key that has a size of n bits, there are 2^n variants. After the iteration of all possible keys, the encryption key will be found. On average, this search requires 2^{n-1} test operations of encryption.

Protection against attacks of this type is an increase of the key size, because an increase of the key size at one bit leads to an increase twice in the number of the encryption key variants.

To increase the efficiency of this method, paralleling is used if the required resources are available, special devices for exhaustive key search and others are applied.

The task of the statistical method is the development of algorithms that determine an unknown key or part of this key. Implementations of this method for particular block ciphers are more efficient than the brute force method.

To the input of the algorithm, a certain amount of pairs (X_i, Y_i) $i = 1 \,..\, n$ of the plaintext and ciphertext is supplied. These pairs are derived from the application of mapping F with key k. It is assumed that the plaintexts are chosen randomly, equiprobably and independently from the whole aggregation. The idea of statistical analysis is that if the results of observations are different, then after a sufficiently large number of observations, it is possible to determine, with a certain probability, the law of observations, that is the searched value.

Linear cryptanalysis combines the search of linear statistical analogues for encryption equations, statistical analysis of the plaintexts and ciphertexts, and methods of coordination and busting. This method examines statistical linear links between the various bits of vectors of the plaintext, ciphertext and the key, and uses these links to determine some bits of the key by the statistical methods. This method uses linear approximations for describing of the work of the cryptographic algorithm.

This method is carried out in two stages:

- forming links between plaintext, ciphertext and key that occur with high probability;
- these links with known pairs of the plaintext – ciphertext are used for getting key bits.

These links are called linear approximations. It is necessary to define the links, the probability of which is not equal 1/2, and use them to find appropriate bits of the key.

To protect against attacks with using linear cryptanalysis it is necessary to achieve that with any change of the plaintext or key, each of the bits of the ciphertext will change with probability Differential cryptanalysis uses pairs of the ciphertext with some differences. The essence of this method is to analyze the evolution of this difference in the process of passing the plaintext through the stages of encryption with one and the same key

The two plaintexts with a fixed distinction are selected. After passing of all stages of encryption and analyzing the distinction in the obtained ciphertexts, different probabilities are assigned for different keys. During further analysis of the following pairs, one of the keys will be more probable – it will be the encryption key.

This method of cryptanalysis has different versions. One of them is the use of impossible differentials (with zero probability). The hacking procedure is as follows: the required number of pairs of the plaintexts with the required distinction is selected; the appropriate ciphertexts are found; the analysis of the received data is performed and all versions of the encryption keys that lead to impossible differentials are considered incorrect and discarded.

Thus, some set of possible keys is received that do not lead to impossible differentials. Differentials with zero probability can be replaced by differentials with minimal probability. At the same time, the procedure of attack on the algorithm is similar to the procedure that is used in the impossible differentials.

There is another kind of attack – slide-attack. The feature of this attack is that its successful application does not depend on the number of rounds of the algorithm in which it is applied. The only requirement for its application is that rounds of the algorithm must be identical [6].

Let it is applied the multiround cryptographic algorithm with round function $E(P, k)$ and the condition that the round keys of the algorithm are identical. Then for attack the pair of the plaintexts is applied, one of which is chosen randomly by the text P, and the other P' is the result of the oneround transformation of the text P:

$$P' = E(P, k) \tag{24}$$

The ciphertexts that refer to the plaintexts, are similarly related to each other:

$$C' = E^{-1}(C, k) \tag{25}$$

Having the pairs of the plaintexts and ciphertexts, which are related by only one round of the encryption, the round key can be obtained, this allows the disclosue of the full algorithm. The slide-pair *capital 'S'?* is defined as follows [7]:

- for the text P and for each plaintext P' the appropriate ciphertexts C and C' are received;
- the value of round key k_1 is calculated by the formula (24);
- the value of round key k_2 is calculated by the formula (25);
- the coincidence of these keys means that the required plaintext is found and $k = k_1 = k_2$ is the round key.

5. Evaluation of the algorithm stability

For the attack on the cipher by the method of exhaustive search, it is necessary to check up to 2^{256} key options. Verification of imitative sustainability was carried out by determining the impact of a change of one of the bits in the encrypted data block to the decrypted block. The average value of the difference between blocks is approximately 50%, regardless of position of the changed bit. To check availability of the avalanche effect it the impact of change of one of the bits of the key or the plaintext block to the ciphertext was investigated. Regardless of the position of the changed bit and its location (in the key or plaintext), the average value of the changes in the ciphertext is approximately 50% [9].

To check the described characteristics, tests with combinations of different types of input data were performed (block of plaintext and key): completely zero vector; completely unit vector; pseudo randomly generated vector.

Sustainability against attacks, based on the methods of differential and linear cryptanalysis, results in difficulty in determining the key knowing the input and output values of the round functions. The functions of complicated transformation are designed in such a way that selection of the round key among them is a difficult task. Since for transformations in the functions, bitwise shift operations are used, including with the key, the part of the input data, which enters the input of such an operation is rejected and therefore, performance of the inverse operation is impossible. In the best case, certain dependencies (26) for each of the round keys, but not the key in an explicit form, can be received. Knowing these dependencies you can distinguish the part of bits of the key, but not the complete key. For example, the dependence (27) shows that you can be sure in only two higher bits of the key and other bits are unknown.

$$K_r \oplus (K_r \ll n) \oplus \ldots = X \tag{26}$$

$$K_r \oplus (K_r \ll 2) = X \tag{27}$$

The application of this approach for masking the value of the round key provides the inability to obtain the key value even with known input and output function values. The selection of bits on which the shift is performed, is providing an opportunity to mask the round key along its length.

Creating a table of approximations for the linear cryptanalysis and creating a table of differentials for the differential cryptanalysis is complicated by the fact, that there is no possibility of abstraction of the key value for each concrete case, because there are no operations performed only with the input value. All operations are dependent on the input value as well as on the key – this is why the creation of tables is needed for each new key value.

Implementation of the slide attack for this algorithm is meaningless, because in addition to the above complexities in the allocation of the key given algorithm has the method of forming round keys, that provides different round keys for each round. Therefore, it is impossible to share cryptographic algorithm to the equal parts that are repeated.

Testing of the *cryptographic algorithm using statistical tests.*

Since any encryption algorithm can be considered as a pseudorandom function which depends on the input text and the key, then to analyze other statistical characteristics of the cipher testing with various sets of statistical tests (DIEHARD, NIST) was carried out [10]. Some test results using the NIST test are shown in Table 1.

Table 1

NIST test results for cryptographic algorithm [11]

No.	Name of test	Result (*p-value*)
1	Frequency (Monobits) Test	0.739918
2	Cumulative Sum (Cusum) Test	0.122325
3	Runs Test	0.122325
4	Test For The Longest Run Of Ones In A Block	0.911413
5	Random Binary Matrix Rank Test	0.350485
6	Discrete Fourier Transform (Spectral) Test	0.066882
7	Non-Overlapping (Aperiodic) Template Matching Test	0.350485
8	Approximate Entropy Test	0.213309
9	Serial Test	0.534146
10	Linear Complexity Test	0.739918

These tests are various statistical tests, results of which are interpreted according to the methodology of determining the probability *p-value*. At small values of probability, the sequence is not considered random and for successful testing, the probability should be higher than 0.01 [11].

According to the testing results, the value of the variables of *p-value* greater than 0.05 was obtained, that is quite a good result

6. Conclusions

In the article, the methods and approaches of the construction of block ciphers have been analysed – several issues related to the security of the cryptographic algorithm as well as the concept of cryptographic strength have been examined. These facts allowed us to determine the requirements for modern cryptographic systems and for block ciphers.

The block encryption algorithm based on the Feistel transformation that is resistant to cryptographic attacks using modern methods of cryptanalysis has been presented. Constructive solutions that allow masking the value of round keys even with the known input and output values of the function have been offered for this algorithm. It can be used in block ciphers to provide resistance to cryptographic attacks.

The most prevalent methods of cryptanalysis of block ciphers have also been reviewed, such as the brute force method, meeting in the middle, statistical method, linear cryptanalysis, differential cryptanalysis and slide attack.

The results of the investigation of the designed cryptographic algorithm have been presented by means of NIST tests, and its resistance to cryptographic attacks has been demonstrated.

References

[1] Sneier B., *Applied Cryptography*: Protocols, Algorithms, and Source Code in C, 1996.

[2] Feistel H., *Cryptography and Computer Privacy*, Scientific American, Vol. 228, No. 5, 1973.

[3] Hoang V.T., Rogaway P., *On Generalized Feistel Networks*, Dept. of Computer Science, University of California, Davis, USA. 2010, 26.

[4] Biham E., Shamir A., *Differential cryptanalysis of DES-like cryptosystems*, Journal of Cryptology, Vol. 4, No. 1, 1991, 3-72.

[5] Wagner D., *The boomerang attack*, U.S. Berkeley.

[6] Chalermpong Worawannotai, Isabelle Stanton. A Tutorial on Slide Attacks.

[7] Ciet M., Piret G., Quisquater J., *Related-Key and Slide Attacks: Analysis*, Connections, and Improvements, 002.

[8] Blaze M., Diffie W., Rivest R.L., Schneier B., Shimomura T., Thompso E., Wiener M., *Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security*, 1996.

[9] Stamp M., Low R.M., *Applied Cryptanalysis: Breaking Ciphers in the Real World*, Wiley-IEEE Press, 2007.

[10] Instructions for using DIEHARD: a battery of tests of randomness, 1997.

[11] www.csrc.nist.gov/groups/ST/toolkit/rng/stats_tests.html.