ANDRIY LAGUN*

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

KRYPTOGRAFICZNA ODPORNOŚĆ NOWEGO ALGORYTMU BLOKOWEGO SZYFROWANIA INFORMACJI OPARTEGO NA SIECI FEISTELA

A b s t r a c t

The paper summarizes research on the cryptographic strength of a new symmetric block cipher based on Feistel network. The classification of cryptographic attacks, depending on the cryptoanalyst'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, Feistel network, cryptographic strength

S t r e s z c z e n i e

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

* Andriy Lagun, Ph.D., associate professor, 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 transmission of information has contributed to the emergence of threats related 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 it requires a higher level of knowledge in this area and are inherently more scientific than engineering problems. This means there is a large number of cryptographic protection methods, the reliability of which is not defined and 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 the stream ciphers, in addition, they can be easy transformed into the 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 cipher - is one form of symmetric ciphers which allows handling 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. The first of them hides the statistical properties of the plaintext and ensures that any change of character of the plaintext or encryption key affects on a large number of ciphertext characters. The other - complicates the tracking of statistical dependencies between ciphertext and plaintext [8].

The main advantage of this class 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 of encryption and enables the use of the same tools to encrypt and decrypt data.

Block ciphers are two mutually related algorithms: an algorithm for encryption and 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.

$$C = E(M, K) \quad (1)$$
$$M = E^{-1}(C, K) \quad (2)$$
$$M = E^{-1}(E(M, K)), \quad (3)$$

where:

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

Block cipher consists of the simple transformation of plaintext that are performed in certain sequence some 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 statistic of the information and dependence of the plaintext. In other words, it is necessary to increase the entropy of the plaintext to such value, when between input and output of the cryptographic algorithm relationship is not observed.

In most cases, operations and transformations which are used in the cryptographic algorithm, should have an inverse transform, this is demonstrated in expression (4). In this case, realization of the operations (which have an inverse) with plaintext will also have an inverse operation. This will be set of inverse transformations applied in an inverse order, as shown in formula (5).

$$E^{-1}(E(M)) = M \quad (4)$$
$$E_3^{-1}(E_2^{-1}(E_1^{-1}(E_3(E_2(E_1(M)))))) = M \quad (5)$$

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. Application of confusion provides certain property, when the change in one byte of plaintext leads to changes in many bytes of ciphertext that is 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 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 uses operations such as logical shifts, multiplication in the Galois field, Hadamard pseudo transformation, 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 reordering of round keys.

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

- block of plaintext is split into two equal parts – left and right;
- to the left part and round key certain function is applied and to the execution result of this function and the right part operation *XOR* is performed;
- 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

It was 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

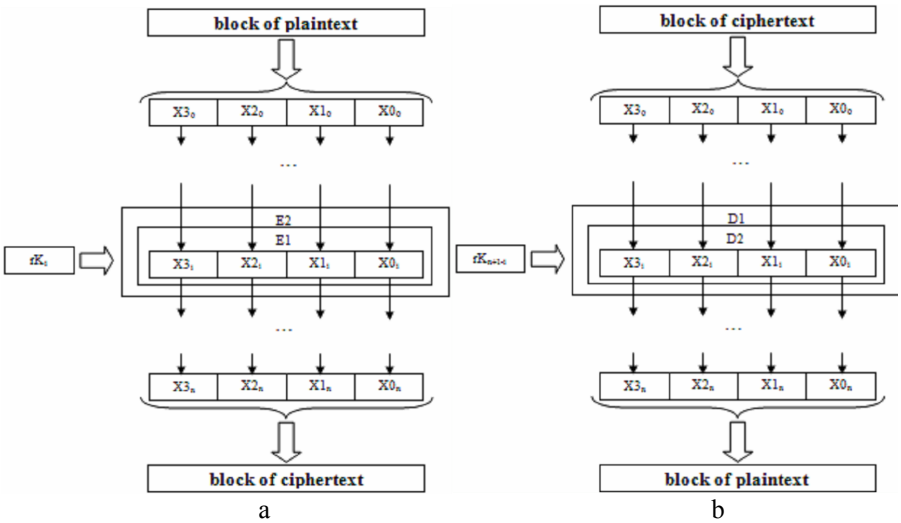

Fig. 1. The general structure of the algorithm: a - encryption scheme; b - decryption scheme

Rys. 1. Podstawowa struktura algorytmu: a – szyfrowanie; b - deszyfrowanie

(Fig. 1a) and the inverse - in decrypted (Fig. 1b). Data encryption and decryption in this algorithm is described by formulas (6) and (7), respectively

$$X_i=E2(E1(X_{i-1},rK_i),rK_i) \quad (6)$$
$$X_i=D1(D2(X_{i-1},rK_{n+1-i}),rK_{n+1-i}) \quad (7)$$

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

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

3.1. The Feistel transform

During the encryption of the information for each pair of blocks, transformations *E1* and *E2* are used. These transformations are alternated between themselves. During the decryption, a pair of transformations *D1* and *D2*, which are inverse of the previous, must be applied in reverse order.

Transformation *E1* 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 argument of function *F1*, 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 $X0_i$ formed by cyclic shift of the $X3_{i-1}$ on i bits. This transformation provides a modification of all input parts, moreover, the dependence of the transformation *E1* on the number of round provides every time another modification of the block.

This transformation is described by the formulas (8) and (9). To perform the inverse operations transformation *D1* is applied, as shown in Fig. 2b. It is performed by the similar principle and has the same properties as transformation *E1* (formulas (10), (11)).

Transformation *E2* is shown in Fig. 2c. On its input block of information comes, that 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 *F2* that forms output part of the transformation $X0_i$ by performing *XOR* operation with the output function *F2* and part $X3_{i-1}$. Outputs $X1_i$ - $X3_i$ are not modified parts $X0_{i-1}$ - $X2_{i-1}$, 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 *D2* is applied, as shown in Fig. 2d. It is performed by the similar principle and has the same properties as transformation *E2*. Transformations *E2* and *D2* are described by the formulas (12) - (15).

$$X0_i = X3_{i-1} << i \quad (8)$$
for $t=1..3$ $$X(t)_i = F1(i, X3_{i-1}, rK_i)_1 \oplus X(t-1)_{i-1} \quad (9)$$
for $t=0..2$ $$X(t)_i = F1(X0_{i-1} >> (n+1-i), rK_{n+1-i})_1 \oplus X(t+1)_{i-1} \quad (10)$$
$$X3_i = X0_{i-1} >> (n+1-i) \quad (11)$$

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

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

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

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

where $F1(i,X,rK)_j$, $F2(i,X3,X2,X1,rK)$ – 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 *E1* and *D1* are focused on diffusion of bits of the whole input block and permutations in one of its parts. Transformations *E2* and *D2* are focused on diffusion of bits in one part of the input block and permutations in the whole block. Availability of diffusion and permutation methods, and also striping of the transformations of different types provide rapid achievement of "avalanche" effect.

3.2. Complicated modification function

Function *F1*, which structure is shown in Fig. 3, contain three input arguments: number of round *i*, data block *X* and round key *rK* sized 64 bits; and return three output values $F1_1 - F1_3$.

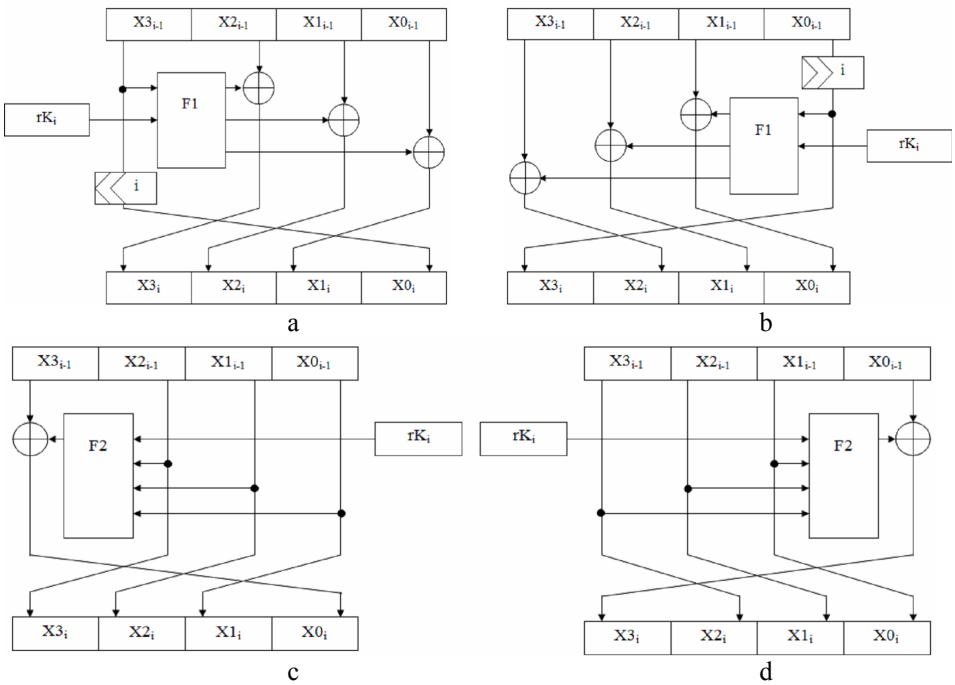

Fig. 2. Structures of used the Feistel networks: a – *E1*, b – *D1*, c – *E2*, d – *D2*

Rys. 2. Struktura używanych sieci Feistela: a – *E1*, b – *D1*, c – *E2*, d – *D2*

Function *F1* uses three types of operations: exclusive OR, addition modulo 2^{64} and the cyclic shift. Using operation of these types provides protection against linear and differential cryptanalysis provided their alternation. The main result of the transform appears at the output of $F1_1$, and the other two outputs are permutations of this result, that are achieved by the use of cyclic shifts (formulas (16) - (18)).

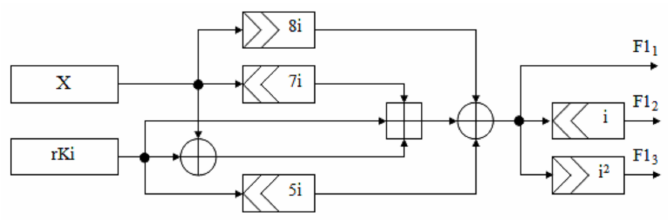

Fig. 3. Structure of complicated modification function *F1*

Rys. 3. Struktura funkcji złożonego przetwarzania *F1*

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

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

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

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

Function *F2*, which structure is shown in Fig. 4, contain five input arguments: number of round *i*, data blocks *X1* – *X3* and round key *rK* sized 64 bits and return one

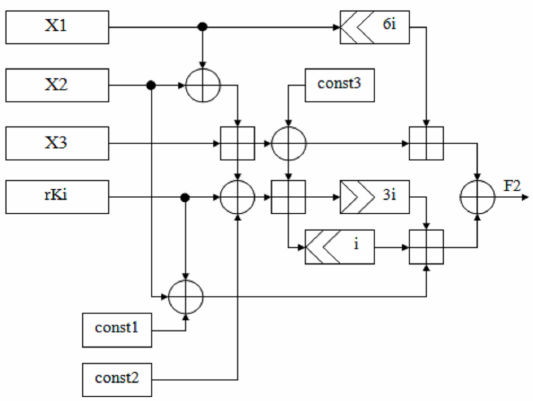

Fig. 4. Structure of complicated modification function *F2*

Rys. 4. Struktura funkcji złożonego przetwarzania *F2*

output value. This function uses three additional constants *const1* - *const3*, which providing additional scattering bits. This function is represented by formulas (19) - (22).

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

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

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

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

Function *F2* is applied to modify only one part, but allows to provide a greater effect of diffusion by the complicated internal structure.

In the implementing of 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 will be a sequence of random of bits. This would bring more sustainability than filling by a periodic sequence.

The following function is applied in the transformations *E2* and *D2*, as well as to form round keys, that allows to form strong subkeys.

3.3. Mechanism of round keys formation

To form round keys 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.

To the input of function F2 the parts of the current encryption key are supplied as input arguments in the order determined depending on the number of the round. The round key is the output value. The resulting function is:

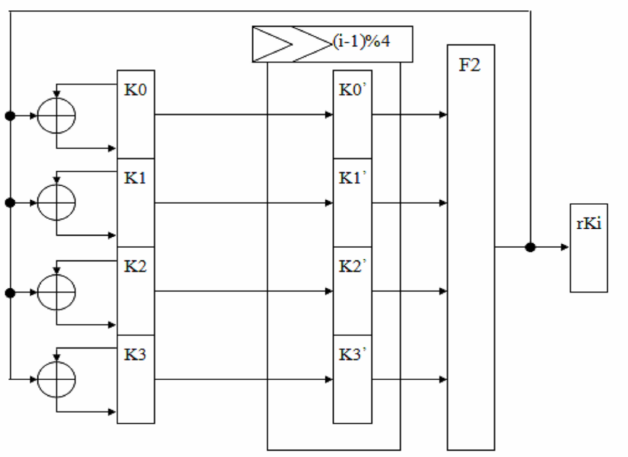

Fig. 5. Mechanism of round keys formation

Rys. 5. Mechanizm tworzenia kluczy powtarzających etapów

If

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

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

4. Overview of cryptanalysis methods

The appearance of new encrypted algorithms leads to the development of their hacking methods. If the purpose is the disclosure as manyas possible ciphers, 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, statistical method, method of meeting in the middle, linear and differential cryptanalysis, the boomerang method, 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 in the number of the encryption key variants in twice. To increase the efficiency of this method parallelization at presence of the required resources, using of speciality devices for the key iteration 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 is 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 regardless from the whole set. The idea of statistical analysis is that if the results of observations are different, at sufficiently large number of observations it is possibly, with a certain probability, to determine the law of distribution of the observations, that is searched parameter.

Linear cryptanalysis combines 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 key, and uses these links to determine by the statistical methods some bits of the key. This method uses a 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, 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 *1/2*.

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 of 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, 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 procedure of attack on the algorithm is similar to the procedure that is used in the impossible differentials.

There is another kind of attacks - 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 related to each other similarly:

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

Having the pairs of the plaintexts and ciphertexts, which are related by only one round of the encryption, the round key can be obtained, that will allow to disclose the full algorithm. Slide-pair is defined as follows [7]:

- for the text P and for the 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} variants of the key. Verification of imitative sustainability was carried out by determining the impact of 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 was investigated the impact of change of one of the bits of the key or the plaintext block to the ciphertext. 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 and pseudo randomly generated vector.

Sustainability against attacks, based on the methods of differential and linear cryptanalysis, consists in difficulty of determining the key knowing the input and output values of the round functions. The functions of complicated transformation are designed in such 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, then the part of the input data, which enters to the input of such an operation is rejected and therefore performance of the inverse operation is impossible. At best case, it can be got certain dependencies (26) for each of the round keys, but not the key in an explicit form. Knowing these dependencies you can distinguish the part of bits of the key, but not completely the 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.

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

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

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. Selection of bits, on which shift is performed, is providing 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 the key value for each concrete case, because there are no operations performed only with the input value. All operations are dependent on input value as well as on the key, that is why 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 cryptographic algorithm using statistical tests.

Since any encryption algorithm can be considered as 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.

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 sequence is not considered random and for successful passing of tests the probability should be higher than 0.01 [11].

According to the results of passed tests it was obtained variables of *p-value*, which are greater than 0.05, that is quite a good result.

Table 1

NIST test results for cryptographic algorithm [11]

№	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

6. Conclusions

In the article the methods and approaches of the construction of block ciphers have been analyzed, several issues related to the security of 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.

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

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

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

References

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

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

[3] Viet Tung Hoang, Phillip Rogaway. On Generalized Feistel Networks. Dept. of Computer Science, University of California, Davis, USA. 2010. – 26 p.

[4] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," Journal of Cryptology, vol. 4, no. 1, pp. 3-72, 1991.

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

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

[7] M. Ciet, G. Piret, J. Quisquater (2002). Related-Key and Slide Attacks: Analysis, Connections, and Improvements.

[8] Matt Blaze, Whitfield Diffie, Ronald L. Rivest , Bruce Schneier, Tsutomu Shimomura, Eric Thompson, Michael Wiener. Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security, 1996.

[9] Mark Stamp, Richard M. Low. 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.