Differential Ananlysis of 3 Round Kuznyechik

Evgeniya Ishchukova
Department of Information Security
Taganrog Institute of Technology
Southern Federal University
Chekhova str., 2, 347928, Taganrog
Russia
uaishukova@sfedu.ru

Ekaterina Tolomanenko
Department of Information Security
Taganrog Institute of Technology
Southern Federal University
Chekhova str., 2, 347928, Taganrog
Russia
kat.tea@mail.ru

Ludmila Babenko
Department of Information Security
Taganrog Institute of Technology
Southern Federal University
Chekhova str., 2, 347928, Taganrog
Russia
blk@tsure.ru

ABSTRACT

In January 2016, a new block encryption standard came into force in the Russian Federation — GOST R 34.12-2015. It includes two algorithms of encryption. The first cipher was previously known under the name GOST28147-89 (or simply GOST). The second algorithm was called Kuznyechik. Kuznyechik is a new symmetric encryption algorithm, based on the SP-network. Up to now there are no publications about the differential properties of the algorithm Kuznyechik. We are the first to examine the properties of main operations and suggest a scheme of 3 rounds differential analysis of cipher Kuznyechik. We examined the differential properties of the non-linear transformation S and the linear transformation L and found out that it's possible a situation when, 1 non-zero byte difference, being a result of the transformation L, is expanded into 16 non-zero bytes, then it passes through the S-boxes, and then collapses again into 1 non-zero byte. The developed scheme allows to affect the active S-boxes a minimum number of times. As a result, for the suggested scheme the possibility of finding the correct pairs of texts is equal to 2^{-108}. We also developed the algorithm of finding a secret key, the complexity of which is equal to $6*2^{-120}$. In this way, the total complexity of the analysis, including searching for the correct pairs of texts and bits of the secret encryption key is equal to $2^{108} + 6*2^{120}$ encryptions.

Also the article contains theoretical calculations of the time required to implement an attack using the most powerful supercomputers in the world [2].

CCS CONCEPTS

• Theory of computation~Cryptographic primitives

ACM Reference format:

SIN '17, October 13–15, 2017, Jaipur, IN, India © 2017 Association for Computing Machinery. ACM ISBN 978-1-4503-5303-8/17/10…$15.00
https://doi.org/10.1145/3136825.3136880

KEYWORDS

GOST, cipher Kuznyechik, GOST R 34.12–2015, differential analysis, block cipher

1 INTRODUCTION

The encryption algorithm Kuznyechik was chosen as the standard GOST R 34.12-2015 and officially came into force on January 1, 2016, that's why today the research of its reliability is a relevant objective. The description of Kuznyechik, as a part of the new encryption standard, is contained in document of the technical standardization committee TC26 «Cryptographic protection of information» in GOST R34.12–2015 [1]. There are several articles, dedicated to various ways of realization of the algorithm Kuznyechik, including using of the special precomputed tables [2].

Up to date you can already find several studies dedicated to the analysis of the reliability of the cipher Kuznyechik. At the international conference CRYPTO 2015 Alex Biryukov, Léo Perrin and Aleksei Udovenko presented a report, in which they shown that, despite the developer's statements, the values of S-boxes of cipher Kuznyechik and hash function Stribog aren't (pseudo) random numbers, but they are generated on the basis of hidden algorithm, which was restored by them using the methods of reverse engineering [3, 4]. Riham AlTawy and Amr M. Youssef described the meet-in the middle attack on five rounds of cipher Kuznyechik, that has a computational complexity of 2^{140}, and requires 2^{153} of memory and 2^{113} of data [5]. In the work [2] are discussed the approaches to the analysis of the algorithm Kuznyechik using a slide-attack. But it was also shown that the considered degree of self-similarity can never be achieved because of the function used in cipher for the generation of round sub keys. Nowadays there are no publications with information about the differential characteristics of encryption algorithm Kuznyechik, and also the publications that describe the possibility of using of this analysis method with minimized number of Kuznyechik rounds.

The method of differential cryptanalysis for the first time was proposed by Eli Biham and Adi Shamir, who applied it to the analysis of the encryption algorithm DES [6, 7]. In their studies, they showed that the algorithm DES is quite robust to this method of cryptanalysis, and any smallest change of the algorithm's structure makes it more vulnerable. However, they still managed to reduce significantly the complexity of analysis of the algorithm DES, reducing it from 2^{56} to 2^{36}. But

requirement of the same number of the open and encrypted pairs of texts (2^{36}) left this attack theoretical for cipher DES. Despite this, the method proved to be very successful. It can be implemented not only to the symmetrical blocking ciphers, but also to the stream ciphers [8], and to the hash-functions [9].

In the present article, we suggest a method of constructing of three-round differential for encryption algorithm Kuznyechik. The developed analysis scheme is based on the use of differential properties of S and L transformations of the algorithm Kuznyechik, and is designed to be able to determine the correct pair of texts for further analysis, which scope is the determination of secret encryption key. The scheme is developed to affect the active nonlinear components (S-boxes) a minimum number of times. As a result, for the suggested scheme the probability of finding the correct pairs of texts is equal to 2^{-108}. Also, we developed an algorithm of finding a secret key, the complexity of which is equal to $6*2^{-120}$. In this way, the total complexity of the analysis, including searching for the correct pairs of texts and bits of the secret encryption key is equal to $2^{108} + 6*2^{120}$ encryptions. The complexity of the suggested scheme is quite high in comparison with possibilities of modern computational tools, but it is much lower than the complexity of key compromising with the full enumeration method.

The article is organized in the following way. The first part of the article contains the description of the encryption algorithm Kuznyechik. The second part contains the study of differential properties of three main transformation of cipher Kuznyechik: exclusive-or operation, nonlinear replacement S-box, linear transformation L. The third part contains the description of constructing process of differential characteristic for three rounds of the cipher Kuznyechik. The key point of this section isn't the separate use of the differential properties of considered transformations, but a study of the work of these transformations in their entirety. Precisely this research allows to construct the three-round characteristics in such way, that in the process will be involved a minimum number of S-boxes, in that way providing the maximum probability value. The fourth part describes the developed and implemented algorithm for finding the right pairs of texts for analysis of the cipher. The fifth part describes the process of key searching due to use of the developed algorithm of three rounds analysis of Kuznyechik. The sixth part contains theoretical calculations of the time required to implement an attack using the most powerful supercomputers in the world, using both simple implementation and implementation with the use of special precomputed tables.

2 DESCRIPTION OF ENCRYPTION ALGORITHM KUZNYECHIK

The encryption algorithm Kuznyechikis a symmetric block cipher with a block length equal to128 bits and key length equal to 256 bits. The cipher Kuznyechik is developed on the principle of SP-network, which allows to perform the transformation with entire input block, and not only over some part of it.

The encryption process consists of nine rounds, and every round includes three transformations: exclusive-or (xor) with round key, substitution with S-boxes (S) and linear transformation (L). One round of encryption is shown in the Fig. 1.

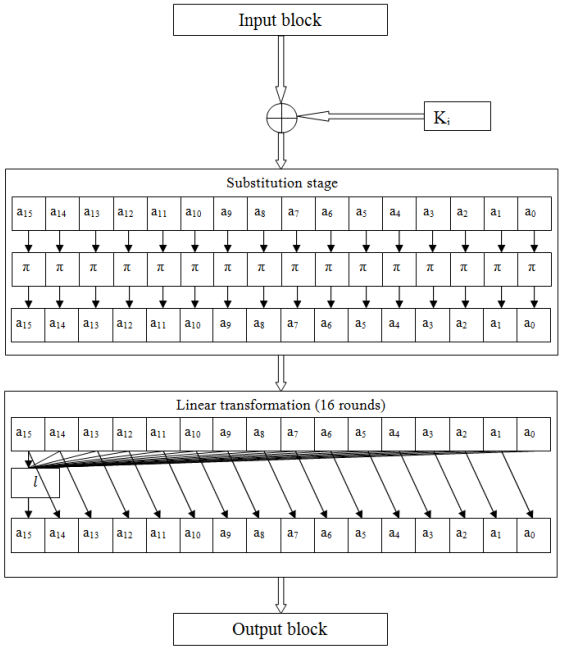

Figure 1: One round of encryption

The original secret key is used for round subkey generation. Its value forms first round subkeys: K1 consists of its most significant bits and K2 consists of its least significant bits. Each subsequent key pair is computed by Feistel scheme. In each round are taken: key xor with a round constant, transformation with a substitution block, linear transformation. Constants used in each round are obtained from the application to the round's number a linear transformation L. A detailed description of the cipher's work, and numerical examples which model the encryption process, can be found under the link [1].

The decryption function is executed in the reverse order, using the inverse transformations L^{-1} and S^{-1}.

3 DIFFERENTIAL ANALYSIS OF THE MAIN COMPONENT'S PROPERTIES OF KUZNYECHIK

Before we begin to consider the differential proprieties of main components of the cipher Kuznyechik, we will speak about several notions and indications, which we will operate in future. The method of differential cryptanalysis is investigate texts not separately, but in pairs. It's necessary to control how is changing the difference of two texts during their pass through the encryption rounds. As a measure of difference is considered the result of exclusive-or operation, which is called the **differential**. Thus, a pair of input texts X and X' is called the **input differential** and is denoted like $\Delta X=X \oplus X'$. A pair of output texts Y and Y', is called the **output differential** and is denoted like ΔY. By a differential characteristic, we mean the most

probable pair of values of the input and output differentials. The **right pair of texts** is a pair (open encrypted text): (X, Y) and (X', Y') for which ΔX=X⊕X' and ΔY=Y⊕Y'. The **inactive S-box** is a replacement block, through which passes the value of difference, which is equal to zero. The **active S-box** is a replacement block, through which passes the value of difference, which isn't equal to zero. The task of differential analysis is come down to the selection of a combination of input and output differentials, for which the total number of active S-boxes would be minimum (but nonzero) passing through all rounds of encryption. So, first, let's consider the differential properties for every transformation.

Exclusive-or operation with a secret key (xor). E. Biham and A. Shamir have already shown that the key can't affect the difference. This is due to the fact that we're considering two parallel encryption processes, where is used the same secret key. This means that due to the properties of exclusive-or operation, the bits of secret key will be excluded – X⊕Ki⊕X'⊕Ki. =X⊕X' = ΔX.

Linear transformation L. The transformation L in this algorithm is linear. This means that it transforms the value of difference with a probability equal to 1. Like the transformation of MixColumnin the encryption algorithm AES, it mixes all the bits of initial state. Therefore, if on the input to the L transformation you had only one nonzero byte, then at the output of L transformation all the bytes will be nonzero. It's easy to show that for the L transformation is realized the following property:

L(a)⊕L(b) = L(a⊕b)

Nonlinear transformation S. Let's denote the differential on the input of the S-box like ΔA, and the differential at the output of S-box let's denote like ΔC. For the algorithm's analysis, it is necessary to complete the table, which will display the possibilities of appearance of different values of ΔC for the determined value ΔA. It is possible to find the algorithm of the such tables creation in the studies of E. Biham and A. Shamir [6, 7]. The resulting table for Kuznyechik is big, and contains 256 rows and 256 columns. One part of the obtained table of probabilities is shown in the Table 1.

Table 1: Part of the table of difference's probabilities, created for the S-box

ΔC / ΔA	0	1	…	3f	…	ff
0	256/256	0/256	…	0/256	…	0/256
1	0/256	0/256	…	4/256	…	2/256
2	0/256	4/256	…	0/256	…	2/256
3	0/256	2/256	…	2/256	…	0/256
…	…	…	…	…	…	…
a5	0/256	0/256	…	2/256	…	0/256
a6	0/256	4/256	…	0/256	…	2/256
…	…	…	…	…	…	…
fe	0/256	0/256	…	0/256	…	0/256
ff	0/256	2/256	…	4/256	…	0/256

The rows of the Table 1 denote the input values of differential ΔA in S-box, and columns denote the corresponding output values of the differential ΔC, obtained at the output of S-box. At the intersection of rows and columns is showed amount of the differentials ΔA/ΔC that have a given input and output difference. As a result of the analysis was established that the probability of matching for values ΔA/ΔC can be equal to 2/256, 4/256, 6/256 and 8/256, and also can be equal to zero. Herewith a nonzero value of the difference at the input of the S-box will be never transformed into a zero difference at the output.

If the analysis succeeds in finding the right pair of texts, then this gives us a possibility to assume, that the differences have passed through the encryption rounds exactly as we have considered them during the construction of round differentials. The combination of differentials ΔA and ΔC allows to suppose the values A⊕Ki and A'⊕Ki. With known A and A' it allows to determine Ki.

4 CONSTRUCTION OF DIFFERENTIAL CHARACTERISTIC FOR THREE KUZNYECHIK ROUNDS

So, in the previous section it was noted that due to the byte mixing because of using of the transformation L, if at the input at the transformation L there was only one nonzero byte, so at the output of the transformation L most probably all the bytes would be nonzero. It means that if at the first encryption round will be used one active S-box, then in the second round of encryption, after transformation L of first round, all 16 blocks will become active. This means a sharp decrease of the probability value when transforming the values of difference in the S-box. Even if in accordance with the Table 1 consider the highest value of probability, equal to 8/256 = $\frac{1}{2^5}$, with 16 active S-boxes the probability value will change in $\left(\frac{1}{2^5}\right)^{16} = \frac{1}{2^{80}}$ times. And this means, that in the next round, most likely, will be also involved all 16 S-boxes. Even with the best value of probability of each considered S-boxes, every round will reduce the probability value in $\frac{1}{2^{80}}$ times. In this way, after three rounds the probability of differential will be equal to (under the best probabilities, which also requires verification) $\frac{1}{2^5} * \frac{1}{2^{80}} * \frac{1}{2^{80}} = \frac{1}{2^{165}}$. After the fourth round the probability will approach the probability of exhaustive search and will be equal to $\frac{1}{2^{245}}$, which will make useless the future analysis. We want to note, that these probability values have been calculated considering the

best values of probability. Although in practice it is still necessary to check whether they can be achieved simultaneously. Nevertheless, such an approach in the calculation of probabilities makes it clear that these calculations are pointless.

Our idea was based on using the properties of S and L transformations together. Initially there was a hypothesis. We had an idea that if one active S-box in the first encryption round will lead to the 16 active S-boxes in the second encryption round, than in certain cases we can get only one active S-box in the third round again. Schematically this can be represented as it done in the Fig. 2.

It was shown above that mixing data with a round subkey doesn't change a differential. In this case, our idea of three-round differential's constructing is the following.

First round: At the algorithm's input is given an input differential ΔX, that have all bytes except one with zero values. The same differential value is given at the input of nonlinear bi-active transformation S, where it is changed into another differential in accordance with the probability table (Table 1). In this way, we get one active S-box. The differential at the output of the transformation S as before contains one nonzero byte. After this the differential value is given at the input of linear transformation L, as a result of which at the output we get the differential value with all 16 nonzero bytes.

Second round: The differential, obtained from the L transformation of the first round, remains unchanged after adding to the second-round subkey. Therefore, for nonlinear transformation S of a second round all the 16 bytes will be active. As a result of work of the S-boxes each byte of the differential can be converted into the other value in accordance with the probability table. At the input of linear transformation L of the second round is given the difference value, that has all16 bytes with nonzero value. After the L-transformation these 16 nonzero bytes will be converted at the differential value, that contains only one nonzero byte.

Third round: The value of the differential obtained in the second round is mixed with the third-round key, which by analogy with first and second rounds doesn't affect the differential value. Therefore, in the S-transformation we get only one active S-box. Nonzero byte of differential in accordance with the probability table is changed into other probable value. Then, as a result of the transformation L, the difference with one nonzero byte is converted to a difference value, which again consists of 16 nonzero bytes.

Therefore, for the considered scheme of analysis we obtain only 18 active S-boxes instead of 33 as was considered before. Even if we assume that we can select the differentials that will correspond to the Fig.2, with the minimum probability values for each of active S-boxes $2/256 = \frac{1}{2^7}$, we will get a three-round differential with probability equal to $\left(\frac{1}{2^7}\right)^{18} = \frac{1}{2^{126}}$, what is less, that the value assumed before $\frac{1}{2^{165}}$. Looking ahead, we want to say that we have managed to get a three-round characteristic, the appearance probability of which is equal to $\frac{1}{2^{108}}$, what means, that during the use of S-boxes were involved not the most minimal probabilities.

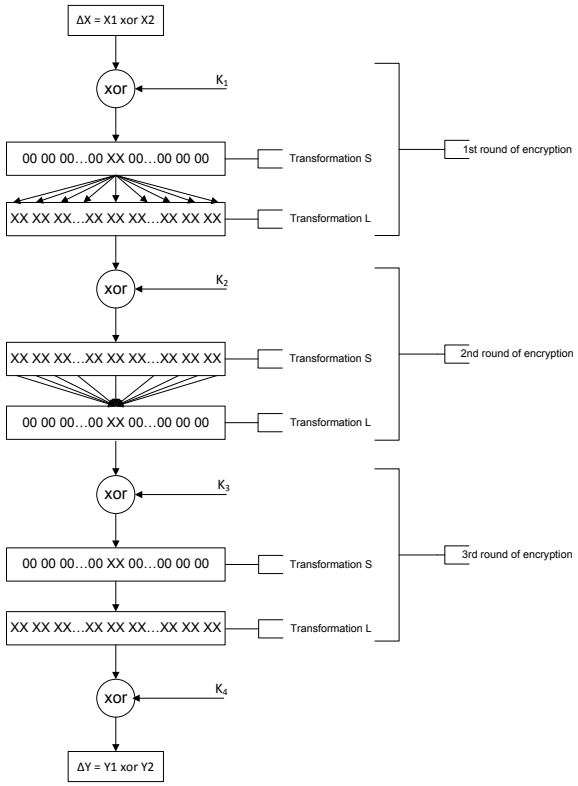

Figure 2: Scheme of the differential analysis of three encryption rounds

So, in order to construct a differential, the scheme of which is shown in the Fig. 2, we need to consider of work of the S and L transformations together. In a general form, this can be represented schematically as done in the Fig. 3. It's necessary that in the first round one nonzero byte of differential after L can be converted into 16 nonzero bytes (left part in the Fig. 3). In the second round it's necessary, that 16 nonzero bytes of differential after L can be converted into 1 nonzero byte. Or, if we go from the opposite, we are interested in such cases when one nonzero byte of differential after L^{-1} can be converted into 16 nonzero bytes (right part in the Fig. 3). Herewith, the differentials of the right and the left parts must be connected with S transformation, in other words, in the probability table must be presented at least a minimum probability (2/256) of difference transformation from the left part of the Fig. 3 to its right part.

Thus, the suggested scheme of relations between S and L transformations implies, that through the transformations L and L^{-1}, it's necessary to pass sixteen different variants of filling one byte from 00 to FF, which as a result of this transformation will be decomposed into 16 bytes. Let's formulate the suggested algorithm in steps.

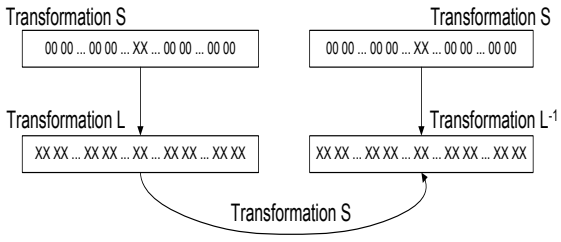

Figure 3: Relation between transformations S and L

First step. At the input of the transformation L is given a block, where only one byte has a nonzero value. Through the L-transformation will pass 16different variants of every possible byte's filling from 00 to FF. Each one nonzero byte value will be decomposed through the transformation L into a value, that contains 16 nonzero bytes. As a result, we get probable differential at the input of the S-transformation of the second round. The total of these values will be 255*16=4080. These are probable values at the output of the L-transformation of the first round (left part of the Fig. 3). The same values will be given at the input of the S-boxes of the second encryption round. Let's denote them like ΔA.

Second step. Is similar to the first step, except that instead of L-transformation is used a transformation L^{-1}. In this way, we get other 4080 values. These are probable values at the input of L-transformation of the second round (right part of the Fig. 3).The same values will be given at the input of the S-boxes of the second encryption round. Let's denote them like ΔC.

Third step. As a result of the first and second steps, we have the differentials ΔA and ΔC at the input and output of the S-transformation of the second round. As for each value ΔA not all the values ΔC are the same, we must determine, what differentials ΔA that are given at the input of the S-boxes from 4080 values will have a nonzero probability to be obtained as ΔC values due to transformation. For this it's necessary to decompose each of 4080 values ΔA into 16 bytes and put them at the input of the probability table. Then in the same way we must decompose into 16 bytes each value ΔC. For some 16-bytes differentials ΔA from 4080 values in accordance with the table will be found the differentials ΔC with nonzero probability. In total, as a result of considering all possible combinations, we obtained 13 pairs of values ΔA/ΔC, which are shown in the Table2.

From the found values ΔA/ΔC it's easy to determine the values of differentials at the input and output of 3-round algorithm Kuznyechik ΔX/ΔY, and the probability of their appearance. For the first pair of values ΔA/ΔC with the highest probability equal to $\frac{6}{256}$ at the algorithm's inputis given a value ΔX = 00000000002000000000000000000000, and at the output are obtained three values with the same probability

ΔY1 = e07d0b92e148a1f4774154a6b06e99f0,

ΔY2 = 0d38eb47ba753eb564333cbda803ac2c,

ΔY3 = acb714a9647d96d76b3a89ab83f1bd71.

Table 2: Found pairs of values ΔA / ΔC

ΔA	ΔC
f3ab8c55c199996fc5a4f2381976846	51ac91f0df2470190ad86a256131163
1a76bc71665284b01a3e595982599369	ba5a9d5e6d2b6431ac6b9cb72dc5a7a1
5cfbaa318fd91c774940bef22a5f86	1f8355405427f8e7d8c71cc07f2288c6
ac8ea817121caa3445efd4b9c43e875f	c6e355f95177d1bd58f9a4145283a143
522cbe6cbcf88eaf4963f28f8f29e62	353cceb5eab273db5790cc909cfa9
2bc22a75e57cbd804b35bf31ee5167	54be1b26f4dbe5b1f6a2a66a61e384d
5f28ebaf31b588b3f8f23923e399f0ef	b4eba9c9151b1fbd907f4f4d419fcdaa
7dbda6246e653ba46ec427fafa9a462f	5d87c43030f77ec08a9f25c0b8413318
efd9d1a17499185749edf1d1d6ee4c	bbd9f4a6c11bf5e154a2c52495e1ddd9
8fb8e26a91ccf9b72d6d5dce4f9ad4a9	b86cc61926ba16e11d19dce66e78450
6a7c998e18379bf720d423721d3c7e63	c6fc783ae4466b402df56e30dff1f8d4
337ddbfefc424a38d45ae559c2d2cd36	fd58a4739c68ed296855b6723e9fb3
5c9ce97665a2afd9172a54a881949c	3a3d1b8c2658ea7f8a958b2f866ecb5b

5. ALGORITHM OF FINDING THE RIGHT PAIRS OF TEXTS

In general case, if you know the value of the differential and its probability, then in the search algorithm for the right pairs of texts there is nothing difficult. The algorithm of searching for the right pairs of texts is following:

1. Randomly generate the text X.
2. Determine the text X': X' =X⊕ΔX.
3. Encrypt the text X, obtain the cipher Y
4. Encrypt the text X', obtain the cipher Y'
5. If ΔY=Y⊕Y', then (X, Y) and (X', Y') – is a right pair of texts, otherwise turn to step 1.

The complexity of the analysis will be determined by the probability of finding such differentials. We have made theoretical calculations of how much time will be spent on searching for one correct pair of texts using the most powerful supercomputers in the world. These calculations will be described in the section 7.

Whatever it was, for us it's impossible to perform such computational calculations. But in order to confirm the efficiency of the proposed differential scheme for three rounds of encryption we decided to go «from the reverse». Knowing how differentials should be converted when passing through the rounds of encryption, not by searching, but selecting the texts and secret keys, that will correspond to the obtained round characteristic. It was done with only purpose – to show that such values of text exist, and the suggested scheme is workable.

As can be seen from the Table 2, there were found 13 pairs of values ΔA/ΔC. From these values it's possible to determine, which values of the differentials were given at the input of the

encryption algorithm, and which will be obtained at the output. For this it's necessary to perform some reverse conversions.

First step. Searching for the open texts. ΔA is given at the input of L^{-1} transformation, reversed with transformation L of the first round, as a result $L^{-1}(\Delta A)$ we get an output of the S transformation of the first round. Then, the previous result is given at the input of S^{-1} transformation of the first round, reversed with transformation S, as a result $S^{-1}(L^{-1}(\Delta A))$ we get a differential given at the input of the algorithm.

Second step. Searching for texts based on three rounds. ΔC is given at the input of transformation L of the second round, and we get L(ΔC). Then the obtained result is given at the input of the S-boxes of the third round, where as a result of rearrangement of S in accordance with the Table 1, we get S(L(ΔC). Ultimately, the last obtained result as given at input of the transformation L, where is changed in accordance with the algorithm. In this way, we get a necessary output, which contains there searched differential ΔY.

If find the bytes, which constituting the differentials ΔA/ΔC, compile the texts composing them and pass them through three-rounds transformations, then for certain values of a secret-key, according to the developed and described method, the correct pair of texts will form the required values of the differential. To prove the operability of the proposed method, were obtained the initial values, some of which are shown in the Table 3 and the final values, some of which are shown in the Table 4. Where A and A' are the texts, constituting the found differential ΔA, so C and C' are the texts composing the differential ΔC. If consider the master key like

K=
8899aabbccddeeff0011223344556677fedcba98765432100123456789abcdef,

than round keysK1, K2, K3 andK4 will be equal to following values:

K1=8899aabbccddeeff0011223344556677;
K2=fedcba98765432100123456789abcdef;
K3=db31485315694343228d6aef8cc78c44;
K4=3d4553d8e9cfec6815ebadc40a9ffd04.

Table 3 – Initial texts

K1⊕(Sinv(Linv(A⊕K2)))	K1(Sinv(Linv(A'⊕K2)))
8998079702eaede8422655046852245d	8998079702fbede8422655046852245d
8998079702eaede8422655046852cf5d	8998079702fbede8422655046852cf5d
8998079702eaede8422655046852825d	8998079702fbede8422655046852825d

Table 4 – Values based on three rounds

K4⊕(L(S(C)))	K4⊕(L(S(C')))
8ccf9f63b68216e7e87fb65cfce4364f	27ac6bf0c98ce6b78ed52e43f34d17cb
7152f68de9e8354c0db9cb0594ee651e	da31021e96e6c51c6b13531a9b47449a
8ef7bcaec365c4890540ec622239d595	2594483dbc6b34d963ea747d2d90f411

6. ALGORITHM OF SEARCHING FOR THE SECRET KEY BASED ON THE ANALYSIS OF THE RIGHT PAIRS OF TEXTS

Considering three rounds of encryption, we use 4 round keys (Fig. 2). A feature of the algorithm Kuznyechik is that the first two round keys are the halves of a 256-bit master-key, on the basis of which are produced other round subkeys. Considering the fact, that suggested algorithm provides for finding of key K1, and then key K2, from which are produced other keys K3 and K4.The scheme of suggested algorithm is shown in the Fig. 4.

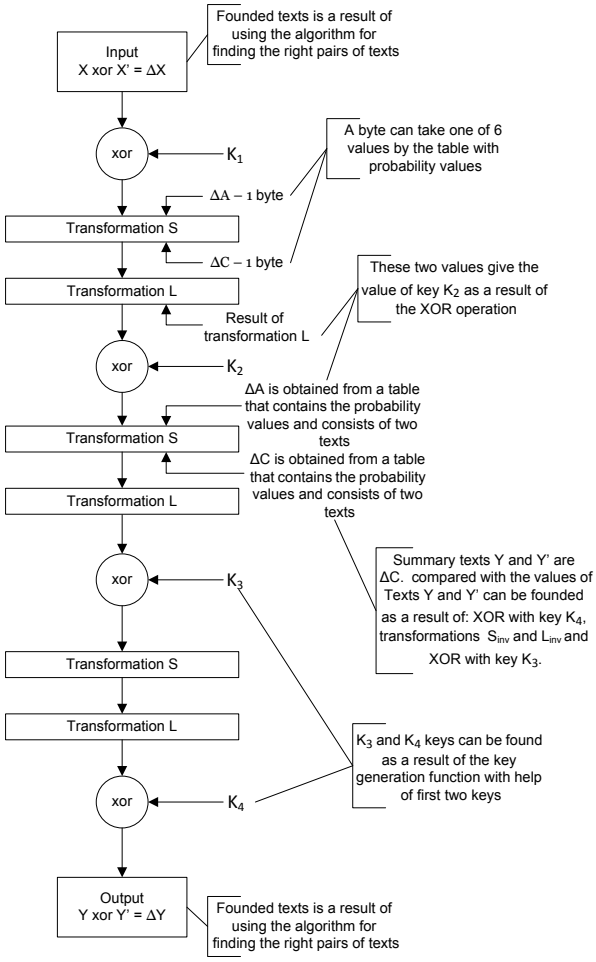

Figure 4: Algorithm of searching for key bits

As it was said in the 4th part of this article, when performing reverse operations on the found ΔA, more specifically $S^{-1}(L^{-1}(\Delta A))$, we get a differential, which is given to the input of the transformation S of the first round, and with $L^{-1}(\Delta A)$ we get the output of the same transformation. In this way, if give to the input of the algorithm, shown on the Fig. 2, the previously selected pair of texts X and X', than these texts while adding to a value, obtained from $S^{-1}(L^{-1}(\Delta A))$, give the value of the key K1. So as the value of $S^{-1}(L^{-1}(\Delta A))$ consists from one nonzero byte (8 bit)

and has a probability equal to 6/256 in accordance with the probability table, than it gives us 6 probable values of one byte of the key K1. Therefore, a number of all possible variants of 128-bit keyK1 is equal to $2^{120}\times6$.

For checking of one possible variant of key K1, it's necessary to perform several operations. Proceeding from the Fig. 4, first we give at input two selected texts X and X', and then add them with one possible key K1 from the total number of possible keys K1. Then the obtained values pass through the S transformation of the first round, and then through the L-transformation. The transformation results are added with values, composing the differential ΔA, that is given at the input of the S-boxes of the second round. On this stage, we get one of possible values of key K2. Then we put together the keys K1 and K2, and obtain the 256-bit master-key, from which we produce the keys K3 and K4.

The next step is to take two possible texts Y and Y', obtained as a result of using algorithm of finding of right pairs of texts for the differential cryptanalysis, and add them to the key K4. Now, obtained values are changed with L^{-1} and S^{-1} transformations of the third round and then they are added to the key K3. Then the same values are changed with L^{-1} transformation of the second round and are compared with the values, composing the differential ΔC. If the values are equal, then the keys are saved as probably correct. If the values are not equal, then the algorithm of the actions is repeated anew: two texts X and X' are added to the next key K1, from all possible values of K1. And all the future actions are performed on the same scheme. In fact, key searching consists of all operations of three-round encryption. That's why in total the algorithm will use a maximum of $2^{120}\times6$ encryptions. In this case all 4 keys will be found, which is much smaller number, than an exhaustive search of the key material, where the total possible number of keys represents a number equal to 2^{128} for only first key K1, a number of values for searching for the second key K2 by the full search is also equal to 2^{128}, which in general makes it possible to estimate the complexity of searching for the key by the full search as 2^{256}.

Therefore, the complexity of keys finding by developed method of cryptanalysis is equal to $2^{120}\times6$ of 3-round encryptions. The total complexity of the performing of differential cryptanalysis of three rounds of the encryption can be estimated as $2^{120}\times6+2^{108}$ of 3-round encryptions.

7. AN APPROXIMATE ESTIMATE OF THE TIME REQUIRED TO ANALYZE THE ALGORITHM KUZNYECHIKON MODERN SUPERCOMPUTERS FROM TOP500

Performing operations based on the developed algorithm of differential cryptanalysis of three rounds of the algorithm Kuznyechik, we can estimate the approximate time costs required for the analysis, for example, on modern supercomputers from TOP 500. For evaluation were selected first 10 computers from the overall rating [10].

Table 5 shows our analysis time calculations for the most powerful supercomputers in the world using the conventional cipher implementation and using special precomputed tables.

The processing time of 1 data block by the processor using 3 rounds of cipher is calculated approximately in accordance with the clock speed of the supercomputer and is based on our experimental data on the measurement of the encryption rate, and also on the basis of data obtained in the work [2].

The resulting analysis time can be resolved by multiplying the analysis complexity by the processing time of one block (using one of the implementations). You can see that even using the most powerful supercomputer TOP500, this analyze remains a theoretical exercise for the cipher Kuznyechik.

Table 5 – Calculation of time costs

№	Supercomputer	Characteristics of supercomputer	Processing time of 1 data block 1, sec (Normal)	Processing time of 1 data block 1, sec (precomputed tables)	Complexity of analysis
1	Sunway Taihu Light	10,649,600 cores, 1.45GHz	0,0027	0.0000018	$2^{85}+6\times2^{92}$
2	Tianhe-2 (MilkyWay-2)	3,120,000cores, 2.2GHz	0.0018	0.0000012	$2^{87}+6\times2^{99}$
3	Piz Daint	361,760cores, 2.6GHz	0.0015	0,000001	$2^{90}+6\times2^{102}$
4	Titan	560,640 cores, 2.2GHz	0.0018	0.0000012	$2^{89}+6\times2^{101}$
5	Sequoia	1, 572,864 cores, 1.60 GHz	0.0025	0.0000017	$2^{88}+6\times2^{100}$
6	Cori	622,336 cores, 1.4GHz	0.0028	0.0000018	$2^{89}+6\times2^{101}$
7	Oakforest	556,104cores, 1.4GHz	0.0028	0.0000018	$2^{89}+6\times2^{101}$
8	K computer	705,024cores, 2.0GHz	0.0019	0.0000013	$2^{89}+6\times2^{101}$
9	Mira	786,432 cores, 1.60GHz	0.0025	0.0000017	$2^{89}+6\times2^{101}$
10	Trinity	301,056 cores, 2.3GHz	0.0017	0.0000012	$2^{90}+6\times2^{102}$

8. CONCLUSION

As a result of work on this project we for the first time studied and obtained the differential properties of the encryption algorithm Kuznyechik. On the basis of our studies was found a connection with S and L transformations, which allowed to

develop the algorithm of differential cryptanalysis of three rounds of the cipher Kuznyechik.

For the suggested scheme of differential cryptanalysis, we developed the algorithm of finding the right pairs of texts for the cipher analysis. On the basis of the three-round differential we developed the algorithm of finding the secret key with less complexity, than complexity when using a key by full searching. The developed algorithms allow to estimate the total complexity of the analysis, which is equal to $2^{108} + 6*2^{120}$.

As a result of the researches was developed and tested the encryption algorithm Kuznyechik, was developed a method of differential analysis of three rounds of algorithm Kuznyechik, and for this algorithm was created a scheme of finding the correct pairs of texts for the differential analysis of three round of encryption. The results of the researches are described in detail, are shown in examples, are structured in chronological order and are visually displayed in the form of illustrations, tables and diagrams in the text of this article. The theoretical calculations were also performed on the time required to conduct an attack using the most powerful supercomputers in the world both using the normal implementation and implementation with use of special precomputed tables.

ACKNOWLEDGMENTS

The results of the work were used in the performing of research works under the grant RFBR N°17-07-00654-a «Development and studying of sequential and parallel analysis algorithms».

REFERENCES

[1] Cryptographically protection of information Block ciphers GOSTR 34.12–2015 [Internet source].– URL: http://tc26.ru/en/standard/gost/GOST_R_34_12_2015_ENG.pdf

[2] E.A. Ishchukova, L.K. Babenko, M.V. Anikeev Fast Implementation and Cryptanalysis of GOST R 34.12-2015 Block Ciphers // 9th International Conference on Security of Information and Networks, SIN 2016, Newark, Nj, 20-22 July 2016. – P. 104 – 111.

[3] Alex Biryukov, Léo Perrin, Aleksei Udovenko. Reverse-Engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1 (Full Version) (2016). https://eprint.iacr.org/2016/071.pdf

[4] Alex Biryukov, Léo Perrin, Aleksei Udovenko. Reverse Engineering the S-Box of Streebog, Kuznechik and Stribob (2015). http://crypto.2015.rump.cr.yp.to/1ea2c6c01144e0e7f6b14b324c5e4562.pdf

[5] Riham AlTawy and Amr M. Youssef A Meet in the Middle Attack on Reduced Round Kuznyechik (April 17, 2015) https://eprint.iacr.org/2015/096.pdf

[6] Biham, E., and Shamir, A., Differential cryptanalysis of DES-like cryptosystems, Journal of Cryptology, Vol. 4, No. 1, pp. 3-72 (full issue), 1991. http://www.cs.bilkent.edu.tr/~selcuk/teaching/cs519/Biham-DC.pdf

[7] Biham, E., and Shamir, A. Differential cryptanalysis of the full 16-round DES, Advances in cryptology, proceedings of CRYPTO'92, pp. 487-496, 1992.

[8] Eli Biham, Orr Dunkelman, Differential Cryptanalysis in Stream Ciphers // Cryptology ePrint Archive, Report 2007/218, 2007. http://eprint.iacr.org/

[9] Eli Biham, Adi Shamir Differential Cryptanalysis of Hash Functions // Differential Cryptanalysis of The Data Encryption Standard, Springer, 1993, ISBN 978-1-4613-9314-6. – P. 133 – 148

[10] Statistics on supercomputers – TOP 500 [Internet source].– URL: https://www.top500.org/lists/2017/06/