ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 6, Issue 2, February 2019

About differential cryptanalysis algorithm of block encryption "Kuznyechik"

Juraev Gayrat Umarovich, IkramovAlisherAkramovich,Marakhimov Avazjon Rakhimovich

Docent,Candidate of physics-mathematics, Department of Information Security, Faculty of Mathematics,National University of Uzbekistan named afterMirzoUlugbek , City Tashkent, Uzbekistan
Researcher, Lomonosov Moscow State University campus in Tashkent, City Tashkent, Uzbekistan
Rector, Doctor of technical sciences professor, National University of Uzbekistan named after MirzoUlugbek , City Tashkent, Uzbekistan

ABSTRACT: This work is devoted to the study of resistance to the differential cryptanalysis method of the block encryption algorithm Kuznyechik, which has been in force since 2016 as a new state standard of the Russian Federation. To this end, a study was conducted, revealing the most probable differentials of the Kuznyechik algorithm with respect to two and three rounds of attack. Parallel technologies are used to calculate the probabilities of attacks for three rounds. As a result, the high resistance of the Kuznyechik algorithm was shown already in three rounds to a differential cryptanalysis method.

KEYWORDS: Block cipher, strength, differential cryptanalysis, input difference, output difference; characteristic, parallel technology.

I.INTRODUCTION

The new symmetric block encryption algorithm “Kuznyechik” as a cryptographic standard was introduced with the updated algorithm GOST 28147-89to the new standard GOST R 34.12-2015, which entered into force on January 1, 2016 in the Russian Federation [1]. The most detailed description of the Kuznyechik algorithm is presented in the standard GOST R 34.13-2015 [2.]

Kuznyechik is a permutation-permutable block symmetric cryptoalgorithm. This type of cipher is well studied and relatively simple in terms of cryptographic analysis and substantiation of the required properties.

The “Kuznyechik” algorithm - is a symmetric block encryption algorithm with a block size of 128 bits and a key length of 256 bits, which is generated by the Feistel network. The block cipher “Kuznyechik”, like the AES algorithm, consists of repeating single-type rounds, including key addition, linear and non-linear transformation. The length of the 128-bit cipher block will provide additional protection and are supposed to resist any attacks against block ciphers using all modern means.

The code "Kuznyechik" is recognized by developers and some researcher’s resistant to the basic methods of cryptanalysis [3,4]. The strength of this algorithm has been little studied. Serious results are given only in [3,4]. Therefore, the analysis of cryptographic transformations of the Kuznyechik encryption algorithm and the establishment of its durability for possible use in data encryption is of great theoretical and practical importance.

II. DIFFERENTIAL CRYPTANALYSIS

Currently, there are various methods of cryptanalysis of symmetric block encryption algorithms. Many of these methods using distributed multiprocessor computing are well parallelized [5, 6]. As a result, you can reduce the time for analysis several times.

Differential cryptanalysis was proposed by well-known Israeli cryptographers E. Biham and A. Shamir in 1990 [7]. Currently, differential cryptanalysis is one of the main methods of cryptanalysis. Therefore, the cryptographic resistance of many cryptographic algorithms is estimated using differential analysis [6-10]. If an algorithm withstands an attack using this method, then it is considered persistent and without fear it can be used to transmit secret information.

Differential cryptanalysis is an attack with selected open texts, i.e. The cryptanalyst can select input and output texts in order to obtain information about the key. To perform differential cryptanalysis, a cryptanalyst must match pairs of inputs X' and X" corresponding to a specific value of ΔX, knowing that for this specific value of ΔX, a specific ΔY will appear with a high probability.

To use differential analysis, you must first determine the differential characteristics with high probability — a sequence of input and output differentials on the rounds, such that the output differential from one round corresponds to the input differential for the next round. The use of highly probable differential characteristics enables us, using information from the last round of the cipher, to get the bits of the last round keys.

In this paper, we study the attack of a special type of differential cryptanalysis on two and three rounds of the Kuznyechik algorithm. When an attack on three rounds was found, parallel technologies were used to increase the speed of the search.

III. TWO ROUND ATTACK

Due to limited computing resources, it is necessary to build a sufficiently effective attack, the calculation of which can be carried out in a reasonable time. The whole algorithm "Kuznyechik" has 10 rounds, the attack of differential analysis for block ciphers is always carried out one less round.

To begin with, the base attack model was chosen, shown in Figure 1. That is, here the purpose of the analysis is to find such an input differential so that at the output of the second round there is the same differential. Then this attack can immediately spread to any even number of rounds.

To find such a state of the system, moving as the number of rounds increases, it is necessary to perform a search $16\times256\times256^{16}=2^{4}\times2^{8}\times2^{128}=2^{140}$. Such a search is impossible to perform on modern computing devices. Therefore, another approach was chosen: an attack is made on 1 round from the beginning and a reverse attack on 1 round from the end. Thanks to the meeting in the middle, you can reduce the brute force to the doable. Moreover, in the developed attack it is not important Δy, it is necessary to find out only the value Δx, on the basis of which pairs of known open texts will be selected for the attack.

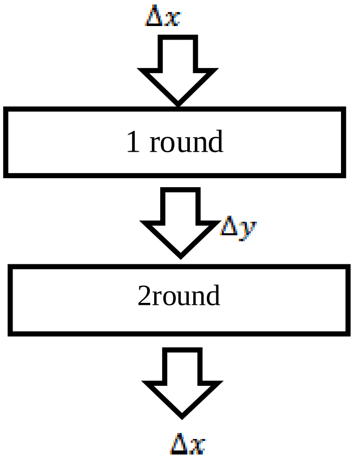

Fig. 1. Kind of a two round attack.

This value should correspond to the maximum possible probability of passing the differential, because based on the probability, the number of text pairs required for an attack is calculated.

To assess the exact value of the probability of passing a given type of attack, the following statement was proved:

Theorem. If the scheme in Fig.1 Δx goes over to differentials Δy_i by the corresponding probabilities p_i, and the differentials Δy_i pass after the second round to a differential with probabilities q_i, then the probability of going from Δx to Δx 2 rounds is calculated by the formula:

$$p=\sum_i p_i\cdot q_i .$$

Evidence. First, we calculate the probability of the chain $\Delta x \to \Delta y_i \to \Delta x$ for an arbitrary i. According to the formulas for differential analysis it will be $p_i \cdot q_i$. Then we need to find $P(U_i(\Delta x \to \Delta y_i \to \Delta x))$. Since every y_i is pairwise different, these events are independent and according to the probability formula the sum of independent events we get:

$$P(U_i(\Delta x \to \Delta y_i \to \Delta x)) = \sum_i P(\Delta x \to \Delta y_i \to \Delta x) = \sum_i p_i \cdot q_i .$$

The theorem is proved.

Table 1. Results of the differential attack search program for 2 rounds of the "Kuznyechik" algorithm

Byte position number	Differential value`	amount
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	213	1048576
12	0	0
13	51	2097152
14	0	0
15	0	0
16	0	0

C / C ++ software was created that implements the following algorithm:
Cycle in all positions **num** bytes (16 total)
Loop over all values of input differential d in byte **num**
Set the **sum** counter to 0
Cycle over all output differentials obtained after nonlinear conversion **dout**
The multiplication of **dout** by the corresponding multiplier of the linear transformation matrix, depending on the byte number **i** of the result. We get the value of **y1**.
Substitution of **d** in the reverse direction of the algorithm. Multiplication by the corresponding factor of the inverse linear transformation matrix for the same byte number **i** of the result. We get the value of **y2**
The calculation of the transition probability of the differential **y2** through the inverse nonlinear transformation to the differential **y1**. Multiply all such results for all numbers **i** from 1 to 16. Add the result of multiplication with the sum and put the sum in **sum**.
End of cycle
If **sum** is greater than **max**, then remember the new maximum.
End of cycle
max output for this **num** byte position
End of cycle

As a result of the program working in sequential mode, the results presented in table 1 were obtained. Here, the number is the values from which you can get the probability by dividing them by the number 256^{17}. Then you can get the input differential of the following form: (0, 0, 0, 0 , 0, 0, 0, 0, 0, 0, 0, 51, 0, 0, 0). Thus, the maximum probability of an attack received is 2^{-115}. So, holding this attack for 8 rounds will have a probability of 2^{-460}, which indicates a serious resistance of the Kuznyechik algorithm to this type of attack.

IV. DEVELOPMENT AND IMPLEMENTATION OF A PARALLEL ALGORITHM FOR CARRYING OUT DIFFERENTIAL ANALYSIS FOR THREE ROUNDS

As noted above, the most effective will be the transition from a two-round attack model to a three-round one (Fig. 2).
A significant difference from the two round model will be a serious increase in the complexity and volume of calculations.
Consider the process of searching for a differential attack for three rounds, taking into account the moves to the forward and reverse sides. The forward move is the calculation of the differential in the direct encryption algorithm starting from the first round. A move in the opposite direction involves the inversion of transformations (since, the algorithm is an SP-network, all its transformations are bijective with a fixed set of round keys) and carrying out differentials starting from the last round. To carry out such an analysis, you need to select the position number of the input byte, the value of the differential on it, the values of the resulting differentials after the first round, since the differential is only on one byte, then after the nonlinear transformation there will be one family of differentials, from which you can multiply by Δy the corresponding linear transformation matrix .

With the help of the reverse stroke, the value of the differentials Δz is calculated. The two round attack of this process ends with a verification of equality $\Delta y = \Delta z$. But, in this case there is another round. Namely, due to it, the enumeration power is increased and the task can no longer be solved without parallel technologies.
The search is performed on the received values Δz. Further, the probability of passing such differentials through one round will be calculated. The total probability Δx for this will be similarly considered as a sum over various Δy and Δz.
It remains to effectively parallelize the resulting algorithm so that the iteration is minimal for each node, and also to reduce the number of messages between nodes (since this is the slowest operation).
If one conducts paralleling by one of the values Δz or Δy, then the result is that the nodes will have to exchange the obtained probability values (they must be multiplied and added). It will be extremely inefficient.

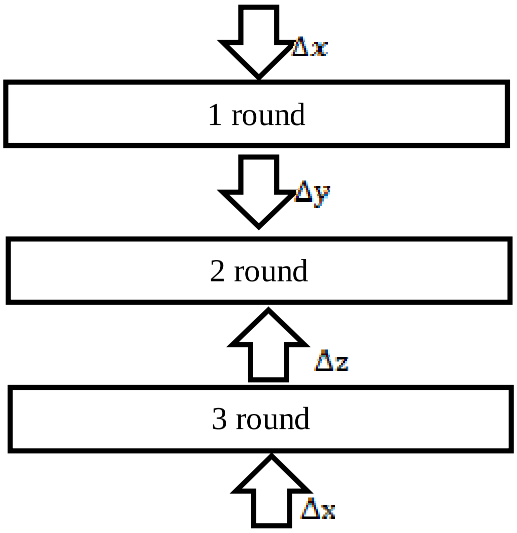

Fig. 2. Type of three round attack.

Therefore, it is best to perform parallelization by value, a maximum is chosen between them and the volume of exchange is only two values - the value of the differential and its total probability. Moreover, to calculate other processes, it will not be necessary to wait for the end of this process, they will simply transmit the resulting total probabilities, and the main node will select the maximum ones from them.

The network architecture of the parallel calculator in this case has a three-level view: the central process (performs the final data analysis, selection of maximum values and their output), 16 groups by byte position number with nonzero differential, processes for finding the probabilities of differential passing with specific values. In each group, it is ideal to have 256 processes (by the number of differential values on each byte) or a smaller power of two for uniform loading.

At the disposal of the authors there was a cluster of 8 two nuclear computers and one server. Each core acted as a separate group, and the number of processes in the group was 1.

A C / C ++ program has been implemented, which is parallelized via the OpenMPI library. On several cores (at 16) the created program was launched and the results of the analysis were obtained (see Table 2). However, since the number of cores for testing was limited, paralleling was carried out by the byte number on which the input differential is not 0. The above idea is suitable for supercomputers or other parallel calculators with a much larger number of cores.

Table 2. Results of the differential attack search program for three rounds of the algorithm "Kuznyechik"

Byte position number	Differential value	Probability
1	117	2^{-193}
2	19	2^{-205}
3	29	2^{-187}
4	75	2^{-212}
5	33	2^{-170}
6	201	2^{-179}
7	193	2^{-183}
8	67	2^{-181}
9	78	2^{-201}
10	97	2^{-185}
11	213	2^{-178}
12	37	2^{-174}
13	51	2^{-192}
14	91	2^{-196}
15	63	2^{-193}
16	77	2^{-195}

As can be seen from table 2, there are no output differentials that appear with a probability greater than 2^{-n} (n is the length of the plaintext block). Thus, differential cryptanalysis seems impossible.

V.CONCLUSION

An attack on two rounds of a special kind of “Kuznyechik” algorithm was built, which allows to continue such a differential attack on an arbitrary even number of rounds.

The analysis was carried out for three rounds of a special type of algorithm that allows you to continue the attack up to the full algorithm (that is, for 9 rounds).

The study showed the high stability of the Kuznyechik algorithm with respect to differential analysis already in three rounds. Thus, differential analysis does not apply to the entire algorithm. The algorithm has greater stability than GOST 28147-89, taking into account the results of L.K. Babenko and Ye.A.Ishchukova [10].

The results of the study can be further used to create more advanced methods for analyzing algorithms based on SP networks. Algorithms using parallel technologies can significantly reduce the time of research. Here, an important issue is the method of dividing a task into subtasks to shorten data exchange intervals.

REFERENCES

[1] Information technology. Cryptographic data security. Block ciphers. URL: tc26.ru/standard/gost/GOST_R_3412-2015.pdf.

[2] Information technology. Cryptographic data security. Block ciphers operation modes. URL: http://www.tc26.ru/standard/gost/GOST_R_3413-2015.pdf.

[3] AlTawy R., Youssef A. Meet in the Middle Attack on Reduced Round Kuznyechik, https://eprint.iacr. org/2015/096.pdf.

[4] Biryukov A., Perrin L., Udovenko A. Reverse-Engineering the S-Box of Streebog, Kuznyechik and Stribob (Full Version). 2016. https://eprint.iacr.org/2016/071.pdf.

[5] Babenko L., Sidorov I. Parallel algorithms for discrete log solving and their effectiveness //Proceedings of the Workshop on Computer Science and Information Technologies (CSIT'2009), Crete, Greece, October 5-8, 2009. Vol. 2. Ufa State Avation Technical University, 2009. – P. 217-222.

[6] Babenko L.K., Ishchukova E.A. Data Distribution Algoritms for Differential Cryptanalysis of DES // Proceeding of the Workshop on Computer Science and Information Technologies (CSIT'2007), Krasnousolsk, UFA, September 13-16, 2007. – Vol. 1. UFA State Aviation Technical University, 2007. – P. 198-201.

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

[8] Biham E., Shamir A. Differential Cryptanalysis of the Full 16-round DES, Crypto'92, Springer-Velgar, 1998. – P. 487.

[9] Biham E., Shamir A. Differential Cryptanalysis of DES-like Cryptosystems, Extended Abstract, Crypto'90, Springer-Velgar, 1998. – P. 2.

[10] Babenko L.K., Ishchukova E.A. Differential Analysis GOST Encryption Algorithm //Proceedings of the 3rd Internaternational Conference of Security of Information and Networks (SIN 2010), ACM. – New York, 2010. – P. 149-157.