Differential Analysis of GOST Encryption Algorithm

Ludmila Babenko
College of Information Security
Taganrog Institute of Technology – Southern Federal University
ul. Chekhova, 2, 347928, Taganrog, Russia
Phone: +7 8634 371905
blk@tsure.ru

Evgeniya Ishchukova
College of Information Security
Taganrog Institute of Technology – Southern Federal University
ul. Chekhova, 2, 347928, Taganrog, Russia
Phone: +7 8634 371905
jekky82@mail.ru

ABSTRACT
In this article we explore the resistance of the GOST 28147-89 algorithm (commonly referred to as GOST) to the attack based on differential cryptanalysis. GOST algorithm is used as a national standard in the Russian Federation. GOST uses variable substitution boxes. It is commonly believed that any values of S-boxes for 32-round GOST encryption algorithm provide sufficient degree of resisting against attacks based on techniques such as linear and differential cryptanalysis.

As the result of our research, we have found out that there is a number of S-boxes with weak properties with respect to differential cryptanalysis. The use of such elements in GOST allows obtaining features that have a fairly high probability that can be used to carry out attacks. So, if we use the same weak block replacement, the probability characteristics for the 32 rounds of GOST can reach 2^{-25}, which makes it relatively easy to get the right pair of texts for analysis.

As the illustration of correctness of our assumptions, we have carried out an attack against 12 rounds of GOST algorithm, which allows us to obtain first round subkey within minutes.

Categories and Subject Descriptors
E.3 [**Data Encryption**]: Standards – *GOST.*

General Terms
Algorithms, Security.

Keywords
GOST, differential cryptanalysis, input difference, output difference, characteristic, symmetric key.

1. INTRODUCTION
The method of differential cryptanalysis, first proposed by E. Biham and A. Shamir for the analysis of DES algorithm [1, 2], is based on tracking of changes in the difference between the two messages as they pass through the rounds of encryption. After the publication of [1, 2], the majority of existing encryption algorithms were analyzed using this method. Studies have shown that the method of differential cryptanalysis is universal, i.e. it can be applied to the analysis of most known symmetric cryptosystems. That is why newly created encryption algorithms are primarily tested for the resistance to this type of analysis.

Our research is aimed to evaluating the resistance to the method of differential cryptanalysis of GOST algorithm defined as the state standard in Russian Federation. In public sources there is relatively little information about possible vulnerabilities of this cipher. One of the most important works is [3], in which the authors propose a variant of GOST algorithm analysis using differential cryptanalysis in related keys (Related-Key Attack) provided use of weak S-boxes. In this article, we propose to consider the possibility of attack against GOST encryption algorithm using the classical method of differential cryptanalysis and to determine the conditions which make implementation of this attack possible.

The characteristic feature of GOST algorithm is the use of customized S-boxes. It is assumed that any filling of S-boxes of thirty-two rounds of encryption is sufficient to resist effective analysis methods such as linear and differential cryptanalysis. Our research shows that there are weak S-boxes which may lead to the successful differential cryptanalysis attack if used in GOST. It had been considered earlier that, if S-boxes are kept in secret, they can be treated as an additional subkey [6]. However, a method has been proposed in [5], which makes it possible to easily restore the values of S-boxes used for data encryption.

The use of customized S-boxes in GOST algorithm leads to the fact that every time (when S-boxes are changed) analysis has to be started from the beginning. Unlike algorithms with fixed S-boxes, in which good features can be identified once and then used for encrypted data analysis. In this paper, we show that the application of our suggested approaches to the analysis, as well as using the characteristic search algorithm proposed in [4], the first stage of analysis, which consists in finding good properties, is not a complicated problem. A good characteristic can be found in a few seconds or even less.

Before proceeding to the solution, we introduce some basic definitions that are used for presentation. By *difference* we mean a result of the bitwise addition modulo 2 (XOR operation) of two separately encrypted texts with the same symmetric key, which are in the same position of the same encryption algorithm. *Input difference* means the difference of input values before cryptographic transformation; *output difference* is the difference

SIN'10, Sept. 7–11, 2010, Taganrog, Rostov-on-Don, Russian Federation.

obtained at the output of the transformation. By *characteristic* we mean a combination of input and output values for n rounds of the encryption algorithm. *Correct pair of texts* means a pair of plaintext-ciphertext combinations, for which the difference of plaintext is equal to the input difference of the characteristic and the difference of ciphertext is equal to the output difference of the characteristics.

This paper is organized as follows. Section 2 describes GOST algorithm. Section 3 is devoted to the influence of major cryptographic primitives of GOST algorithm to transformation of the difference between processed texts. Section 4 contains the definition of weak S-boxes for GOST with respect to differential cryptanalysis method and an algorithm for searching similar blocks. Section 5 shows how one can obtain a good characteristic for 32 rounds of GOST using weak S-boxes. Section 6 explains how to use correct pairs of texts to extract information about the symmetric key. Section 7 presents the results of an attack implemented using the proposed approach against a simplified version of GOST. Conclusions on progress and highlights of further research directions are made.

2. GOST OVERVIEW

GOST 28147-89 encryption algorithm is the state standard of the Russian Federation and its use is mandatory for Russian state organizations. GOST algorithm is a symmetric block cipher, which conforms to Feistel scheme. 64-bit blocks of data are submitted to the input and converted into 64-bit blocks of encrypted data by 256-bit key. In each round the right side of plain text messages is processed by function F, which converts data with three cryptographic operations: adding data and subkey modulo 2^{32}, substitution of data using S-boxes, and left cyclic shift by 11 positions. Output of F-function is added modulo 2 to the left part of the plaintext, then right and left sides are swapped for next round. The algorithm has 32 rounds. In the last round of encryption right and left parts are not swapped. The overall dataflow diagram of GOST is shown in Figure 1.

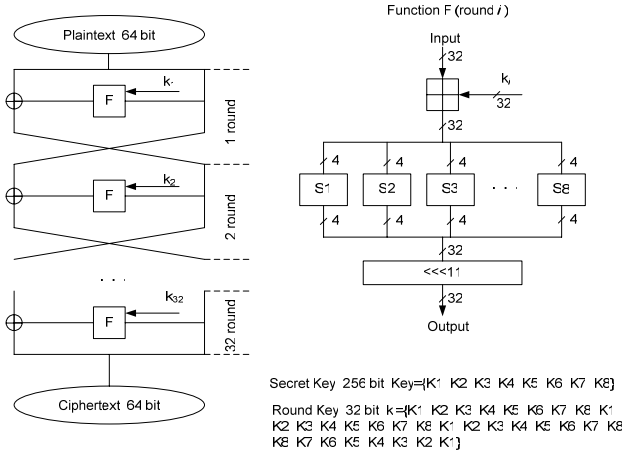

Figure 1. GOST dataflow.

GOST uses 8 S-boxes, which convert 4-bit input to 4-bit output. Unlike most encryption algorithms, GOST has no predefined S-boxes and any values can be used for them.

Secret key contains 256 bits and is represented as a sequence of eight 32-bit words: K1, K2, K3, K4, K5, K6, K7 and K8. In each round of encryption one of these 32-bit words is used as a round subkey. When round subkey is calculated, the following principle is used: from round 1 to round 24 the order is straight, (K1, K2, K3, K4, K5, K6, K7, K8, K1, K2, etc); from round 25 to round 32 reversed order is used (K8, K7, K6, K5, K4, K3, K2, K1).

Thus, it appears that the same subkey K1 is used at both the first and the last rounds.

3. DIFFERENTIAL PROPERTIES OF GOST

3.1 Addition modulo 2^{32}

The method of differential cryptanalysis is based on tracking changes in dissimilarity between two texts. To determine the dissimilarity using the addition operation modulo 2 as a result of addition gives non-zero bits in those positions in which the two original texts have different values of bits. That is why in DES algorithm, the key value does not influence the change in the difference between the texts, because the same input values will give 0 as the result of addition modulo two. Unlike DES, GOST uses addition modulo 2^{32}. Therefore it is necessary to investigate the influence of such additions to transformation of differences.

We carried out an analysis of 2, 3, 4 and 5-bit numbers, which were added with each other respectively, modulo 2^2, 2^3, 2^4 and 2^5. For each value of the input difference, not only possible variants of its formation were considered, but also different versions of secret key values. Using an inductive analysis method we formulated rules for determining the probability that the difference will remain unchanged when arguments are added modulo 2^n:

1. Any value of the input difference may remain unchanged. The probability of this mapping is defined as follows:

$$p = \frac{1}{2^k}, \text{ if } \Delta_{in} < 2^{n-1} \quad (1),$$

$$p = \frac{1}{2^{k-1}}, \text{ if } \Delta_{in} \geq 2^{n-1} \quad (2),$$

where k is the number of nonzero digits in the input difference Δ_{in}.

2. For the input difference $\Delta_{in} = 0$, the output difference Δ_{out} reaches 0 with the probability of p = 1.

3. For the input difference $\Delta_{in} = 2^{n-1}$ the output difference Δ_{out} reaches 2^{n-1} with the probability of p = 1.

More detailed description can be found in [4].

3.2 Substitution with S-boxes

Dissimilarity of different pairs of texts enciphered by cryptographic operations leads to dissimilarity of obtained ciphertext with a certain probability. These probabilities can be found for S-boxes by building analysis tables. Tables are built according to the following principle: the columns contain all possible combinations of input difference of ΔA for a given S-box; rows correspond to all possible combinations of output difference of ΔC for the same S-box, and cells contain numbers of matching ΔC values with a predefined value of ΔA (or the ratio between the obtained number of matches and the total possible number of matches, which in turn can be defined as 2^n, where n is the input capacity of S-boxes.) Pairs of differences ΔA and ΔC

characterized by the highest probability (or the probability close to maximum) can be used for analysis in order to find the secret key.

3.3 Cyclic shift

Shift operation in GOST is one of three basic operations that comprise the round transformation function. Differential cryptanalysis usually implies the difference between two texts. Suppose we have two texts: A and B. Their difference is defined as A ⊕ B. If we shift each of the values of A and B cyclically to the left by s bits, then we obtain the difference (A <<s) ⊕ (B <<s), characterized by the following property:

$$(A \ll s) \oplus (B \ll s) = (A \oplus B) \ll s \qquad (3)$$

If we consider this operation in relation to GOST, we get:

$$(A \ll 11) \oplus (B \ll 11) = (A \oplus B) \ll 11,$$

i.e., in order to obtain the correct difference at the output of a cyclic shift operation, one has to shift the input difference cyclically left by 11 digits.

4. WEAK S-BOXES FOR GOST: SEARCH AND ANALYSIS

GOST relies on eight S-boxes, which are not fixed. That is, it is implied that randomly generated S-boxes can be used. Cryptographic strength should be provided by a sufficient number of encryption rounds (i.e., 32). It had been thought that if S-boxes are kept in secret, one can consider them as additional parts of the key. However, it has been demonstrated in [5] that particular values of S-boxes used for encryption can be reconstructed quite easily.

In this regard, it is reasonable to consider weak S-boxes for GOST and estimate the complexity of differential cryptanalysis attack based on their use.

The first problem here is how to identify weak blocks and how to obtain them. To answer this question we have to recall that the probability of the fact that the difference remains unchanged after addition modulo 2^{32} is directly dependent on the number of nonzero positions in it. Therefore, we have assumed that the weak ones are the boxes for which the input difference ΔA contains only one high bit (i.e., only 4 variants of such difference exist, namely ΔA = 1, ΔA = 2, ΔA = 4, and ΔA = 8) will be replaced by the output difference ΔC, which also contain only one high bit (i.e., ΔC = 1, ΔC = 2, ΔC = 4, or ΔC = 8).

If we assume that the difference ΔA obtained from adding two input messages X and X1, we can compare all the options for the differences ΔA as shown in Table 1. Parentheses in Table 1 indicate corresponding input values in decimal form.

After that, we compared the value of each entry i with the corresponding output a_i and presented the results in a graph of connections as shown in Figure 2. Arrows in Figure 2 imply connections for inputs i, which are used to obtain values ΔA = 1, ΔA = 2, ΔA = 4, and ΔA = 8. Roman numerals near arrows indicate connection indexes (i.e., connection I is for ΔA = 1, connection II is for ΔA = 2, connection III is for ΔA = 4, and connection IV is for ΔA = 8). For example, the value ΔA = 2 (connection II) can be obtained either if X = 0 and X1 = 2 or if X = 1 and X1 = 3, etc.

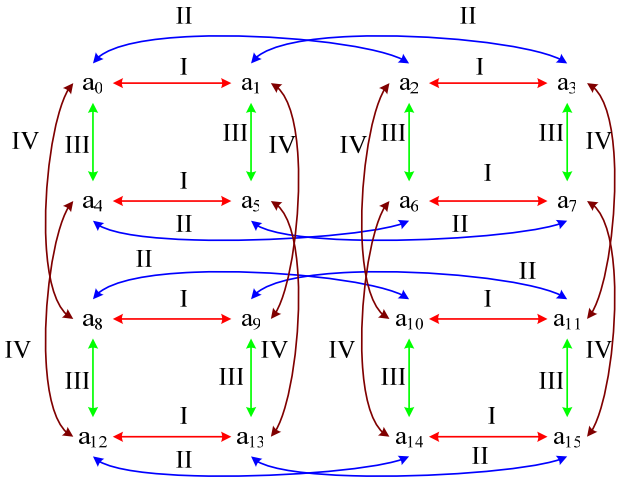

Figure 2. Connection graph for possible outputs of S-box

Only one of the four output values of ΔC (ΔC = 1, ΔC = 2, ΔC = 4, ΔC = 8) can match each input value of ΔA. Thus, 24 combinations of quadruples of output differences can match the four input differences ΔA = 1, ΔA = 2, ΔA = 4, and ΔA = 8 (see Table. 2).

Denote the output difference value for ΔA = j as ΔC_j. Hence the following fifteen formulas can be obtained according to Figure 2:

$a_1 = a_0 \oplus \Delta C_1$; $a_{10} = a_2 \oplus \Delta C_4$;

$a_2 = a_0 \oplus \Delta C_2$; $a_7 = a_3 \oplus \Delta C_3$;

$a_4 = a_0 \oplus \Delta C_3$; $a_{11} = a_3 \oplus \Delta C_4$;

$a_8 = a_0 \oplus \Delta C_4$; $a_{12} = a_4 \oplus \Delta C_4$;

$a_3 = a_1 \oplus \Delta C_2$; $a_{14} = a_6 \oplus \Delta C_4$;

$a_5 = a_1 \oplus \Delta C_3$; $a_{15} = a_7 \oplus \Delta C_4$;

$a_9 = a_1 \oplus \Delta C_4$; $a_{13} = a_{15} \oplus \Delta C_2$;

$a_6 = a_2 \oplus \Delta C_3$;

In fact, 28 formulas can be constructed altogether, however the excessive ones will duplicate the former 15.

Sixteen different S-boxes can be generated by varying the value of a_0 from 0 to 15 for each possible ΔC combination. So we can generate 384 (16 by 24) different weak S-boxes.

As an example, we used the combination No. 11 from Table 2. By applying the formulas obtained above we get the following:

$a_1 = a_0 \oplus \Delta C_1; \rightarrow a_1 = 1 \oplus 2; \rightarrow a_1 = 3.$

$a_2 = a_0 \oplus \Delta C_2; \rightarrow a_2 = 1 \oplus 8; \rightarrow a_2 = 9.$

$a_4 = a_0 \oplus \Delta C_3; \rightarrow a_4 = 1 \oplus 1; \rightarrow a_4 = 0.$

$a_8 = a_0 \oplus \Delta C_4; \rightarrow a_8 = 1 \oplus 4; \rightarrow a_8 = 5.$

$a_3 = a_1 \oplus \Delta C_2; \rightarrow a_3 = 3 \oplus 8; \rightarrow a_3 = 11.$

$a_5 = a_1 \oplus \Delta C_3; \rightarrow a_5 = 3 \oplus 1; \rightarrow a_5 = 2.$

$a_9 = a_1 \oplus \Delta C_4; \rightarrow a_9 = 3 \oplus 4; \rightarrow a_9 = 7.$

$a_6 = a_2 \oplus \Delta C_3; \rightarrow a_6 = 9 \oplus 1; \rightarrow a_6 = 8.$

Table 1. Variations of input values for ΔA differences

Input X	Input X1 ΔA=0001 (1)	Input X1 ΔA=0010 (2)	Input X1 ΔA=0100 (4)	Input X1 ΔA=1000 (8)
0000 (0)	0001 (1)	0010 (2)	0100 (4)	1000 (8)
0001 (1)	0000 (0)	0011 (3)	0101 (5)	1001 (9)
0010 (2)	0011 (3)	0000 (0)	0110 (6)	1010 (10)
0011 (3)	0010 (2)	0001 (1)	0111 (7)	1011 (11)
0100 (4)	0101 (5)	0110 (6)	0000 (0)	1100 (12)
0101 (5)	0100 (4)	0111 (7)	0001 (1)	1101 (13)
0110 (6)	0111 (7)	0100 (4)	0010 (2)	1110 (14)
0111 (7)	0110 (6)	0101 (5)	0011 (3)	1111 (15)
1000 (8)	1001 (9)	1010 (10)	1100 (12)	0000 (0)
1001 (9)	1000 (8)	1011 (11)	1101 (13)	0001 (1)
1010 (10)	1011 (11)	1000 (8)	1110 (14)	0010 (2)
1011 (11)	1010 (10)	1001 (9)	1111 (15)	0011 (3)
1100 (12)	1101 (13)	1110 (14)	1000 (8)	0100 (4)
1101 (13)	1100 (12)	1111 (15)	1001 (9)	0101 (5)
1110 (14)	1111 (15)	1100 (12)	1010 (10)	0110 (6)
1111 (15)	1110 (14)	1101 (13)	1011 (11)	0111 (7)

Table 2. Variants of conformity of ΔC differences

	No 1 ΔC	No 2 ΔC	No 3 ΔC	No 4 ΔC	No 5 ΔC	No 6 ΔC	No 7 ΔC	No 8 ΔC	No 9 ΔC	No 10 ΔC	No 11 ΔC	No 12 ΔC
ΔA=1	1	1	1	1	1	1	2	2	2	2	2	2
ΔA=2	2	2	4	4	8	8	1	1	4	4	8	8
ΔA=4	4	8	2	8	2	4	4	8	1	8	1	4
ΔA=8	8	4	8	2	4	2	8	4	8	1	4	1
	No 13 ΔC	No 14 ΔC	No 15 ΔC	No 16 ΔC	No 17 ΔC	No 18 ΔC	No 19 ΔC	No 20 ΔC	No 21 ΔC	No 22 ΔC	No 23 ΔC	No 24 ΔC
ΔA=1	4	4	4	4	4	4	8	8	8	8	8	8
ΔA=2	1	1	2	2	8	8	1	1	2	2	4	4
ΔA=4	2	8	8	1	1	2	2	4	1	4	1	2
ΔA=8	8	2	1	8	2	1	4	2	4	1	2	1

$a_{10} = a_2 \oplus \Delta C_4; \rightarrow a_{10} = 9 \oplus 4; \rightarrow a_{10} = 13.$

$a_7 = a_3 \oplus \Delta C_3; \rightarrow a_7 = 11 \oplus 1; \rightarrow a_7 = 10.$

$a_{11} = a_3 \oplus \Delta C_4; \rightarrow a_{11} = 11 \oplus 4; \rightarrow a_{11} = 15.$

$a_{12} = a_4 \oplus \Delta C_4; \rightarrow a_{12} = 0 \oplus 4; \rightarrow a_{12} = 4.$

$a_{14} = a_6 \oplus \Delta C_4; \rightarrow a_{14} = 8 \oplus 4; \rightarrow a_{14} = 12.$

$a_{15} = a_7 \oplus \Delta C_4; \rightarrow a_{15} = 10 \oplus 4; \rightarrow a_{15} = 14.$

$a_{13} = a_{15} \oplus \Delta C_2; \rightarrow a_{13} = 14 \oplus 8; \rightarrow a_{13} = 6.$

The constructed S-box is presented in Table 3 and its probabalistic analysis table is presented in Table 4.

It may seem at a glance that the constructed S-box cannot be used for attack because the probabilities in Table 4 are either 0 or 1, but this is not so. In fact S-boxes are not the only component of GOST that affect variation of characteristic's probability. Integer addition modulo 2^{32} affects difference transformation as well. Under these conditions, the probability that the difference remains unchanged depends on the quantity of non-zero digits of the difference, which is directed to transformation input. It is possible to find characteristics for algorithm analysis using S-boxes such as the one defined in Table 3. In this case, the probability that the difference remains unchanged will be 1 while the common characteristic probability will depend on addition modulo 2^{32}. In the next section we demonstrate how the S-box defined in Table 3 can be used for analysis.

5. FINDING CHARACTERISTICS OF GOST

We have considered the differential properties of the basic cryptographic transformations used in GOST namely operation of integer addition modulo 2^{32}, bit substitution using S-boxes and cyclic shift by 11 digits. Besides that we have defined the features of weak S-boxes, which, if used in GOST, increase chances of success of the differential cryptanalysis attack. In this section we survey the ways of using properties of weak S-boxes for finding good characteristics, which provide data about the secret key in reasonable time.

In this paper we always use the same S-box presented in Table 3 for demonstration.

Before finding good characteristics, we have to find out which characteristics cannot be used for analysis. According to the rules of difference transformation caused by addition modulo 2^n, if $\Delta_{in} = 2n - 1$, then $\Delta_{out} = 2n - 1$ with probability $p = 1$. Thus, if the right part of an output difference of the characteristics will be equal to 80000000_x, such characteristic cannot be used for finding the secret key, because the probability that the characteristic passes the last round of encryption is also $p = 1$. All the more so, in accordance with a round subkey searching algorithm, which will be discussed in Section 6, we estimate the secret key using tetrads, which correspond to the data passing through S-boxes. Therefore, the tetrads of the right part of characteristics output difference, which contain the values of 0_x or 8_x cannot be used for finding secret key either.

It turns out that it is necessary to find a characteristic, which contains values other than 0_x or 8_x in the right part of the output difference in each tetrad. According to a property integer addition modulo 2^{32}, finding unique characteristic with non-zero values in each tetrad leads to a little probability, because a difference containing at least 8 high bits passes through integer addition each round. It is much easier to receive some characteristics with sufficiently high probabilities, each of which involves one or more tetrads.

A universal algorithm for finding good characteristics for GOST has been presented in [4]. Input values of the difference are the input of the algorithm. All possible variations of converting the difference are considered (taking features of S-box analysis tables into account) and only those output difference are selected, whose probability is either maximum or not lower than a predefined threshold (in accordance with initial algorithm settings). As soon as for the chosen S-box sample (Table 3), analysis table (Table 4) contains only maximum probabilities ($p = 1$), round transformation option for characteristics input difference will be defined uniquely. In this regard, searching output difference with a predefined input difference takes less than a second. We used this algorithm to test all possible options of input differences in tetrads which may take one of the five values namely 0_x, 1_x, 2_x, 4_x, and 8_x. As soon as the input of GOST contains only 64 bits (16 tetrads), we tried $5^{16} \approx 2^{37.15}$ variants. Analysis of 32 rounds of GOST has shown that there is a large amount of input/output difference pairs for which the characteristic has quite high probability (from 2^{-25} to 2^{-33}) and can be used for analysis. As we had expected, the input differences, for which several tetrads (at least three) contained non-zero tetrad, were not characterized by good probabilities. So there is a sound reason to conclude that there is no need to do exhaustive search of input differences. It suffices to consider several options, where different tetrads containing different values and use them later for analysis. Such selection combined with the algorithm described in [4] can take from several minutes to half an hour.

Consider one of the found characteristics in detail. Its analysis is presented in Table 5. In Table 5 the input value ΔX_i indicates the difference between i-th round, and ΔY_i is the output difference of i-th round. Thus, the input of the encryption algorithm received the input difference $\Delta X1 = 80400000\ 00000000_x$. In accordance with $\Delta X1$, the difference 00000000_x, which remains unchanged with probability $p1 = 1$, goes to the input of function F of the first round of encryption. After that, the output of F function is added to the left part of $\Delta X1$ (80400000_x) modulo two, and the parts are swapped. As the result, the input of the second round of encryption is the difference $\Delta X2 = 00000000\ 80400000_x$. Non-zero difference 80400000_x serves as an argument of F function. The first transformation performed by F function is integer addition modulo 2^{32}. Since $80400000_x > 231$, using formula (2) we can determine the probability with which the value 80400000_x of the difference remains unchanged $p2 = \frac{1}{2}$. The cumulative probability for the two-round characteristic will be $p = p1*p2 = \frac{1}{2}$. The next step is transformation with S-boxes. According to Table 4, the value of 8_x is always transformed to 4_x; and the value of 4_x, to 1_x. Thus, after substitution by means of S-boxes, the difference is converted to 40100000_x.

Table 3. Constructed S-box

Input	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Output	1	3	9	11	0	2	8	10	5	7	13	15	4	6	12	14

Table 4. Probabilistic table for analysis of constructed S-boxes

	ΔC=0	ΔC=1	ΔC=2	ΔC=3	ΔC=4	ΔC=5	ΔC=6	ΔC=7	ΔC=8	ΔC=9	ΔC=10	ΔC=11	ΔC=12	ΔC=13	ΔC=14	ΔC=15
ΔA=0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ΔA=1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
ΔA=2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
ΔA=3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
ΔA=4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ΔA=5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
ΔA=6	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
ΔA=7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
ΔA=8	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
ΔA=9	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
ΔA=10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
ΔA=11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
ΔA=12	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
ΔA=13	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
ΔA=14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
ΔA=15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

The last transformation is the cyclic left shift by 11 positions: 40100000_x <<11 = 80000200_x. We have determined that the output of F on the second round is 80000200_x with the probability $\frac{1}{2}$. After addition and swapping we find that the input of the third round is ΔX3 = 80400000 80000200_x. All further transformations are performed in a similar fashion. The values of input and output differences for each round are shown in Table 5. It should be noted that difference starts repeating cyclically after five rounds (the repeating unit is highlighted in Table 5). Besides that, differences are transformed in each round by integer addition modulo 2^{32} with a probability of $\frac{1}{2}$ at least.

On the contrary, in round 7 of 32 (rounds 1, 6, 11, 16, 21, 26, 31) the difference 00000000_x comes to the input of F and remains unchanged with probability p = 1. The final probability of getting the characteristic is $\frac{1}{2^{25}}$.

6. TIPS FOR FINDING THE SECRET KEY

We discussed how one can use weak S-boxes in GOST for finding characteristics holding with sufficiently high probability. This can be used for analysis of GOST. The question how this knowledge can be used for finding secret keys has to be answered.

As we have already mentioned, it is quite easy to find round characteristics whose probability belong to the interval from 2^{-25} to 2^{-33}. If possible the total number of characteristics should be such that no tetrads of output difference exist whose values are either 0_x or 8_x. These probabilities let us expect that correct pairs of texts, which are suitable for analysis, are found easily. According to the birthday paradox, in order to find a correct pair of texts corresponding to a characteristic with probability $\frac{1}{2^{25}}$, one has to analyze $2^{33.5}$ text pairs.

We are considering the first and the last round of encryption for each correct text pair. In accordance with the original GOST description, the same round subkey K1is used at the first and the last round. Thus, the analysis of the first and the last round will allow us to make an assumption about the importance of first round subkey. Analysis of the first round is impossible for the characteristic shown in Table 5, because its input receives the difference 00000000_x. However, is does not necessarily hold for other characteristics, so the first round should also be taken into account.

The fact that the correct pair of texts corresponding to a given characteristic is found allows to assume that the difference was transformed at encryption rounds exactly as it was defined by the constructed characteristic. Thus, when the right pair of texts is known, right parts of the original message XR and XR1 that came to F of the first round, are known either. We also know that after integer addition modulo 2^{32} the difference in these texts remains unchanged. For the first round subkey K1, we consider eight tetrads, namely k1, k2, k3, k4, k5, k6, k7, and k8, in accordance with the number of used S-boxes. For each tetrad we try 16 possible variants of key fragment (from 0000 to 1111). The options that preserve the difference in tetrads and XR1 XR after integer addition will be treated as options for round subkey fragments. Furthermore, it should be taken into account that whenever plaintext tetrads (XR or XR1) are added to round subkey tetrads, carries from lower bits can occur. Therefore, for all tetrads except the lowest one, we should consider the option when an extra 1 is added to the lowest bit. Analyzing correct text pairs we can see that some values of each round subkey tetrads occur more often than others and this fact lets us to make assumptions about possible values of the round subkey.

A hint by E. Biham and A. Shamir [1] for analysis of DES algorithm can help us to improve results. The idea is to search only the left half of the input characteristic when searching correct text pairs for the full 16-bit DES. The right half can take any value. In this case, the probability of the situation when the difference remains unchanged at the last round is not taken into account for finding out the overall characteristic probability. The fact that lower halves of correct text pairs coincide with lower halves of the characteristics makes it possible to assume that the difference between texts is transformed by the rounds as it was defined when the characteristic was constructed. The difference was transformed to one of the possible ways only in the last round.

Therefore, if we, when finding correct text pairs, assume that the criterion for their selection is just the right half of the output difference, it makes it possible to analyze the last round of encryption in more detail. In this case, we have to get the difference from the output of integer addition modulo 2^{32} by function F of the last round of encryption. To do so, the difference formed by the left halves of the encrypted messages of correct text pairs has to be added modulo two to the left part of the estimated value of the difference on the input of the last round. Thus, we obtain the value of the difference that occurred at the output of function F at the last round. Then the final value of the difference has to be shifted right by 11 digits and in accordance with the table of analysis for S-boxes and the value of the difference, which was received at the input of S-boxes, has to be found. This value is the expected difference of the output of integer addition modulo 2^{32}. Further key testing should be performed as described for the first round of encryption.

As the result the correct text pair analysis, a number of possible values (up to several thousands) of round subkey K1will be formed. After that, it is necessary to analyze the same right pairs of texts, though not with subkey fragments, but with selected values of the all possible 32-bit round subkey values. After the testing, only a few keys (less than ten) will remain and one of them will be the real round subkey.

7. EXPERIMENTAL RESULTS

In order to verify the effectiveness of the method, we simulated an attack on GOST with the number of rounds reduced to 12. In this case we used 8 round subkeys so that in the first and last round the same round subkey is used. We used a weak S-box defined in Table 3. Attack on full GOST will look the same way except that it will demand much more text pairs for finding a correct pair.

At first we have found 10 characteristics for 12 rounds of GOST presented in Table 6. The table shows that right sides of output differences ΔY have from one to three non-zero tetrad while at least one tetrad is either 1_x, 2_x, or 4_x.

Table 5. Difference transformations on 32 rounds of GOST

Round	ΔXi	ΔYi	p	Round	ΔXi	ΔYi	p
1	80400000 00000000	00000000 80400000	1	17	00000000 80400000	80400000 80000200	$\frac{1}{2^{13}}$
2	00000000 80400000	80400000 80000200	$\frac{1}{2}$	18	80400000 80000200	80000200 80000200	$\frac{1}{2^{14}}$
3	80400000 80000200	80000200 80000200	$\frac{1}{2^2}$	19	80000200 80000200	80000200 80400000	$\frac{1}{2^{15}}$
4	80000200 80000200	80000200 80400000	$\frac{1}{2^3}$	20	80000200 80400000	80400000 00000000	$\frac{1}{2^{16}}$
5	80000200 80400000	80400000 00000000	$\frac{1}{2^4}$	21	80400000 00000000	00000000 80400000	$\frac{1}{2^{16}}$
6	80400000 00000000	00000000 80400000	$\frac{1}{2^4}$	22	00000000 80400000	80400000 80000200	$\frac{1}{2^{17}}$
7	00000000 80400000	80400000 80000200	$\frac{1}{2^5}$	23	80400000 80000200	80000200 80000200	$\frac{1}{2^{18}}$
8	80400000 80000200	80000200 80000200	$\frac{1}{2^6}$	24	80000200 80000200	80000200 80400000	$\frac{1}{2^{19}}$
9	80000200 80000200	80000200 80400000	$\frac{1}{2^7}$	25	80000200 80400000	80400000 00000000	$\frac{1}{2^{20}}$
10	80000200 80400000	80400000 00000000	$\frac{1}{2^8}$	26	80400000 00000000	00000000 80400000	$\frac{1}{2^{20}}$
11	80400000 00000000	00000000 80400000	$\frac{1}{2^8}$	27	00000000 80400000	80400000 80000200	$\frac{1}{2^{21}}$
12	00000000 80400000	80400000 80000200	$\frac{1}{2^9}$	28	80400000 80000200	80000200 80000200	$\frac{1}{2^{22}}$
13	80400000 80000200	80000200 80000200	$\frac{1}{2^{10}}$	28	80000200 80000200	80000200 80400000	$\frac{1}{2^{23}}$
14	80000200 80000200	80000200 80400000	$\frac{1}{2^{11}}$	30	80000200 80400000	80400000 00000000	$\frac{1}{2^{24}}$
15	80000200 80400000	80400000 00000000	$\frac{1}{2^{12}}$	31	80400000 00000000	00000000 80400000	$\frac{1}{2^{24}}$
16	80400000 00000000	00000000 80400000	$\frac{1}{2^{12}}$	32	00000000 80400000	80000200 80400000	$\frac{1}{2^{25}}$

Table 6. Characteristics for 12-round GOST

№	ΔX	ΔY	ΔYL^{-1}	p	№	ΔX	ΔY	ΔYL^{-1}	p
1	00001000 00000001	00010000 10000000	00010100	$\frac{1}{2^{18}}$	6	00008000 00000040	00000040 02008040	02008000	$\frac{1}{2^{18}}$
2	00000800 00000004	00000004 00200804	00200800	$\frac{1}{2^{18}}$	7	00020000 00000080	00000080 40020080	40020000	$\frac{1}{2^{18}}$
3	00020000 00000008	00000008 04002008	04002000	$\frac{1}{2^{18}}$	8	00100000 00000100	01000000 00000010	01010000	$\frac{1}{2^{18}}$
4	00010000 00000010	00100000 00000001	00101000	$\frac{1}{2^{18}}$	9	00400000 00000200	00000200 80400200	80400000	$\frac{1}{2^{13}}$
5	00040000 00000020	00000020 08040020	08040000	$\frac{1}{2^{18}}$	10	80000000 00400000	00400000 80400200	80000200	$\frac{1}{2^{11}}$

For each characteristic in Table 6 we tested 100 000 pairs of texts in order to find a correct text pair. We used the technique described in Section 6 for this purpose. As the result of primary selection of tetrads of the secret key, 16384 possible keys have been selected from 2^{32} total values. Subsequent testing of the full round subkey was selecting 5 possible values in average (depending on the number of the found correct text pairs, this value fluctuated from 2 to 10). Full analysis of 12 rounds of GOST using ten round characteristics presented in Table 6 took from 1 to 2 minutes in average (experiments were carried out with Intel Celeron M CPU 530 1.73 GHz, RAM 1007Mb). We have carried out around 1000 experiments using different values of round subkeys and the result was always positive.

8. CONCLUSION

In this paper we have envisaged the possibility of an attack on GOST using the method of differential cryptanalysis provided that the algorithm uses weak S-boxes. We have shown which blocks can be considered weak and offered a way to find such blocks quickly. As the result of using weak S-boxes for full 32-round GOST, finding characteristics with probabilities in the range from 2^{-25} to 2^{-33} becomes possible that, in turn, allows expecting higher chances of a successful attack. In addition, we have proposed a method of analyzing correct text pairs for finding the secret key.

In order to demonstrate the possibility of analysis, we carried out a simulation of an attack on 12-round GOST with weak blocks. The low time of analysis allows us expect faster success of analyzing the full 32-round algorithm.

This paper reviews generation of characteristics for GOST analysis using the same S-box. The foreseen direction of further research is the study of the influence of weak S-boxes combinations on characteristic probability, and identifying those ones that may lead to cipher breaking.

9. ACKNOWLEDGMENTS

The authors would like to thank Maxim Anikeev for valuable comments.

10. REFERENCES

[1] Biham, E., Shamir, A. 1998. Differential Cryptanalysis of the Full 16-round DES. In: *Crypto'92,* Springer-Verlag, 487.

[2] Biham, E., Shamir A. 1998. Differential Cryptanalysis of DES-like Cryptosystems, Extended Abstract. In: *Crypto'90,* Springer-Verlag, 2.

[3] Kelsey, J., Schneier, B., Wagner, D. *Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SARER, and Triple-DES*, http://www.schneier.com.

[4] Babenko, L.K., Ischukova, E.A. 2007. Application of recursive search algorithms in B-trees for differential cryptanalysis of GOST 28147-89 cipher. In: *Proc. IX Intl. Scientific and Practical Conf. "Information Security",* vol. 2, Taganrog, TSURE, 92-97, (in Russian).

[5] Saarien, M.-J. *A Chosen Key Attack Against the Secret S-boxes of GOST,* Helsinki University of Technology, Finland, http://www.m.-js.com.

[6] Panasenko, S.P. 2009 *Encryption. Special reference*, Saint-Petersburg, BHV-Peterburg, 576, (in Russian).