The Encryption Algorithms GOST-IDEA16-2 and GOST-RFWKIDEA16-2

By Gulom Tuychiev
National University of Uzbekistan, Uzbekistan

Abstract- In the paper created a block encryption algorithms GOST28147-89-IDEA16-2 and GOST28147-89-RFWKIDEA16-2 based on networks IDEA16-2 and RFWKIDEA16-2, with the use the round function of the encryption algorithm GOST 28147-89. The block length of created block encryption algorithm is 128 bits, the number of rounds is 8, 12 and 16.

Keywords: GOST 28147-89, Lai-Massey scheme, round function, round keys, output transformation.

GJCST-E Classification : G.4, E.3
The Encryption Algorithms GOST-IDEA16-2 and GOST-RFWKIDEA16-2

Gulom Tuychiev

Abstract- In this paper, applying the round function of the encryption algorithm GOST 28147-89 as round functions of the networks IDEA16-2 [14] and RFWKIDEA16-2 [15], developed new encryption algorithms GOST28147-89-IDEA16-2 and GOST28147-89-RFWKIDEA16-2.

In the encryption algorithms GOST28147-89-IDEA16-2 and GOST28147-89-RFWKIDEA16-2 block length is 256 bits, the key length is changed from 256 bits to 1024 bits in increments of 128 bits and a number of rounds equal to 8, 12, 16, allowing the user depending on the degree of secrecy of information and speed of encryption to choose the number of rounds and key length. Below is the structure of the proposed encryption algorithm.

II. THE STRUCTURE OF THE ENCRYPTION ALGORITHM GOST28147-89-IDEA16-2

In the encryption algorithm GOST28147-89-IDEA16-2 the length of subblocks $X^0$, $X^1$, $X^2$, ..., $X^{15}$, length of round keys $K_{2(4i-1)}$, $K_{2(4i-1)+1}$, $K_{2(4i-1)+2}$, ..., $K_{2(4i-1)+15}$, $i=1,...,n+1$, $K_{24(n-1)+16}$, $K_{24(n-1)+17}$, $K_{24(n-1)+18}$, ..., $K_{24(n-1)+23}$ are equal to 8-bits. The length of the input and output blocks of round functions is 32 bits. This encryption algorithm round function GOST 28147-89 is applied twice and in each round function employed eight S-boxes, i.e., the total number of S-boxes is 16. The structure of the encryption algorithm GOST28147-89-PES16-2 is shown in Figure 1 and the S-boxes shown in Table 1.

Consider the round function block encryption algorithm GOST28147-89-IDEA16-2. First the 8-bit subblocks $T^0$, $T^1$, ..., $T^7$ combined from 32-bit subblocks, i.e., $T^0 = T^0 || T^1 || T^2 || T^3 || T^4 || T^5 || T^6 || T^7$. Subblocks $S^0$, $S^1$, $T_i$ are summed round keys $K_{24(i-1)+16}$, $K_{24(i-1)+17}$, $K_{24(i-1)+18}$, ..., $K_{24(i-1)+23}$, i.e., $S^0 = T^0 + (K_{24(i-1)+16} || K_{24(i-1)+17} || K_{24(i-1)+18} || K_{24(i-1)+19})$, $S^1 = T^1 + (K_{24(i-1)+20} || K_{24(i-1)+21} || K_{24(i-1)+22} || K_{24(i-1)+23})$.

I. INTRODUCTION

The encryption algorithm GOST 28147-89 is a standard encryption algorithm of the Russian Federation. It is based on a Feistel network. This encryption algorithm is suitable for hardware and software implementation, meets the necessary cryptographic requirements for resistance and, therefore, does not impose restrictions on the degree of secrecy of the information being protected. The algorithm implements the encryption of 64-bit blocks of data using the 256 bit key. In round functions used eight S-box of size 4x4 and operation of the cyclic shift by 11 bits. To date GOST 28147-89 is resistant to cryptographic attacks.

On the basis of encryption algorithm IDEA and scheme Lai-Massey developed the networks IDEA16-2 and RFWKIDEA16-2, consisting from two round function. In the networks IDEA16-2 and RFWKIDEA16-2, similarly as in the Feistel network, when it encryption and decryption using the same algorithm. In the networks used two round function having four input and output blocks and as the round function can use any transformation.

As the round function networks IDEA4-2 [1], RFWKIDEA4-2 [5], PES4-2 [6], RFWKLPES4-2 [7], PES8-4 [2], RFWKLPES8-4 [8] using the round function of the encryption algorithm GOST 28147-89 [4] created the encryption algorithm GOST28147-89-IDEA4-2 [9], GOST28147-89-RFWKIDEA4-2 [10], GOST28147-89-PES4-2 [11], GOST28147-89-RFWKLPES4-2 [12], GOST28147-89-PES8-4 [13] and GOST28147-89-RFWKLPES8-4 [13].

Author: Candidate technical science (Ph.d), the teacher of National University of Uzbekistan, Uzbekistan, Tashkent. e-mail: blasterjon@gmail.com
The Encryption Algorithms GOST-IDEA16-2 and GOST-RFWKIDEA16-2

Figure 1: The scheme n-rounded encryption algorithm GOST28147-89-IDEA16-2

Table 1: The S-boxes of encryption algorithm GOST28147-89-RFWKPEs4-2

<table>
<thead>
<tr>
<th>S0</th>
<th>0x0</th>
<th>0x1</th>
<th>0x2</th>
<th>0x3</th>
<th>0x4</th>
<th>0x5</th>
<th>0x6</th>
<th>0x7</th>
<th>0x8</th>
<th>0x9</th>
<th>0xA</th>
<th>0xB</th>
<th>0xC</th>
<th>0xD</th>
<th>0xE</th>
<th>0xF</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>0x5</td>
<td>0x4</td>
<td>0x3</td>
<td>0x2</td>
<td>0x1</td>
<td>0x0</td>
<td>0x9</td>
<td>0x8</td>
<td>0x7</td>
<td>0x6</td>
<td>0x5</td>
<td>0x4</td>
<td>0x3</td>
<td>0x2</td>
<td>0x1</td>
<td>0x0</td>
</tr>
<tr>
<td>S2</td>
<td>0xE</td>
<td>0xBD</td>
<td>0xEB</td>
<td>0xEA</td>
<td>0xEC</td>
<td>0xED</td>
<td>0xEE</td>
<td>0xEF</td>
<td>0xF0</td>
<td>0xF1</td>
<td>0xF2</td>
<td>0xF3</td>
<td>0xF4</td>
<td>0xF5</td>
<td>0xF6</td>
<td>0xF7</td>
</tr>
<tr>
<td>S3</td>
<td>0x8</td>
<td>0x9</td>
<td>0xA</td>
<td>0xB</td>
<td>0xC</td>
<td>0xD</td>
<td>0xE</td>
<td>0xF</td>
<td>0x0</td>
<td>0x1</td>
<td>0x2</td>
<td>0x3</td>
<td>0x4</td>
<td>0x5</td>
<td>0x6</td>
<td>0x7</td>
</tr>
<tr>
<td>S4</td>
<td>0xF</td>
<td>0x1</td>
<td>0x2</td>
<td>0x3</td>
<td>0x4</td>
<td>0x5</td>
<td>0x6</td>
<td>0x7</td>
<td>0x8</td>
<td>0x9</td>
<td>0xA</td>
<td>0xB</td>
<td>0xC</td>
<td>0xD</td>
<td>0xE</td>
<td>0xF</td>
</tr>
<tr>
<td>S5</td>
<td>0x1</td>
<td>0x2</td>
<td>0x3</td>
<td>0x4</td>
<td>0x5</td>
<td>0x6</td>
<td>0x7</td>
<td>0x8</td>
<td>0x9</td>
<td>0xA</td>
<td>0xB</td>
<td>0xC</td>
<td>0xD</td>
<td>0xE</td>
<td>0xF</td>
<td>0x0</td>
</tr>
<tr>
<td>S6</td>
<td>0x0</td>
<td>0x1</td>
<td>0x2</td>
<td>0x3</td>
<td>0x4</td>
<td>0x5</td>
<td>0x6</td>
<td>0x7</td>
<td>0x8</td>
<td>0x9</td>
<td>0xA</td>
<td>0xB</td>
<td>0xC</td>
<td>0xD</td>
<td>0xE</td>
<td>0xF</td>
</tr>
<tr>
<td>S7</td>
<td>0x7</td>
<td>0x8</td>
<td>0x9</td>
<td>0xA</td>
<td>0xB</td>
<td>0xC</td>
<td>0xD</td>
<td>0xE</td>
<td>0xF</td>
<td>0x0</td>
<td>0x1</td>
<td>0x2</td>
<td>0x3</td>
<td>0x4</td>
<td>0x5</td>
<td>0x6</td>
</tr>
<tr>
<td>S8</td>
<td>0xF</td>
<td>0x1</td>
<td>0x2</td>
<td>0x3</td>
<td>0x4</td>
<td>0x5</td>
<td>0x6</td>
<td>0x7</td>
<td>0x8</td>
<td>0x9</td>
<td>0xA</td>
<td>0xB</td>
<td>0xC</td>
<td>0xD</td>
<td>0xE</td>
<td>0xF</td>
</tr>
<tr>
<td>S9</td>
<td>0x5</td>
<td>0x4</td>
<td>0x3</td>
<td>0x2</td>
<td>0x1</td>
<td>0x0</td>
<td>0x9</td>
<td>0x8</td>
<td>0x7</td>
<td>0x6</td>
<td>0x5</td>
<td>0x4</td>
<td>0x3</td>
<td>0x2</td>
<td>0x1</td>
<td>0x0</td>
</tr>
<tr>
<td>SA7</td>
<td>0x9</td>
<td>0x8</td>
<td>0x7</td>
<td>0x6</td>
<td>0x5</td>
<td>0x4</td>
<td>0x3</td>
<td>0x2</td>
<td>0x1</td>
<td>0x0</td>
<td>0x9</td>
<td>0x8</td>
<td>0x7</td>
<td>0x6</td>
<td>0x5</td>
<td>0x4</td>
</tr>
<tr>
<td>S10</td>
<td>0x0</td>
<td>0x1</td>
<td>0x2</td>
<td>0x3</td>
<td>0x4</td>
<td>0x5</td>
<td>0x6</td>
<td>0x7</td>
<td>0x8</td>
<td>0x9</td>
<td>0xA</td>
<td>0xB</td>
<td>0xC</td>
<td>0xD</td>
<td>0xE</td>
<td>0xF</td>
</tr>
<tr>
<td>S11</td>
<td>0x1</td>
<td>0x2</td>
<td>0x3</td>
<td>0x4</td>
<td>0x5</td>
<td>0x6</td>
<td>0x7</td>
<td>0x8</td>
<td>0x9</td>
<td>0xA</td>
<td>0xB</td>
<td>0xC</td>
<td>0xD</td>
<td>0xE</td>
<td>0xF</td>
<td>0x0</td>
</tr>
<tr>
<td>S12</td>
<td>0x0</td>
<td>0x1</td>
<td>0x2</td>
<td>0x3</td>
<td>0x4</td>
<td>0x5</td>
<td>0x6</td>
<td>0x7</td>
<td>0x8</td>
<td>0x9</td>
<td>0xA</td>
<td>0xB</td>
<td>0xC</td>
<td>0xD</td>
<td>0xE</td>
<td>0xF</td>
</tr>
<tr>
<td>S13</td>
<td>0x1</td>
<td>0x2</td>
<td>0x3</td>
<td>0x4</td>
<td>0x5</td>
<td>0x6</td>
<td>0x7</td>
<td>0x8</td>
<td>0x9</td>
<td>0xA</td>
<td>0xB</td>
<td>0xC</td>
<td>0xD</td>
<td>0xE</td>
<td>0xF</td>
<td>0x0</td>
</tr>
<tr>
<td>S14</td>
<td>0x0</td>
<td>0x1</td>
<td>0x2</td>
<td>0x3</td>
<td>0x4</td>
<td>0x5</td>
<td>0x6</td>
<td>0x7</td>
<td>0x8</td>
<td>0x9</td>
<td>0xA</td>
<td>0xB</td>
<td>0xC</td>
<td>0xD</td>
<td>0xE</td>
<td>0xF</td>
</tr>
<tr>
<td>S15</td>
<td>0x0</td>
<td>0x1</td>
<td>0x2</td>
<td>0x3</td>
<td>0x4</td>
<td>0x5</td>
<td>0x6</td>
<td>0x7</td>
<td>0x8</td>
<td>0x9</td>
<td>0xA</td>
<td>0xB</td>
<td>0xC</td>
<td>0xD</td>
<td>0xE</td>
<td>0xF</td>
</tr>
</tbody>
</table>

© 2016 Global Journals Inc. (US)
32-bit subblocks \( S^0 \), \( S^1 \) divided into eight four bit subblocks
\[ S^0 = s^0_0 \oplus s^0_1 \oplus s^0_2 \oplus s^0_3 \oplus s^0_4 \oplus s^0_5 \oplus s^0_6 \oplus s^0_7, \]
\[ S^1 = s^1_0 \oplus s^1_1 \oplus s^1_2 \oplus s^1_3 \oplus s^1_4 \oplus s^1_5 \oplus s^1_6 \oplus s^1_7. \]
Four bit subblocks \( s^0_i \), \( s^1_i \), \( i = 0 \ldots 7 \) transformed into the S-boxes:
\[ R^j = S_{j}(s^0_i) \oplus S_{j}(s^1_i) \oplus S_{j}(s^2_i) \oplus S_{j}(s^3_i) \oplus S_{j}(s^4_i) \oplus S_{j}(s^5_i) \oplus S_{j}(s^6_i) \oplus S_{j}(s^7_i). \]

The resulting 32-bit subblocks \( R^0 \), \( R^1 \) cyclically shifted left by 11 bits and we obtain subblocks \( Y^0 \), \( Y^1 \):
\[ Y^0 = R^0 \lll 11, \quad Y^1 = R^1 \lll 11. \]
Then 32-bit subblocks \( Y^0 \), \( Y^1 \) divided into four 8-bit subblocks \( Y^0 \), \( Y^1 \), \( Y^7 \) i.e.,
\[ Y_0 = Y^0 \oplus Y^1 \oplus Y^2 \oplus Y^3, \quad Y_1 = Y^4 \oplus Y^5 \oplus Y^6 \oplus Y^7. \]

Consider the encryption process of encryption algorithm GOST28147-89-IDEA16-2. Initially the 128-bit plaintext \( X \) partitioned into subblocks of 8-bits \( X^0 \), \( X^1 \), \( X^2 \), ..., \( X^{15} \), and performs the following steps:

1. subblocks \( X^0 \), \( X^1 \), \( X^2 \), ..., \( X^{15} \) summed by XOR respectively with round key \( K_{24(n-1)} \), \( K_{24(n-1)+1} \), ..., \( K_{24(n-1)+15} \), \( i = 0 \ldots n+1 \) and calculated 8-bit subblocks \( T^0 \), \( T^1 \), \( T^2 \), ..., \( T^7 \). This step can be represented as follows:
\[ T_0 = (X^0_0 + K_{24(0)+1}) \oplus (X^0_1 \cdot K_{24(0)+8}), \]
\[ T_1 = (X^1_1 \cdot K_{24(1)+1}) \oplus (X^1_0 + K_{24(1)+9}), \]
\[ T_2 = (X^2_2 + K_{24(2)+1}) \oplus (X^2_1 \cdot K_{24(2)+10}), \]
\[ T_3 = (X^3_3 \cdot K_{24(3)+1}) \oplus (X^3_2 + K_{24(3)+11}), \]
\[ T_4 = (X^4_4 + K_{24(4)+1}) \oplus (X^4_3 \cdot K_{24(4)+12}), \]
\[ T_5 = (X^5_5 \cdot K_{24(5)+1}) \oplus (X^5_4 + K_{24(5)+13}), \]
\[ T_6 = (X^6_6 + K_{24(6)+1}) \oplus (X^6_5 \cdot K_{24(6)+14}), \]
\[ T_7 = (X^7_7 \cdot K_{24(7)+1}) \oplus (X^7_6 + K_{24(7)+15}), \]

3. to 8-bit subblocks \( T^0 \), \( T^1 \), \( T^2 \), ..., \( T^7 \) applied round functions and get 8-bit subblocks \( Y^0 \), \( Y^1 \), \( Y^2 \), ..., \( Y^7 \).

4. subblocks \( Y^0 \), \( Y^1 \), \( Y^2 \), ..., \( Y^7 \) are summed to XOR with subblocks \( X^0 \), \( X^1 \), \( X^2 \), ..., \( X^{15} \), \( \text{e} \).
\[ X'_j = X_j \oplus Y_j, \quad X'_{j+8} = X'_{j+8} \oplus Y_{j+8}, \]
5. at the end of the round subblocks swapped, i.e.,
\[ X'_j = X_{j+14}, \quad j = 1 \ldots 14, \quad i = 1. \]

6. repeating steps 2-5 \( n \) times, i.e., \( i = 2 \ldots n \) obtain subblocks \( X'_{i,j} \), \( X'_{i,15} \), \( \cdots, X'_{i,15} \).

7. in output transformation round keys \( K_{24(n+1)} \), \( K_{24(n+1)+1} \), \( \cdots, K_{24(n+1)+15} \) are multiplied and summed into subblocks, i.e.
\[ X_{i+1} = X_{i+1}^0 + K_{24(n+1)} \]
\[ X_{i+1}^0 = X_{i+1}^0 + K_{24(n+1)+1} \]
\[ X_{i+1}^n = X_{i+1}^n + K_{24(n+1)+n}, \]
8. subblocks \( X^0_{i+1} \), \( X^1_{i+1} \), \( \cdots, X^{15}_{i+1} \) are summed to XOR with the round \( K_{24(n+1)+16} \), \( K_{24(n+1)+17} \), \( K_{24(n+1)+18} \), \( \cdots, K_{24(n+1)+31} \), \( i = 0 \ldots 7 \).

As ciphertext plaintext \( X \) receives the combined 8-bit subblocks \( X^0_{i+1} \oplus X^1_{i+1} \oplus X^{12}_{i+1} \oplus \cdots \oplus X^{15}_{i+1} \).

III. Key Generation of the Encryption Algorithm GOST28147-89-IDEA16-2

In \( n \)-round encryption algorithm GOST28147-89-IDEA16-2 in each round used twenty four round keys of the 8-bit and output transformation sixteen round keys of the 8-bit. In addition, before the first round and after the output transformation we used sixteen round keys of 8-bits. Total number of 8-bit round keys is equal to \( 24n+48 \). In Figure 4 encryption used encryption round keys \( K^r \) instead of \( K^r \), while decryption used decryption round keys \( K^r \). If \( n=8 \) then need 240 to generate round keys, if \( n=12 \), you need to generate 336 round keys and if \( n=16 \) need 432 to generate round keys.

The key encryption algorithm \( K \) of length \( l \) (256 ≤ \( l \) ≤ 1024) bits is divided into 8-bit round keys \( K^r \), \( K^r \), ..., \( K^{r}_{\text{length-1}} \), \( \text{Lenght} = l / 8 \), here \( K = \{ k_0, k_1, \ldots, k_{l-1} \} \).
\[ K^r = \{ k_0, k_1, \ldots, k_{15} \}, \quad K^r = \{ k_1, k_3, \ldots, k_{13} \}, \]
\[ K^r_{\text{length-1}} = \{ k_2, k_4, \ldots, k_{14} \}. \]
Then we calculate \( K = K^r \oplus K^r \oplus \cdots \oplus K^r_{\text{length-1}} \). If \( K = 0 \) then \( K \) is chosen as 0xC5, i.e. \( K \) = 0xC5. Round keys \( K^r \), \( i = \text{Lenght}-24n+47 \) are computed as follows
The Encryption Algorithms GOST-IDEA16-2 and GOST-RFKWIDEA16-2

In the encryption algorithm GOST2814-89-RFKWIDEA16-2 the length of subblocks $X^0$, $X^1$, $X^2$, ..., $X^{15}$, length of round keys $K_{16(i-1)}$, $K_{16(i-1)+1}$, $K_{16(i-1)+2}$, ..., $K_{16(i-1)+15}$, $i = 1, \ldots n+1$, $X^0_{16+1}$, $X^{16}_{16+1}$, $\ldots$, are equal to 8-bits. The length of the input and output blocks of round functions is 32 bits. This encryption algorithm round function GOST 2814-89 is applied twice and in each round function employed eight S-boxes, i.e. the total number of S-boxes is 16. The structure of the encryption algorithm GOST2814-89-PES16-2 is shown in Figure 2 and the S-boxes shown in Table 1.

Consider the round function block encryption algorithm GOST2814-89-RFKWIDEA16-2. First the 8-bit subblocks $T^0$, $T^1$, ..., $T^7$ combined from 32-bit subblocks, i.e. $T^0 = T^0 || T^1 || T^2 || T^3$, $T^4 = T^4 || T^5 || T^6 || T^7$. 32-bit subblocks $T^0$, $T^i$, divided into eight four bit subblocks $T^0 = t^0_0 || t^0_1 || t^0_2 || t^0_3 || t^0_4 || t^0_5 || t^0_6 || t^0_7$, $T^i = t^i_0 || t^i_1 || t^i_2 || t^i_3$.

Four bit subblocks $t^0_0$, $t^0_1$, $i = 0 \ldots 7$ transformed into the S-boxes: $R^0 = S_0(t^0_0) || S_1(t^0_1) || S_2(t^0_2) || S_3(t^0_3) || S_4(t^0_4) || S_5(t^0_5) || S_6(t^0_6) || S_7(t^0_7)$.

The resulting 32-bit subblocks $R^0$, $R^1$ cyclically shifted left by 11 bits and we obtain subblocks $Y^0$, $Y^1$, $Y^2$, $Y^3$, $Y^4$, $Y^5$, $Y^6$ and $Y^7$. Thereafter 32-bit subblocks $Y^0$, $Y^i$, divided into four 8-bit subblocks $Y^0$, $Y^1$, ..., $Y^7$ i.e., $Y^0 = Y^0 || Y^1 || Y^2 || Y^3$, $Y^i = Y^4 || Y^5 || Y^6 || Y^7$.

Consider the encryption process of encryption algorithm GOST2814-89-IDEA16-2. Initially the 128-bit plaintext $\$X\$ partitioned into subblocks of 8-bits $X^0$, $X^1$, $X^2$, ..., $X^{15}$, and performs the following steps:

1. **Decryption round keys applied to the first round and after the output transformation associated with the encryption round keys as follows:** $K^c_{244+36+i} = K^c_{244+32+i}$
2. **Decryption round keys of the output transformation associate with the basis of encryption round keys $K^c$ and decryption round keys of the output transformation associate with of encryption round keys as follows:**

$$K_\text{enc} = Sbox0(K_{\text{enc}-\text{LongKey}}) \oplus Sbox1(\text{RotWord}(K_{\text{enc}-\text{LongKey}}))$$

$K_\text{enc}$, the encryption key is cyclic shift to the left of 1 bit of the 16-bit subblock, the basis of encryption round keys $K_{\text{enc}}$ and after the output transformation associated with the encryption round keys applied to the first round

$$K^d_{244+36+j} = K^c_{244+32+j}, \quad j = 0 \ldots 7.$$
1. subblocks $X_0^i$, $X_1^i$, $X_2^i$, ..., $X_{15}^i$ summed by XOR respectively with round key $K_{16(i-1)+i}$, $K_{16(i-1)+i+1}$, ...

   i.e. $X'_j = X_0^i \oplus K_{16(i-1)+j}$, $j = 0...7$

2. subblocks $X_0^i$, $X_1^i$, $X_2^i$, ..., $X_{15}^i$ multiplied and summed respectively with the round keys $K_{16(i-1)+1}$, $K_{16(i-1)+2}$, ..., $K_{16(i-1)+15}$ and calculated 8-bit subblocks $T_0$, $T_1$, $T_2$, ..., $T_7$. This step can be represented as follows:

   $T_0 = (X_0^i + K_{16(i-1)+1}) \oplus (X_1^i \cdot K_{16(i-1)+8})$,
   $T_1 = (X_1^i \cdot K_{16(i-1)+1}) \oplus (X_2^i + K_{16(i-1)+9})$,
   $T_2 = (X_2^i + K_{16(i-1)+2}) \oplus (X_3^i \cdot K_{16(i-1)+10})$,
   $T_3 = (X_3^i \cdot K_{16(i-1)+3}) \oplus (X_4^i + K_{16(i-1)+11})$,
   $T_4 = (X_4^i + K_{16(i-1)+4}) \oplus (X_5^i \cdot K_{16(i-1)+12})$,
   $T_5 = (X_5^i \cdot K_{16(i-1)+5}) \oplus (X_6^i + K_{16(i-1)+13})$,
   $T_6 = (X_6^i + K_{16(i-1)+6}) \oplus (X_7^i \cdot K_{16(i-1)+14})$,
   $T_7 = (X_7^i \cdot K_{16(i-1)+7}) \oplus (X_8^i + K_{16(i-1)+15})$, $i = 1$.

3. to 8-bit subblocks $T^0$, $T^1$, $T^2$, ..., $T^7$ applied round functions and get 8-bit subblocks $Y^0$, $Y^1$, $Y^2$, ..., $Y^7$.

4. subblocks $Y_0^i$, $Y_1^i$, $Y_2^i$, ..., $Y_7^i$ are summed to XOR with subblocks $X_0^i$, $X_1^i$, $X_2^i$, ..., $X_{15}^i$ i.e. $X'_j = X_0^i \oplus Y^j$, $X_1^i = X_1^i \oplus Y^{j-1}$, $X_2^i = X_2^i \oplus Y^{j-2}$, ...

5. at the end of the round subblocks swapped, i.e.,

   $X'_j = X_{15-i}^i$, $j = 1...14$, $i = 1$.

6. repeating steps 2-5 $n$ times, i.e., $i = 2...n$ obtain subblocks $X_0^i$, $X_1^i$, $X_2^i$, ..., $X_{15}^i$. 
From the 8-bit and output transformation sixteen round keys of

\[ \text{encryption algorithm GOST 28147-89, RWW/DIA16-2} \]

In round encryption algorithm GOST 28147-89, DIA16-2 in each round used sixteen round keys of the 8-bit. In addition, before the first round and after the output transformation we used sixteen round keys of 8-bit. Total number of 8-bit round keys is equal to

\[ n \times 16 = 8n \].

In Figure 4 encryption used round keys \( K' \) instead of \( K \), while decryption used round keys \( K' \). cipher text \( X' \) receives the combined 8-bit subblocks \( X'' \ldots X' \), are summed to XOR with the round key \( K'' \), are computed on the basis of encryption round keys \( K' \) and decryption round keys of the output transformation associate with encryption round keys as follows.

\[
\begin{align*}
\text{Decryption round keys of the first round:} \\
K'' &= (K''_1, \ldots, K''_n, \ldots, K''_n), \\
K'' &= (K''_1, \ldots, K''_n, \ldots, K''_n), \\
K' &= (K'_1, \ldots, K'_n, \ldots, K'_n), \\
K' &= (K'_1, \ldots, K'_n, \ldots, K'_n).
\end{align*}
\]

As a result of this study built a new block encryption algorithms called GOST 28147-89, DIA16-2. The algorithm is based on a networks, DIA16-2 and RWW/DIA16-2 and GOST 28147-89, RWW/DIA16-2 using the round function of GOST 28147-89. Length of rounds and key lengths is variable, wherein the user rounds and the output transformation associated with the encryption round keys as follows: \( K'_1 = K'_1 \).
following Table 2 summarizes options openly declared S-box such as: \( \text{deg} \) -degree of algebraic nonlinearity; \( NL \) -nonlinearity; \( \lambda \) -resistance to linear cryptanalysis; \( \delta \) -resistance to differential cryptanalysis; SAC-strict avalanche criterion; BIC-bit independence criterion. To S-box was resistant to cryptanalysis it is necessary that the values \( \text{deg} \) and \( NL \) were large, and the values \( \lambda \), \( \delta \), SAC and BIC small. In block cipher algorithms GOST28147-89-IDEA16-2 and GOST28147-89-RFWKIDEA16-2 for all S-boxes, the following equation: \( \text{deg} = 3, \ NL = 4, \ \lambda = 0.5, \ \delta = 3/8, \ SAC \leq 2, \ BIC \leq 4 \), i.e. resistance is not lower than the algorithm GOST 28147-89. These S-boxes are created based on Nyberg construction [3].

Table 2: Parameters of the S-boxes encryption algorithm GOST 28147-89

<table>
<thead>
<tr>
<th>No</th>
<th>Parameters</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>deg</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>NL</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>( \lambda )</td>
<td>0.5</td>
<td>3/4</td>
<td>3/4</td>
<td>3/4</td>
<td>3/4</td>
<td>3/4</td>
<td>3/4</td>
<td>3/4</td>
</tr>
<tr>
<td>4</td>
<td>( \delta )</td>
<td>3/8</td>
<td>3/8</td>
<td>3/8</td>
<td>3/8</td>
<td>1/4</td>
<td>3/8</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>SAC</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>BIC</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

IV. CONCLUSIONS

In this way, built a new block encryption algorithms called GOST28147-89-IDEA16-2 and GOST28147-89-RFWKIDEA16-2 based on networks IDEA16-2 and RFWKIDEA16-2 using the round function of GOST 28147-89. Installed that the resistance offered by the author block cipher algorithm not lower than the resistance of the algorithm GOST 28147-89.

REFERENCES RÉFÉRENCES REFERENCIAS

5. Tuychiev G.N. The networks RFWKIDEA4–2, IDEA4–1 and RFWKIDEA4–1 // Acta of Turin polytechnic university in Tashkent, 2013, №3 pp. 71-77
7. Tuychiev G.N. About networks PES4–1 and RFWKPES4–2, RFWKPES4–1 developed on the basis of network PES4–2 // Uzbek journal of the problems of informatics and energetics. –Tashkent, 2015, №12, pp. 100-105