Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 16 Issue 1 Version 1.0 Year 2016
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

The Encryption Algorithms GOST28147-89-IDEA8-4 and GOST28147-89-RFWKIDEA8-4

By Gulom Tuychiev

National University of Uzbekistan

Abstract- In the paper created a new encryption algorithms GOST28147–89–IDEA8–4 and GOST28147–89–RFWKIDEA8– 4 based on networks IDEA8–4 and RFWKIDEA8–4, with the use the round function of the encryption algorithm GOST 28147–89. The block length of created block encryption algorithm is 256 bits, the number of rounds is 8, 12 and 16.

Keywords: feystel network, lai–massey scheme, round function, round keys, output transformation, multipli-cation, addition, s–box.

GJCST-C Classification: E.3

Strictly as per the compliance and regulations of:

The Encryption Algorithms GOST28147–89–IDEA8–4 and GOST28147–89–RFWKIDEA8–4

Gulom Tuychiev

Abstract- **In the paper created a new encryption algorithms GOST28147–89–IDEA8–4 and GOST28147–89–RFWKIDEA8–4 based on networks IDEA8–4 and RFWKIDEA8–4, with the use the round function of the encryption algorithm GOST 28147–89. The block length of created block encryption algorithm is 256 bits, the number of rounds is 8, 12 and 16.**

Keywords: feystel network, lai–massey scheme, round function, round keys, output transformation, multiplication, addition, s–box.

I. Introduction

The encryption algorithm GOST 28147–89 [4] 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 Lai–Massey scheme developed the networks IDEA8–4 [6] and RFWKIDEA8–4 [7], consisting from four round function. In the networks IDEA8–4 and RFWKIDEA8–4, similarly as in the Feistel network, in encryption and decryption using the same algorithm. In the networks used four round function having one 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 [8], RFWKPES4–2 [8], PES8–4 [2], RFWKPES8–4 [10], IDEA16–2 [11], RFWKIDEA16–2 [12] encryption algorithm GOST 28147–89 created the encryption algorithm GOST28147–89–IDEA4–2 [13], GOST28147–89–RFWKIDEA4–2 [14], GOST28147–89–PES4–2 [15], GOST28147–89–RFWKPES4–2 [16], GOST28147–89–PES8–4, GOST28147–89–RFWKPES8–4 [17], GOST28147–89–IDEA16–2, GOST28147–89–RFWKIDEA16–2 [18].

Author: Technical Sciences (Ph.D.), National University of Uzbekistan. He received Pd.D. degree in specialty mathematic from the National University of Uzbekistan. e- mail: blasterjon@gmail.com

In this paper, applying the round function of the encryption algorithm GOST 28147–89 as round functions of the networks IDEA8–4 and RFWKIDEA8–4, developed new encryption algorithms GOST28147–89–IDEA8–4 and GOST28147–89–RFWKI DEA8–4. In the encryption algorithms GOST28147–89–IDEA8–4 and GOST28147–89–RFWKIDEA8–4 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 Encryption Algorithm GOST28147–89–IDEA8–4

The structure of the encryption algorithm GOST28147–89–IDEA8–4. In the encryption algorithm GOST28147–89–IDEA8–4 length of the subblocks X^0, X^1, …, X^7, length of the round keys $K_{12(i-1)}$, $K_{12(i-1)+1}$, …, $K_{12(i-1)+7}$, $i=\overline{1...n+1}$, $K_{12(i-1)+8}$, $K_{12(i-1)+9}$, $K_{12(i-1)+10}$, $K_{12(i-1)+11}$, $i=\overline{1...n}$ and K_{12n+8}, K_{12n+9}, ..., K_{12n+23} are equal to 32–bits. In this encryption algorithm the round function GOST 28147–89 is applied four time and in each round function used eight S–boxes, i.e. the total number of S–boxes is 32. The structure of the encryption algorithm GOST28147–89–IDEA8–4 is shown in Figure 1 and the S–boxes shown in Table 1.

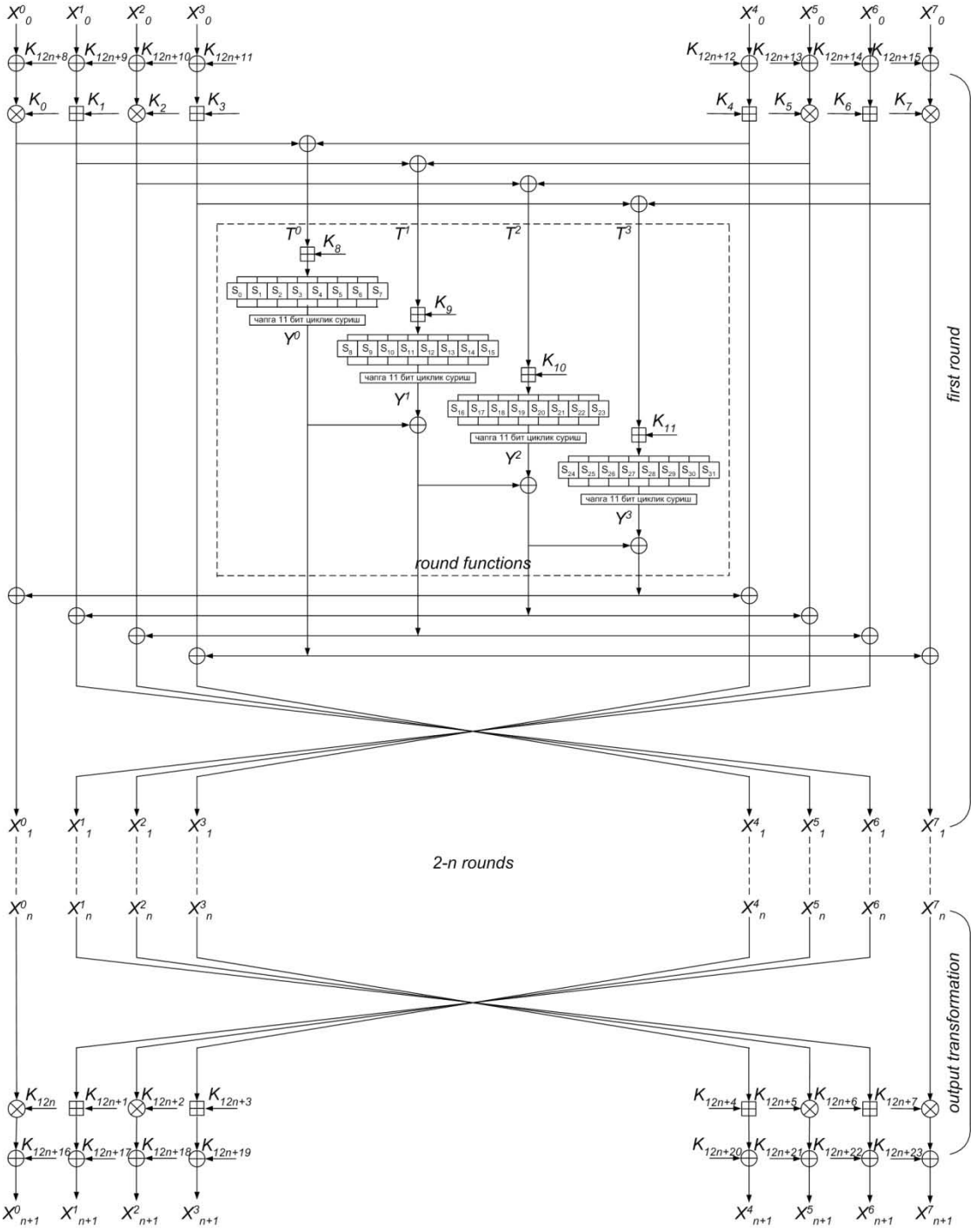

Figure 1: The scheme n–rounded encryption algorithm GOST28147–89–IDEA8–4

Consider the round function of a encryption algorithm GOST28147–89–IDEA8–4. The 32–bit subblocks T^0, T^1, T^2, T^3 are summed round keys $K_{12(i-1)+8}$, $K_{12(i-1)+9}$, $K_{12(i-1)+10}$, $K_{12(i-1)+11}$, $i=\overline{1...n}$, i.e. $S^0 = T^0 + K_{12(i-1)+8}$, $S^1 = T^1 + K_{12(i-1)+9}$, $S^2 = T^2 + K_{12(i-1)+10}$, $S^3 = T^3 + K_{12(i-1)+11}$. 32–bit subblocks S^0, S^1, S^2, S^3 divided into eight four–bit subblocks, i.e. $S^0 = s_0^0 \| s_1^0 \| s_2^0 \| s_3^0 \| s_4^0 \| s_5^0 \| s_6^0 \| s_7^0$, $S^1 = s_0^1 \| s_1^1 \| s_2^1 \| s_3^1 \| s_4^1 \| s_5^1 \| s_6^1 \| s_7^1$, $S^2 = s_0^2 \| s_1^2 \| s_2^2 \| s_3^2 \| s_4^2 \| s_5^2 \| s_6^2 \| s_7^2$, $S^3 = s_0^3 \| s_1^3 \| s_2^3 \| s_3^3 \| s_4^3 \| s_5^3 \| s_6^3 \| s_7^3$. The four–bit subblocks s_i^0, s_i^1, s_i^2, s_i^3, $i=\overline{0...7}$ transformed into the S–boxes: $R^0 = S_0(s_0^0) \| S_1(s_1^0) \| S_2(s_2^0) \| S_3(s_3^0) \| S_4(s_4^0) \| S_5(s_5^0) \| S_6(s_6^0) \| S_7(s_7^0)$, $R^1 = S_8(s_0^1) \| S_9(s_1^1) \| S_{10}(s_2^1) \| S_{11}(s_3^1) \| S_{12}(s_4^1) \| S_{13}(s_5^1) \| S_{14}(s_6^1) \| S_{15}(s_7^1)$, $R^2 = S_{16}(s_0^2) \| S_{17}(s_1^2) \| S_{18}(s_2^2) \| S_{19}(s_3^2) \| S_{20}(s_4^2) \| S_{21}(s_5^2) \|$

$S_{22}(s_6^2) \| S_{23}(s_7^2)$, $R^3 = S_{24}(s_0^3) \| S_{25}(s_1^3) \| S_{26}(s_2^3) \| S_{27}(s_3^3) \| S_{28}(s_4^3) \| S_{29}(s_5^3) \| S_{30}(s_6^3) \| S_{31}(s_7^3)$. The resulting 32–bit subblocks R^0, R^1, R^2, R^3 cyclically shifted left by 11 bits and obtain subblocks Y^0, Y^1, Y^2, Y^3: $Y^0 = R^0 <<11$, $Y^1 = R^1 <<11$, $Y^2 = R^2 <<11$, $Y^3 = R^3 <<11$.

Table 1: The S–box of encryption algorithms

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
S0	0x4	0x5	0xA	0x8	0xD	0x9	0xE	0x2	0x6	0xF	0xC	0x7	0x0	0x3	0x1	0xB
S1	0x5	0x4	0xB	0x9	0xC	0x8	0xF	0x3	0x7	0xE	0xD	0x6	0x1	0x2	0x0	0xA
S2	0x6	0x7	0x8	0xA	0xF	0xB	0xC	0x0	0x4	0xD	0xE	0x5	0x2	0x1	0x3	0x9
S3	0x7	0x6	0x9	0xB	0xE	0xA	0xD	0x1	0x5	0xC	0xF	0x4	0x3	0x0	0x2	0x8
S4	0x8	0x9	0x6	0x4	0x1	0x5	0x2	0xE	0xA	0x3	0x0	0xB	0xC	0xF	0xD	0x7
S5	0x9	0x8	0x7	0x5	0x0	0x4	0x3	0xF	0xB	0x2	0x1	0xA	0xD	0xE	0xC	0x6
S6	0xA	0xB	0x4	0x6	0x3	0x7	0x0	0xC	0x8	0x1	0x2	0x9	0xE	0xD	0xF	0x5
S7	0xB	0xA	0x5	0x7	0x2	0x6	0x1	0xD	0x9	0x0	0x3	0x8	0xF	0xC	0xE	0x4
S8	0xC	0xD	0x2	0x0	0x5	0x1	0x6	0xA	0xE	0x7	0x4	0xF	0x8	0xB	0x9	0x3
S9	0xE	0xF	0x0	0x2	0x7	0x3	0x4	0x8	0xC	0x5	0x6	0xD	0xA	0x9	0xB	0x1
S10	0xF	0xE	0x1	0x3	0x6	0x2	0x5	0x9	0xD	0x4	0x7	0xC	0xB	0x8	0xA	0x0
S11	0x1	0x8	0x7	0xD	0x0	0x4	0x3	0xF	0xB	0xA	0x9	0x2	0x5	0x6	0xC	0xE
S12	0x2	0xB	0x4	0xE	0x3	0x7	0x0	0xC	0x8	0x9	0xA	0x1	0x6	0x5	0xF	0xD
S13	0x3	0xA	0x5	0xF	0x2	0x6	0x1	0xD	0x9	0x8	0xB	0x0	0x7	0x4	0xE	0xC
S14	0x4	0x5	0xA	0x0	0xD	0x1	0x6	0x2	0xE	0x7	0xC	0xF	0x8	0x3	0x9	0xB
S15	0x5	0x4	0xB	0x1	0xC	0x0	0x7	0x3	0xF	0x6	0xD	0xE	0x9	0x2	0x8	0xA
S16	0x6	0x7	0x8	0x2	0xF	0x3	0x4	0x0	0xC	0x5	0xE	0xD	0xA	0x1	0xB	0x9
S17	0x7	0x6	0x9	0x3	0xE	0x2	0x5	0x1	0xD	0x4	0xF	0xC	0xB	0x0	0xA	0x8
S18	0x8	0x9	0x6	0xC	0x1	0xD	0xA	0xE	0x2	0xB	0x0	0x3	0x4	0xF	0x5	0x7
S19	0x9	0x8	0x7	0xD	0x0	0xC	0xB	0xF	0x3	0xA	0x1	0x2	0x5	0xE	0x4	0x6
S20	0xA	0xB	0x4	0xE	0x3	0xF	0x8	0xC	0x0	0x9	0x2	0x1	0x6	0xD	0x7	0x5
S21	0xB	0xA	0x5	0xF	0x2	0xE	0x9	0xD	0x1	0x8	0x3	0x0	0x7	0xC	0x6	0x4
S22	0xC	0xD	0x2	0x8	0x5	0x9	0xE	0xA	0x6	0xF	0x4	0x7	0x0	0xB	0x1	0x3
S23	0xD	0xC	0x3	0x9	0x4	0x8	0xF	0xB	0x7	0xE	0x5	0x6	0x1	0xA	0x0	0x2
S24	0x1	0x8	0x7	0x5	0x0	0xC	0xB	0xF	0x3	0x2	0x9	0xA	0xD	0x6	0x4	0xE
S25	0x2	0xB	0x4	0x6	0x3	0xF	0x8	0xC	0x0	0x1	0xA	0x9	0xE	0x5	0x7	0xD
S26	0x3	0xA	0x5	0x7	0x2	0xE	0x9	0xD	0x1	0x0	0xB	0x8	0xF	0x4	0x6	0xC
S27	0xF	0xE	0x1	0xB	0x6	0xA	0xD	0x9	0x5	0xC	0x7	0x4	0x3	0x8	0x2	0x0
S28	0xE	0xF	0x0	0xA	0x7	0xB	0xC	0x8	0x4	0xD	0x6	0x5	0x2	0x9	0x3	0x1
S29	0xA	0xB	0xC	0xE	0x3	0xF	0x0	0x4	0x8	0x1	0x2	0x9	0x6	0x5	0x7	0xD
S30	0xB	0xA	0xD	0xF	0x2	0xE	0x1	0x5	0x9	0x0	0x3	0x8	0x7	0x4	0x6	0xC
S31	0xC	0xD	0xA	0x8	0x5	0x9	0x6	0x2	0xE	0x7	0x4	0xF	0x0	0x3	0x1	0xB

Consider the encryption process of encryption algorithm GOST28147–89–IDEA8–4. Initially the 256–bit plaintext *X* partitioned into subblocks of 32–bits X_0^0, X_0^1, …, X_0^7 and runs the following steps:

1. subblocks X_0^0, X_0^1, …, X_0^7 summed by XOR with the keys K_{12n+8}, K_{12n+9}, …, K_{12n+15}: $X_0^j = X_0^j \oplus K_{12n+8+j}$, $j = \overline{0...7}$.
2. subblocks X_0^0, X_0^1, …, X_0^7 multiplied and summed with the round keys $K_{12(i-1)}$, $K_{12(i-1)+1}$, …, $K_{12(i-1)+7}$ and calculated 32–bit subblocks T^0, T^1, T^2, T^3 as follows: $T^0 = (X_{i-1}^0 \cdot K_{12(i-1)}) \oplus (X_{i-1}^4 + K_{12(i-1)+4})$, $T^1 = (X_{i-1}^1 \cdot K_{12(i-1)+1}) \oplus (X_{i-1}^5 + K_{12(i-1)+5})$, $T^2 = (X_{i-1}^2 \cdot K_{12(i-1)+2}) \oplus (X_{i-1}^6 + K_{12(i-1)+6})$, $T^3 = (X_{i-1}^3 \cdot K_{12(i-1)+3}) \oplus (X_{i-1}^7 + K_{12(i-1)+7})$, $i = 1$
3. to sublocks T^0, T^1, T^2, T^3 applying the round function and get the 32–bit subblocks Y^0, Y^1, Y^2, Y^3.
4. subblocks Y^0, Y^1, Y^2, Y^3 are summed to XOR with subblocks X_{i-1}^0, X_{i-1}^1, …, X_{i-1}^7, i.e. $X_{i-1}^0 = X_{i-1}^0 \oplus Y^3$, $X_{i-1}^1 = X_{i-1}^1 \oplus Y^2$, $X_{i-1}^2 = X_{i-1}^2 \oplus Y^1$, $X_{i-1}^3 = X_{i-1}^3 \oplus Y^0$, $X_{i-1}^4 = X_{i-1}^4 \oplus Y^3$, $X_{i-1}^5 = X_{i-1}^5 \oplus Y^2$, $X_{i-1}^6 = X_{i-1}^6 \oplus Y^1$, $X_{i-1}^7 = X_{i-1}^7 \oplus Y^0$, $i = 1$.
5. At the end of the round subblocks swapped, i.e, $X_i^0 = X_{i-1}^0$, $X_i^1 = X_{i-1}^6$, $X_i^2 = X_{i-1}^5$, $X_i^3 = X_{i-1}^4$, $X_i^4 = X_{i-1}^3$, $X_i^5 = X_{i-1}^2$, $X_i^6 = X_{i-1}^1$, $X_i^7 = X_{i-1}^7$, $i = 1$.

6. repeating the steps 2–5 n time, i.e. $i=\overline{2...n}$, obtained the subblocks X_n^0, X_n^1, …, X_n^7.
7. in output transformation round keys K_{12n}, K_{12n+1}, …, K_{12n+7} are multiplied and summed into subblocks X_n^0, X_n^1, …, X_n^7, i.e. $X_{n+1}^0 = X_n^0 \cdot K_{12n}$, $X_{n+1}^1 = X_n^6 + K_{12n+1}$, $X_{n+1}^2 = X_n^5 \cdot K_{12n+2}$, $X_{n+1}^3 = X_n^4 + K_{12n+3}$, $X_{n+1}^4 = X_n^3 + K_{12n+4}$, $X_{n+1}^5 = X_n^2 \cdot K_{12n+5}$, $X_{n+1}^6 = X_n^1 + K_{12n+6}$, $X_{n+1}^7 = X_n^7 \cdot K_{12n+7}$.
8. subblocks X_{n+1}^0, X_{n+1}^1, …, X_{n+1}^7 are summed by XOR with the round keys K_{12n+16}, K_{12n+17}, .., K_{12n+23}: $X_{n+1}^j = X_{n+1}^j \oplus K_{12n+16+j}$, $j=\overline{0...7}$.

As ciphertext receives the combined 32–bit subblocks $X_{n+1}^0 \| X_{n+1}^1 \| X_{n+1}^2 \| ... \| X_{n+1}^7$.

In the encryption algorithm GOST28147–89–IDEA8–4 when encryption and decryption using the same algorithm, only when decryption calculates the inverse of round keys depending on operations and are applied in reverse order. One important goal of encryption is key generation.

Key generation of the encryption algorithm GOST28147–89–IDEA8–4. In the n–round encryption algorithm GOST28147–89–IDEA8–4 used in each round 12 round keys of 32 bits and the output transformation of 8 round keys of 32 bits. In addition, prior to the first round and after the output transformation is applied 8 round keys on 32 bits. The total number of 32–bit round keys is equal to 12n+24. Hence, if n=8 then necessary 120, if n=12 then 168 and if n=16 then 216 to generate round keys.

The key of the encryption algorithm length of l ($256 \le l \le 1024$) bits is divided into 32–bit round keys K_0^c, K_1^c, …, $K_{Lenght-1}^c$, $Lenght = l/32$, here $K=\{k_0,k_1,...,k_{l-1}\}$, $K_0^c=\{k_0,k_1,...,k_{31}\}$, $K_1^c=\{k_{32},k_{33},...,k_{63}\}$, …, $K_{Lenght-1}^c=\{k_{l-32},k_{l-31},...,k_{l-1}\}$. Then calculated $K_L = K_0^c \oplus K_1^c \oplus ... \oplus K_{Lenght-1}^c$. If $K_L = 0$ then as K_L selected 0xC5C31537, i.e. $K_L = 0xC5C31537$. Round keys K_i^c, $i=\overline{Lenght...12n+23}$ calculated as follows: $K_i^c = SBox0(K_{i-Lenght}^c) \oplus SBox1(RotWord32(K_{i-Lenght+1}^c)) \oplus K_L$. After each generation of round keys value K_L cyclically shifted left by 1 bit. Here *RotWord32()*–cyclic shift 32 bit subblock to the left by 1 bit, *SBox()*–convert 32–bit subblock in S–box and $SBox0(A) = S_0(a_0) \| S_1(a_1) \| S_2(a_2) \| S_3(a_3) \| S_4(a_4) \| S_5(a_5) \| S_6(a_6) \| S_7(a_7)$, $SBox1(A) = S_8(a_0) \| S_9(a_1) \| S_{10}(a_2) \| S_{11}(a_3) \| S_{12}(a_4) \| S_{13}(a_5) \| S_{14}(a_6) \| S_{15}(a_7)$, $A = a_0 \| a_1 \| a_2 \| a_3 \| a_4 \| a_5 \| a_6 \| a_7$ and a_i – the four–bit sub–block.

Decryption round keys are computed on the basis of encryption round keys and decryption round keys output transformation associate with of encryption round keys as follows:

$$(K_{12n}^d, K_{12n+1}^d, K_{12n+2}^d, K_{12n+3}^d, K_{12n+4}^d, K_{12n+5}^d, K_{12n+6}^d, K_{12n+7}^d) = ((K_0^c)^{-1}, -K_1^c, (K_2^c)^{-1}, -K_3^c, -K_4^c, (K_5^c)^{-1}, -K_6^c, (K_7^c)^{-1}).$$

Decryption round keys of the second, third and *n*–round associates with the encryption round keys as follows:

$$(K_{12(i-1)}^d, K_{12(i-1)+1}^d, K_{12(i-1)+2}^d, K_{12(i-1)+3}^d, K_{12(i-1)+4}^d, K_{12(i-1)+5}^d, K_{12(i-1)+6}^d, K_{12(i-1)+7}^d, K_{12(i-1)+8}^d, K_{12(i-1)+9}^d, K_{12(i-1)+10}^d, K_{12(i-1)+11}^d) = ((K_{12(n-i+1)}^c)^{-1}, -K_{6(n-i+1)+6}^c, (K_{12(n-i+1)+5}^c)^{-1}, -K_{12(n-i+1)+4}^c, -K_{12(n-i+1)+3}^c, (K_{6(n-i+1)+2}^c)^{-1}, -K_{12(n-i+1)+1}^c, (K_{12(n-i+1)+7}^c)^{-1}, K_{12(n-i)+8}^c, K_{12(n-i)+9}^c, K_{12(n-i)+10}^c, K_{12(n-i)+11}^c), i=\overline{2...n}.$$

Decryption keys of the first round associated with the encryption keys as follows:

$$(K_0^d, K_1^d, K_2^d, K_3^d, K_4^d, K_5^d, K_6^d, K_7^d, K_8^d, K_9^d, K_{10}^d, K_{11}^d) = ((K_{12n}^c)^{-1}, -K_{12n+1}^c, (K_{12n+2}^c)^{-1}, -K_{12n+3}^c, -K_{12n+4}^c, (K_{12n+5}^c)^{-1}, -K_{12n+6}^c, (K_{12n+7}^c)^{-1}, K_{12(n-1)+8}^c, K_{12(n-1)+9}^c, K_{12(n-1)+10}^c, K_{12(n-1)+11}^c).$$

Decryption round keys applied to the first round and after the conversion of the output associated with encryption keys as follows: $K_{12n+8+j}^d = K_{12n+16+j}^c$, $K_{12n+16+j}^d = K_{12n+8+j}^c$, $j=\overline{0...7}$.

III. The Encryption Algorithm GOST28147–89–RFWKIDEA8–4.

The structure of the encryption algorithm GOST28147–89–RFWKIDEA8–4. In the encryption algorithm GOST28147–89–RFWKIDEA8–4 length of the subblocks X^0, X^1, …, X^7, length of the round keys $K_{8(i-1)}$, $K_{8(i-1)+1}$, …, $K_{8(i-1)+7}$, $i=\overline{1...n+1}$, K_{8n+8}, K_{8n+5}, …, K_{8n+23} are equal to 32–bits. In this encryption algorithm the round function GOST 28147–89 is applied four time and in each round function used eight S–boxes, i.e. the total number of S–boxes is 32. The structure of the encryption algorithm GOST28147–89–IDEA8–4 is shown in Figure 2 and the S–boxes shown in Table 1.

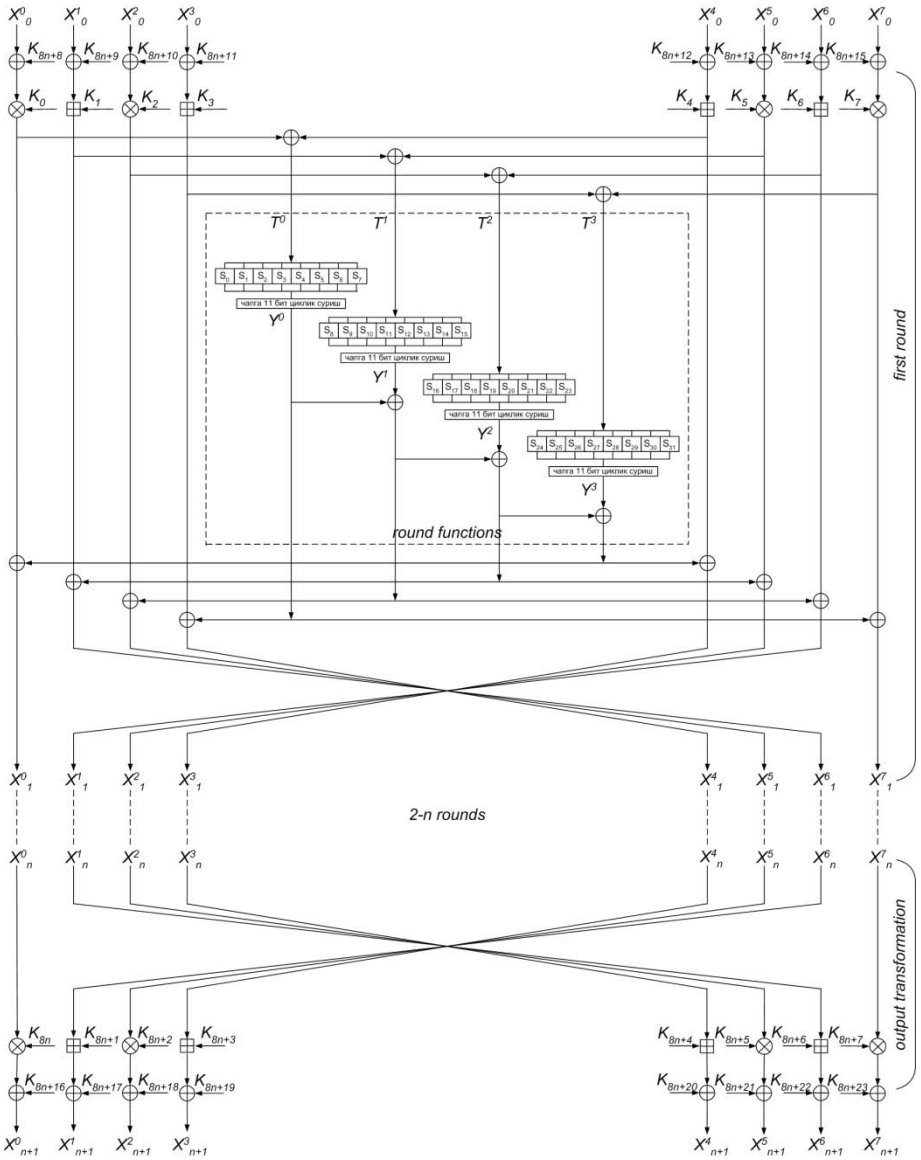

Figure 2: The scheme n–rounded encryption algorithm GOST28147–89–RFWKIDEA8–4

Consider the round function of encryption algorithm GOST28147–89–RFWKIDEA8–4. First 32–bit subblocks T^0, T^1, T^2, T^3 divided into eight four–bit sub–blocks, i.e. $T^0 = t_0^0 \| t_1^0 \| t_2^0 \| t_3^0 \| t_4^0 \| t_5^0 \| t_6^0 \| t_7^0$, $T^1 = t_0^1 \| t_1^1 \| t_2^1 \| t_3^1 \| t_4^1 \| t_5^1 \| t_6^1 \| t_7^1$, $T^2 = t_0^2 \| t_1^2 \| t_2^2 \| t_3^2 \| t_4^2 \| t_5^2 \| t_6^2 \| t_7^2$, $T^3 = t_0^3 \| t_1^3 \| t_2^3 \| t_3^3 \| t_4^3 \| t_5^3 \| t_6^3 \| t_7^3$. The four–bit subblocks t_i^0, t_i^1, t_i^2, t_i^3, $i = \overline{0...7}$ converted to S–box: $R^0 = S_0(t_0^0) \| S_1(t_1^0) \| S_2(t_2^0) \| S_3(t_3^0) \| S_4(t_4^0) \| S_5(t_5^0) \| S_6(t_6^0) \| S_7(t_7^0)$, $R^1 = S_8(t_0^1) \| S_9(t_1^1) \| S_{10}(t_2^1) \| S_{11}(t_3^1) \| S_{12}(t_4^1) \| S_{13}(t_5^1) \| S_{14}(t_6^1) \| S_{15}(t_7^1)$, $R^2 = S_{16}(t_0^2) \| S_{17}(t_1^2) \| S_{18}(t_2^2) \| S_{19}(t_3^2) \| S_{20}(t_4^2) \| S_{21}(t_5^2) \| S_{22}(t_6^2) \| S_{23}(t_7^2)$, $R^0 = S_{24}(t_0^3) \| S_{25}(t_1^3) \| S_{26}(t_2^3) \| S_{27}(t_3^3) \| S_{28}(t_4^3) \| S_{29}(t_5^3) \| S_{30}(t_6^3) \| S_{31}(t_7^3)$. Received 32–bit subblocks R^0, R^1, R^2, R^3 cyclically shifted to the left by 11 bits and get the subblocks Y^0, Y^1, Y^2, Y^3: $Y^0 = R^0 <<11$, $Y^1 = R^1 <<11$, $Y^2 = R^2 <<11$, $Y^3 = R^3 <<11$.

Consider the encryption process of encryption algorithm GOST28147–89–RFWKIDEA8–4. Initially the 256–bit plaintext *X* partitioned into subblocks of 32–bits X_0^0, X_0^1, …, X_0^7 and performs the following steps:

1. sublocks X_0^0, X_0^1, …, X_0^7 summed by XOR with the round keys K_{8n+8}, K_{8n+9}, …, K_{8n+15}: $X_0^j = X_0^j \oplus K_{8n+8+j}$, $j = \overline{0...7}$.
2. sublocks X_0^0, X_0^1, …, X_0^7 are multiplied and summed to the round keys $K_{8(i-1)}$, $K_{8(i-1)+1}$, …, $K_{8(i-1)+7}$ and calculates a 32–bit subblocks T^0, T^1, T^2, T^3 as follows: $T^0 = (X_{i-1}^0 \cdot K_{8(i-1)}) \oplus (X_{i-1}^4 + K_{8(i-1)+4})$,

$T^1 = (X^1_{i-1} \cdot K_{8(i-1)+1}) \oplus (X^5_{i-1} + K_{8(i-1)+5})$,
$T^2 = (X^2_{i-1} \cdot K_{8(i-1)+2}) \oplus (X^6_{i-1} + K_{8(i-1)+6})$,
$T^3 = (X^3_{i-1} \cdot K_{8(i-1)+3}) \oplus (X^7_{i-1} + K_{8(i-1)+7})$, $i = 1$.

3. to sublocks T^0, T^1, T^2, T^3 applying the round function and get the 32–bit subblocks Y^0, Y^1, Y^2 Y^3.
4. subblocks Y^0, Y^1, Y^2, Y^3 are summed to XOR with subblocks X^0_{i-1}, X^1_{i-1}, …, X^7_{i-1}, i.e. $X^0_{i-1} = X^0_{i-1} \oplus Y^3$ $X^1_{i-1} = X^1_{i-1} \oplus Y^2$, $X^2_{i-1} = X^2_{i-1} \oplus Y^1$, $X^3_{i-1} = X^3_{i-1} \oplus Y^0$ $X^4_{i-1} = X^4_{i-1} \oplus Y^3$, $X^5_{i-1} = X^5_{i-1} \oplus Y^2$, $X^6_{i-1} = X^6_{i-1} \oplus Y^1$ $X^7_{i-1} = X^7_{i-1} \oplus Y^0$, $i = 1$.
5. At the end of the round subblocks swapped, i.e, $X^0_i = X^0_{i-1}$, $X^1_i = X^6_{i-1}$, $X^2_i = X^5_{i-1}$, $X^3_i = X^4_{i-1}$ $X^4_i = X^3_{i-1}$, $X^5_i = X^2_{i-1}$, $X^6_i = X^1_{i-1}$, $X^7_i = X^7_{i-1}$, $i = 1$.
6. repeating the steps 2–5 n time, i.e. $i = \overline{2...n}$, obtained the subblocks X^0_n, X^1_n, …, X^7_n
7. in output transformation round keys K_{8n}, K_{8n+1}, …, K_{8n+7} are multiplied and summed into subblocks X^0_n, X^1_n, …, X^7_n, i.e. $X^0_{n+1} = X^0_n \cdot K_{8n}$, $X^1_{n+1} = X^6_n + K_{8n+1}$, $X^2_{n+1} = X^5_n \cdot K_{8n+2}$, $X^3_{n+1} = X^4_n + K_{8n+3}$, $X^4_{n+1} = X^3_n + K_{8n+4}$, $X^5_{n+1} = X^2_n \cdot K_{8n+5}$, $X^6_{n+1} = X^1_n + K_{8n+6}$, $X^7_{n+1} = X^7_n \cdot K_{8n+7}$.
8. subblocks X^0_{n+1}, X^1_{n+1}, ..., X^7_{n+1} are summed by XOR with the round keys K_{8n+16}, K_{8n+17}, .., K_{8n+23}: $X^j_{n+1} = X^j_{n+1} \oplus K_{8n+16+j}$, $j = \overline{0...7}$.

As ciphertext receives the combined 32–bit subblocks $X^0_{n+1} \| X^1_{n+1} \| X^2_{n+1} \| ... \| X^7_{n+1}$.

In the encryption algorithm GOST28147–89–RFWKIDEA8–4 when encryption and decryption using the same algorithm, only when decryption calculates the inverse of round keys depending on operations and are applied in reverse order. One important goal of encryption is key generation.

Key generation of the encryption algorithm GOST28147–89–RFWKIDEA8–4. In the n–round encryption algorithm GOST28147–89–RFWKIDEA8–4 used in each round 8 round keys of 32 bits and the output transformation of 8 round keys of 32 bits. In addition, prior to the first round and after the output transformation is applied 8 round keys on 32 bits. The total number of 32–bit round keys is equal to 8n+24.

The key length of the encryption algorithm l ($256 \le l \le 1024$) bits is divided into 32–bit round keys K^c_0, K^c_1, …, $K^c_{Lenght-1}$, $Lenght = l/32$, here $K = \{k_0, k_1, ..., k_{l-1}\}$, $K^c_0 = \{k_0, k_1, ..., k_{31}\}$, $K^c_1 = \{k_{32}, k_{33}, ..., k_{63}\}$, …, $K^c_{Lenght-1} = \{k_{l-32}, k_{l-31}, ..., k_{l-1}\}$. Then calculated $K_L = K^c_0 \oplus K^c_1 \oplus ... \oplus K^c_{Lenght-1}$. If $K_L = 0$ then as K_L selected 0xC5C31537, i.e. $K_L = 0$xC5C31537. Round keys K^c_i, $i = \overline{Lenght...8n+23}$ calculated as follows: $K^c_i = SBox0(K^c_{i-Lenght}) \oplus SBox1(RotWord32(K^c_{i-Lenght+1}))$ $\oplus K_L$. After each generation of round keys value K_L cyclically shifted left by 1 bit.

Decryption round keys are computed on the basis of encryption round keys and decryption round keys of the first round associate with of encryption round keys as follows:

$$(K^d_0, K^d_1, K^d_2, K^d_3, K^d_4, K^d_5, K^d_6, K^d_7) = ((K^c_{8n})^{-1}, -K^c_{8n+1},$$
$$(K^c_{8n+2})^{-1}, -K^c_{8n+3}, -K^c_{8n+4}, (K^c_{8n+5})^{-1}, -K^c_{8n+6}, -K^c_{8n+7}).$$

Decryption round keys of the second, third and n–round associates with the encryption round keys as follows:

$$(K^d_{8(i-1)}, K^d_{8(i-1)+1}, K^d_{8(i-1)+2}, K^d_{8(i-1)+3}, K^d_{8(i-1)+4}, K^d_{8(i-1)+5}, K^d_{8(i-1)+6},$$
$$K^d_{8(i-1)+7}) = ((K^c_{8(n-i+1)})^{-1}, -K^c_{8(n-i+1)+6}, (K^c_{8(n-i+1)+5})^{-1}, -K^c_{8(n-i+1)+4},$$
$$-K^c_{8(n-i+1)+3}, (K^c_{8(n-i+1)+2})^{-1}, -K^c_{8(n-i+1)+1}, (K^c_{8(n-i+1)+7})^{-1}), i = \overline{2...n}.$$

Decryption keys output transformation associated with the encryption keys as follows:

$$(K^d_{8n}, K^d_{8n+1}, K^d_{8n+2}, K^d_{8n+3}, K^d_{8n+4}, K^d_{8n+5}, K^d_{8n+6}, K^d_{8n+7}) =$$
$$((K^c_0)^{-1}, -K^c_1, (K^c_2)^{-1}, -K^c_3, -K^c_4, (K^c_5)^{-1}, -K^c_6, (K^c_7)^{-1}).$$

Decryption round keys applied to the first round and after the conversion of the output associated with encryption keys as follows: $K^d_{8n+8+j} = K^c_{8n+16+j}$, $K^d_{8n+16+j} = K^c_{8n+8+j}$, $j = \overline{0...7}$.

IV. Results

As a result of this study built a new block encryption algorithms called GOST28147–89–IDEA8–4 and GOST28147–89–RFWKIDEA8–4. This algorithm is based on a networks IDEA16–2 and RFWKIDEA16–2 using the round function of GOST 28147–89. Length of block encryption algorithm is 256 bits, the number of rounds and key lengths is variable. Wherein the user depending on the degree of secrecy of the information and speed of encryption can select the number of rounds and key length.

It is known that S–box of the block encryption algorithm GOST 28147–89 are confidential and are used as long–term keys. In Table 2 below describes the options openly declared S–box such as: deg–degree of the algebraic nonlinearity; NL –nonlinearity; λ –relative resistance to the linear cryptanalysis; δ –relative resistance to differential cryptanalysis; SAC – criterion strict avalanche effect; the BIC criterion of independence of output bits. For S–box was resistant to crypt attack it is necessary that the values deg and NL were large, and the values λ, δ, SAC and BIC small.

Table 2: Parameters of the S–boxes of the GOST 28147–89

№	Parameters	S1	S2	S3	S4	S5	S6	S7	S8
1	deg	2	3	3	2	3	3	2	2
2	NL	4	2	2	2	2	2	2	2
3	λ	0.5	3/4	3/4	3/4	3/4	3/4	3/4	3/4
4	δ	3/8	3/8	3/8	3/8	1/4	3/8	0.5	0.5
5	SAC	2	2	2	4	2	4	2	2
6	BIC	4	2	4	4	4	4	2	4

To S–Box was resistant to cryptanalysis it is necessary that the values deg and NL were large, and the values λ, δ, SAC and BIC small. In block cipher algorithms GOST28147–89–IDEA8–4 and GOST28147–89–RFWKIDEA8–4 for all S–boxes, the following equation: $\deg = 3$, $NL = 4$, $\lambda = 0.5$, $\delta = 3/8$, SAC=4, BIC=4. i.e. resistance is not lower than the algorithm GOST28147–89. These S–boxes are created based on Nyberg construction [3].

References Références Referencias

1. Aripov M., Tuychiev G.. The network IDEA4-2, consists from two round functions // Infocommuni cations: Networks–Technologies–Solutions. 2012, N4 (24), Tashkent", pp.55-59.
2. Aripov M., Tuychiev G.. The network PES8-4, consi-sts from four round functions // Materials of the international scientific conference конференции «Modern problems of applied mathematics and information technologies-Al-Khorezmiy 2012», 2012, Vol.2, Tashkent, pp.16-19.
3. Bakhtiyorov U., Tuychiev G. About Generation Resistance S-Box And Boolean Function On The Basis Of Nyberg Construction // Materials scientific-technical conference «Applied mathematics and information security», Tashkent, 2014, 28–30 april, - pp. 317–324
4. GOST 28147-89. National Standard of the USSR. Information processing systems. Cryptographic protection. Algorithm cryptographic transformation.
5. Tuychiev G.. The networks RFWKIDEA4-2, IDEA4-1 and RFWKIDEA4-1 // Acta of Turin polytechnic university in Tashkent, 2013, N3, Tashkent, pp.71-77.
6. Tuychiev G.N. The network IDEA8–4, consists from four round functions // Infocommunications: Networks–Technologies–Solutions. –Tashkent, 2013, №2 (26), pp. 55–59.
7. Tuychiev G.N. About networks IDEA8–2, IDEA8–1 and RFWKIDEA8–4, RFWKIDEA8–2, RFWKIDEA8–1 developed on the basis of network IDEA8–4 // Uzbek mathematical journal, –Tashkent, 2014, №3, pp. 104–118
8. Tuychiev G.. The network PES4-2, consists from two round functions // Uzbek journal of the problems of informatics and energetics, 2013, N 5-6, Tashkent, pp.107-111.
9. Tuychiev G.. About networks PES4-1 and RFWKP-ES4-2, RFWKPES4-1 developed on the basis of network PES4-2 // Uzbek journal of the problems of informatics and energetics, 2015, N1-2, Tashkent, pp.100-105.
10. Tuychiev G.. About networks RFWKPES8-4, RFWK-PES8-2, RFWKPES8-1, developed on the basis of network PES8-4 // Materials of the international scientific conference «Modern problems of applied mathematics and information technologies-Al-Khor-ezmiy 2014», 2014, vol.2, Samarkand, pp.32-36.
11. Tuychiev G.N. About networks IDEA16–4, IDEA16–2, IDEA16–1, created on the basis of network IDEA16–8 // Compilation of theses and reports republican seminar «Information security in the sphere communication and information. Problems and their solutions» –Tashkent, 2014
12. Tuychiev G.N. About networks RFWKIDEA16–8, RFWKIDEA16–4, RFWKIDEA16–2, RFWKIDEA16–1, created on the basis network IDEA16–8 // Ukrainian Scientific Journal of Information Security, –Kyev, 2014, vol. 20, issue 3, pp. 259–263
13. Tuychiev G.. Creating a data encryption algorithm based on network IDEA4-2, with the use the round function of the encryption algorithm GOST 28147-89 // Infocommunications: Networks-Technologies-Solutions, 2014, N4 (32), Tashkent, pp.49-54.
14. Tuychiev G.. Creating a encryption algorithm based on network RFWKIDEA4-2 with the use the round function of the GOST 28147-89 // International Conference on Emerging Trends in Technology, Science and Upcoming Research in Computer Science (ICDAVIM-2015), //printed in International Journal of Advanced Technology in Engineering and Science, 2015, vol.3, N1, pp.427-432.
15. Tuychiev G.. Creating a encryption algorithm based on network PES4-2 with the use the round function of the GOST 28147-89 // TUIT Bulleten", 2015, N4 (34), Tashkent, pp.132-136.

16. Tuychiev G.. Creating a encryption algorithm based on network RFWKPES4-2 with the use the round function of the GOST 28147-89 // International Journal of Multidisciplinary in Cryptology and Information Security, 2015, N2, vol.4, pp.14-17.
17. Tuychiev G.. The encryption algorithms GOST28147-89-PES8-4 and GOST28147-89-RFWKPES8-4 // «Information Security in the light of the Strategy Kazakhstan-2050»: proceedings III International scientific-practical conference (15-16 October 2015, Astana), 2015, Astana, pp.355-371.
18. Tuychiev G. The Encryption Algorithms GOST-IDEA16-2 and GOST-RFWKIDEA16-2 // Global journal of Computer science and technology: E Network, Web & security, vol 16, Issue 1, pp 30-38.