ISSN 2320 -2610
Volume 4, No.2, March - April 2015
International Journal of Multidisciplinary in Cryptology and Information Security
Available Online at http://www.warse.org/ijmcis/static/pdf/file/ijmcis01422015.pdf

Creating a encryption algorithm based on network RFWKPES4–2 using of the round function encryption algorithm GOST 28147–89

Gulom Tuychiev
National University of Uzbekistan, Uzbekistan, e-mail: blasterjon@gmail.com

ABSTRACT

In this paper create a new block encryption algorithm based on network RFWKPES4–2, with the use the round function of algorithm GOST 28147–89. The block length of created encryption algorithm is 128 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.

1. INTRODUCTION

The encryption algorithm GOST 28147-89 [1] 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.

As the round function network IDEA4-2 [2] using the round function of the encryption algorithm GOST 28147-89 created the encryption algorithm GOST28147-89-IDEA4-2 [8]. In addition, by using transformations SubBytes(), ShiftRows(), MixColumns(), and AddRoundKey() the AES encryption algorithm as round functions of networks IDEA8-1 [4], RFWKIDEA8-1 [4], PES8-1 [5], RFWKPES8-1 [6], IDEA16-1 [7] created encryption algorithms AES-IDEA8-1 [9], AES-RFWKIDEA8-1 [10], AES-PES8-1 [11], AES-RFWKPES8-1 [12], AES-IDEA16-1 [13]. The network RFWKPES4-2 is given in the article [2] and as the Feistel network, when encryption and decryption using the same algorithm. In the network RFWKPES4-2 was used two round functions and as round functions, may be used any conversion.

In this article, applying the round function of the encryption algorithm GOST 28147-89 as round functions of the network RFWKPES4-2, developed encryption algorithm GOST28147-89-RFWKPES4-2, which has the advantage of speed encryption and resistance. In the proposed encryption algorithm GOST28147-89-RFWKPES4-2 block length is 128 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 will be listed the structure of the proposed encryption algorithm.

2. THE ENCRYPTION ALGORITHM GOST28147-89-RFWKPES4-2.

2.1 The structure of the encryption algorithm GOST28147-89-RFWKPES4-2.

In the encryption algorithm GOST28147-89-RFWKPES4-2 length of the subblocks X^0, X^1, X^2, X^3, the length of the round keys $K_{4(i-1)}$, $K_{4(i-1)+1}$, $K_{4(i-1)+2}$, $K_{4(i-1)+3}$, $i=\overline{1...n+1}$, K_{4n+4}, K_{4n+5}, ..., K_{4n+11}, as well as the length of the input and output units round function is equal to 32 bits. In this algorithm the encryption round function of GOST 28147-89 is used twice and in each round functions used eight S-box, i.e. the total number of S-box is 16. The structure of the encryption algorithm GOST28147-89-RFWKPES4-2 is shown in Figure 1.

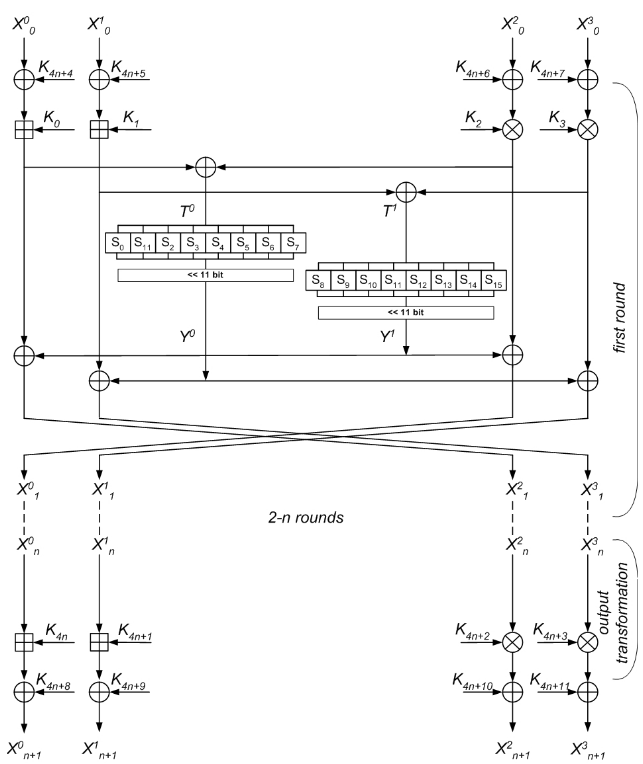

Figure 1: The scheme *n*–rounded encryption algorithm GOST28147–89–RFWKPES4–2

Consider the round function of a encryption algorithm GOST28147-89-RFWKPES4-2. First 32-bit subblocks T^0, T^1 divided into eight four-bit sub-blocks, i.e. $T^0 = t_0^0 \| t_1^0 \| t_2^0 \| t_3^0 \| t_4^0 \| t_5^0 \| t_6^0 \| t_7^0$, $T^1 = t_0^1 \| t_1^1 \| t_2^1 \| t_3^1 \| t_4^1 \| t_5^1 \| t_6^1 \| t_7^1$. The four-bit subblocks t_i^0, t_i^1, $i = \overline{0...7}$ converted to S-box: $R^0 = S_0(t_0^0) \| S_1(t_1^0) \| S_2(t_2^0) \| S_3(t_3^0) \| S_4(t_4^0) \| S_5(t_5^0) \| S_6(t_6^0) \| S_7(t_7^0)$, $R^1 = S_8(t_0^1) \| S_9(t_1^1) \| S_{10}(t_2^1) \| S_{11}(t_3^1) \| S_{12}(t_4^1) \| S_{13}(t_5^1) \| S_{14}(t_6^1) \| S_{15}(t_7^1)$. Received 32-bit subblocks R^0, R^1 cyclically shifted to the left by 11 bits and get the subblocks Y^0, Y^1: $Y^0 = R^0 <<11$, $Y^1 = R^1 <<11$. The S–box of the encryption algorithm are shown in Table 1.

Table 1: The S–box encryption algorithm GOST28147–89–RFWKPES4–2

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
S0	0xA	0x0	0x2	0xF	0x3	0x5	0x4	0x1	0x7	0xE	0x9	0xD	0xC	0xB	0x6	0x8
S1	0x2	0x4	0xC	0x0	0xD	0x6	0x7	0x5	0xE	0x1	0xB	0x8	0x9	0x3	0xF	0xA
S2	0xE	0xC	0xD	0x0	0xA	0x2	0x5	0xB	0x3	0x7	0x8	0x1	0x6	0x9	0x4	0xF
S3	0xA	0xF	0x1	0xB	0x3	0xE	0xC	0xD	0x0	0x9	0x6	0x5	0x7	0x8	0x2	0x4
S4	0xA	0x3	0xD	0xC	0xE	0x5	0x6	0x0	0xB	0xF	0x7	0x2	0x1	0x9	0x8	0x4
S5	0x8	0xD	0x2	0x6	0x1	0x3	0x0	0xE	0xC	0x5	0x4	0x9	0xA	0xB	0xF	0x7
S6	0xD	0x3	0xF	0x6	0x9	0x8	0xE	0x5	0x4	0x0	0x7	0xA	0xC	0xB	0x2	0x1
S7	0xC	0xA	0xB	0xF	0x5	0x9	0x7	0x4	0x8	0x1	0x3	0xE	0x0	0x2	0x6	0xD
S8	0xA	0x3	0x2	0xC	0x0	0x5	0x7	0x1	0x4	0xE	0x9	0xD	0xF	0x8	0x6	0xB
S9	0x0	0x3	0x7	0xA	0xF	0x9	0x1	0xB	0xD	0x2	0xC	0xE	0x6	0x8	0x5	0x4
S10	0xC	0xD	0xB	0x4	0x8	0x5	0x6	0xE	0x3	0x7	0x9	0x2	0x1	0xF	0x0	0xA
S11	0xA	0x0	0xE	0xC	0xF	0x6	0x7	0x1	0x8	0xD	0x5	0x2	0x3	0xB	0x9	0x4
S12	0xA	0xC	0x3	0x7	0x0	0x1	0x2	0xF	0xE	0x4	0x6	0x8	0xB	0x9	0xD	0x5
S13	0x8	0xB	0x5	0x1	0xA	0x2	0xD	0x4	0xC	0xE	0x9	0xF	0x0	0x7	0x3	0x6
S14	0xE	0x8	0x0	0xA	0xF	0xC	0x3	0x7	0x4	0x5	0x9	0x2	0xD	0x1	0xB	0x6
S15	0xC	0xB	0xA	0x9	0x0	0xE	0x4	0x1	0xF	0x3	0x7	0x8	0x2	0x6	0x5	0xD

Consider the process of encryption in the encryption algorithm GOST28147-89-RFWKPES4-2. First 128-bit block of plaintext is divided into 32-bit subblocks X_0^0, X_0^1, X_0^2, X_0^3 and runs the following steps:

1. sublocks X_0^0, X_0^1, X_0^2, X_0^3 summarized by XOR with the corresponding round keys K_{4n+4}, K_{4n+5}, K_{4n+6}, K_{4n+7}: $X_0^i = X_0^i \oplus K_{4n+4+i}$, $j = \overline{0...3}$.
2. sublocks X_0^0, X_0^1, X_0^2, X_0^3 are multiplied and summed accordingly with round keys $K_{4(i-1)}$, $K_{4(i-1)+1}$, $K_{4(i-1)+2}$, $K_{4(i-1)+3}$ и calculates a 32-bit subblocks T^0, T^1. This step can be represented as follows: $T^0 = (X_{i-1}^0 + K_{4(i-1)}) \oplus (X_{i-1}^2 \cdot K_{4(i-1)+2})$, $T^1 = (X_{i-1}^1 + K_{4(i-1)+1}) \oplus (X_{i-1}^3 \cdot K_{4(i-1)+3})$, $i = 1$
3. to the T^0, T^1 sublocks apply the round function and get the 32-bit subblocks Y^0, Y^1.
4. subblocks Y^0, Y^1 are summed by XOR with subblocks X_{i-1}^0, X_{i-1}^1, X_{i-1}^2, X_{i-1}^3, i.e. $X_{i-1}^0 = X_{i-1}^0 \oplus Y^1$, $X_{i-1}^1 = X_{i-1}^1 \oplus Y^0$, $X_{i-1}^2 = X_{i-1}^2 \oplus Y^1$, $X_{i-1}^3 = X_{i-1}^3 \oplus Y^0$, $i = 1$.
5. At the end of the round subblocks swapped, i.e, $X_i^0 = X_{i-1}^2$, $X_i^1 = X_{i-1}^3$, $X_i^2 = X_{i-1}^0$, $X_i^3 = X_{i-1}^1$, $i = 1$.
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^2, X_n^3
7. in output transformation round keys are multiplied and summed into subblocks, i.e. $X_{n+1}^0 = X_n^0 + K_{4n}$, $X_{n+1}^1 = X_n^1 + K_{4n+1}$, $X_{n+1}^2 = X_n^2 \cdot K_{4n+2}$, $X_{n+1}^3 = X_n^3 \cdot K_{4n+3}$.
8. subblocks X_{n+1}^0, X_{n+1}^1, X_{n+1}^2, X_{n+1}^3 are summed by XOR with the round keys K_{4n+8}, K_{4n+9}, K_{4n+10}, K_{4n+11}: $X_{n+1}^j = X_{n+1}^j \oplus K_{4n+8+j}$, $j = \overline{0...3}$. As ciphertext receives the combined 32–bit subblocks $X_{n+1}^0 \| X_{n+1}^1 \| X_{n+1}^2 \| X_{n+1}^3$.

In the encryption algorithm GOST28147-89-RFWKPES4-2 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.

2.2 Key generation of the encryption algorithm GOST28147–89–RFWKPES4–2.

In the *n*-round encryption algorithm GOST28147-89-RFWKPES4-2 used in each round four round keys of 32 bits and the output transformation of four round keys of 32 bits. In addition, prior to the first round and after the output transformation is applied four round keys on 32 bits. The total number of 32-bit round keys is equal to 4*n*+12. Hence, if *n*=8 then need 44 to generate round keys, if *n*=12, you need to generate 60 round keys and if *n*=16 need 76 to generate round keys. When encryption in Fig.1 instead K_i used the round keys K_i^c, and when decryption the round keys K_i^d.

The key length of the encryption algorithm 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, $i = \overline{Lenght...4n+11}$ calculated as follows: $K_i^c = SBox_0 32(K_{i-Lenght}^c) \oplus SBox_1 32(RotWord(K_{i-Lenght+1}^c)) \oplus K_L$. 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, *SBox32()*–convert 32-bit subblock in S-box and $SBox0(A) = S_0(a_0) \| S_1(a_1) \| S_2(a_2) \| S_3(a_3) \| S_4(a_4) \| S_5(a_5) \| S_6(a_6) \| S_7(a_7)$, $SBox1(A) = S_7(a_0) \| S_8(a_1) \| S_9(a_2) \| S_{10}(a_3) \| S_{11}(a_4) \| S_{12}(a_5) \| S_{13}(a_6) \| S_{14}(a_7)$, $A = a_0 \| a_1 \| a_2 \| a_3 \| a_4 \| a_5 \| a_6 \| a_7$ and a_i – the four-bit sub-block.

Decryption round keys K_i^d calculated on the basis of encryption round keys K_i^c and decryption keys output transformation associated with the encryption keys as follows:

$$(K_{4n}^d, K_{4n+1}^d, K_{4n+2}^d, K_{4n+3}^d) = (-K_0^c, -K_1^c, K_2^{c^{-1}}, K_3^{c^{-1}}) \quad (1)$$

Similarly, the decryption keys of the first, second, third and n-round are associated with the keys of the encoding as follows:

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

Decryption round keys applied to the first round and after the conversion of the output associated with encryption keys as follows:

$$K^d_{4n+4+j} = K^c_{4n+8+j}, \; K^d_{4n+8+j} = K^c_{4n+4+j}, \; j = \overline{0...3}.$$

For example, if the number of rounds of encryption algorithm is 16, (1) the formula is as follows:

$$(K^d_{64}, K^d_{65}, K^d_{66}, K^d_{67}) = (-K^c_0, -K^c_1, K^{c^{-1}}_2, K^{c^{-1}}_3)$$

In the same way, according to the formula (2) the round keys for the decryption of the first, second and sixteenth round is calculated as follows:

$$(K^d_0, K^d_1, K^d_2, K^d_3, K^d_4, K^d_5) = (-K^c_{64}, -K^c_{65}, (K^c_{66})^{-1}, (K^c_{67})^{-1})$$

$$(K^d_6, K^d_7, K^d_8, K^d_9, K^d_{10}, K^d_{11}) = (-K^c_{60}, -K^c_{61}, (K^c_{62})^{-1}, (K^c_{63})^{-1})$$

$$(K^d_{60}, K^d_{61}, K^d_{62}, K^d_{63}) = (-K^c_4, -K^c_5, (K^c_6)^{-1}, (K^c_7)^{-1})$$

Similarly, the round keys are calculated cipher upon number of rounds equal to 8 and 12.

3. RESULTS

As a result of the present research constructed a new block encryption algorithm called GOST28147-89-RFWKPES4-2. This algorithm is built on the basis of the network RFWKPES4-2 using the round function of GOST 28147-89. The block length is 128 bits, the number of rounds and key length are variable. This user depending on the degree of secrecy of information and speed of encryption can choose 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

№	Parameter	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

In block encryption algorithm GOST28147-89-RFWKPES4-2 for all S–box the following equality: $\deg = 3$, $NL = 4$, $\lambda = 0.5$, $\delta = 3/8$, SAC=4, BIC=4 i.e. resistance not lower than algorithm GOST 28147-89.

Studies shows that the speed of the encryption algorithm block cipher GOST28147-89-RFWKPES4-2 faster than GOST 28147-89. Created on 16-round algorithm 1.25 times faster than 32 round algorithm GOST 28147-89.

So, we have constructed a new block encryption algorithm called GOST28147-89-RFWKPES4-2 network-based RFWKPES4-2 using the round function of GOST 28147-89. Installed that the resistance offered by the author of the block encryption algorithm is not lower than the resistance of the algorithm GOST 28147-89.

REFERENCES

1. GOST 28147–89. National Standard of the USSR. Information processing systems. Cryptographic protection. Algorithm cryptographic transformation.
2. Aripov M.M. Tuychiev G.N. The network IDEA4–2, consists from two round functions // Infocommunications: Networks–Technologies–Solutions. –Tashkent, 2012, №4 (24), pp. 55–59.
3. 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, №1, pp. 97–103.
4. 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
5. Tuychiev G.N. About networks PES8–2 and PES8–1, developed on the basis of network PES8–4 // Transactions of the international scientific conference «Modern problems of applied mathematics and information technologies–Al–Khorezmiy 2012», Volume № II, –Samarkand, 2014, pp. 28–32.
6. Tuychiev G.N. About networks RFWKPES8–4, RFWKPES8–2, RFWKPES8–1, developed on the basis of network PES8–4 // Transactions of the international scientific conference «Modern problems of applied mathematics and information technologies–Al–Khorezmiy 2012», Volume № 2, –Samarkand, 2014, pp. 32–36
7. 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
8. 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. –Tashkent, 2014, №4 (32), pp. 49–54.

9. Tuychiev G. New encryption algorithm based on network IDEA8-1 using of the transformation of the encryption algorithm AES // IPASJ International Journal of Computer Science, 2015, Volume 3, Issue 1, pp. 1-6
10. Tuychiev G. New encryption algorithm based on network RFWKIDEA8-1 using transformation of AES encryption algorithm // International Journal of Computer Networks and Communications Security, 2015, Vol. 3, №. 2, pp. 43–47
11. Tuychiev G. New encryption algorithm based on network PES8-1 using of the transformations of the encryption algorithm AES // International Journal of Multidisciplinary in Cryptology and Information Security, 2015, vol.4., №1, pp. 1-5
12. Tuychiev G. New encryption algorithm based on network RFWKPES8-1 using of the transformations of the encryption algorithm AES // International Journal of Multidisciplinary in Cryptology and Information Security, 2014, vol.3., №6, pp. 31-34
13. Tuychiev G. New encryption algorithm based on network IDEA16-1 using of the transformation of the encryption algorithm AES // IPASJ International Journal of Information Technology, 2015, Volume 3, Issue 1, pp. 6-12