International Journal of Advanced Technology in Engineering and Science www.ijates.com
Volume No 03, Special Issue No. 01, April 2015 ISSN (online): 2348 – 7550

CREATING A ENCRYPTION ALGORITHM BASED ON NETWORK RFWKIDEA4-2 WITH THE USE THE ROUND FUNCTION OF THE GOST 28147-89

Tuychiev G.

National University of Uzbekistan, Republic of Uzbekistan, Tashkent

ABSTRACT

In this article we create a block encryption algorithm based on network RFWKIDEA4-2, with the use the round function of 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:* *Feystel Network, Network, Round Function, Round, Round Keys, Output Transformation, Subblock, S-Box

I. INTRODUCTİON

GOST 28147-89 is a standard encryption algorithm of the Russian Federation. It is based on the basis of a Feistel network. This encryption algorithm for hardware and software implementation, satisfies the necessary requirements for cryptographic resistance and therefore imposes no restrictions on the degree of secrecy protected information. Implements the encryption algorithm 64-bit blocks of data via a 256-bit key. In the round function employed eight S-boxes a size 4x4 and cyclic shift operation on 11 bits. So far, 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 by 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 RFWKIDEA4–2 presented in the paper [1] and as in a Feistel network, with encryption and decryption using the same algorithm. In a network RFWKIDEA4–2 applied two round functions and as the round function, you can use any transformation.

In this article, we applied round function encryption algorithm GOST 28147-89 as a round function network RFWKIDEA4-2, designed encryption algorithm GOST28147-89- RFWKIDEA4-2, which has the advantage of speed and resistance of encryption. The proposed encryption algorithm GOST28147-89-RFWKIDEA4-2 block length is 128 bits, key length is changed from 256 bits to 1024 bits in steps 128 bits and the number of rounds *n* is 8, 12, 16, that allows the user depending on the degree of secrecy of information and the speed select the number of rounds of encryption and key length. Below is the structure of the proposed encryption algorithm.

II. THE STRUCTURE OF THE ENCRYPTION ALGORITHM GOST28147–89–RFWKIDEA4–2

In the encryption algorithm GOST28147–89–RFWKIDEA4–2 the length of subblocks X^0, X^1, X^2, X^3, length of 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} and 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 equal to 16. The structure of the encryption algorithm GOST28147–89–RFWKIDEA4–2 is shown in Fig.1.

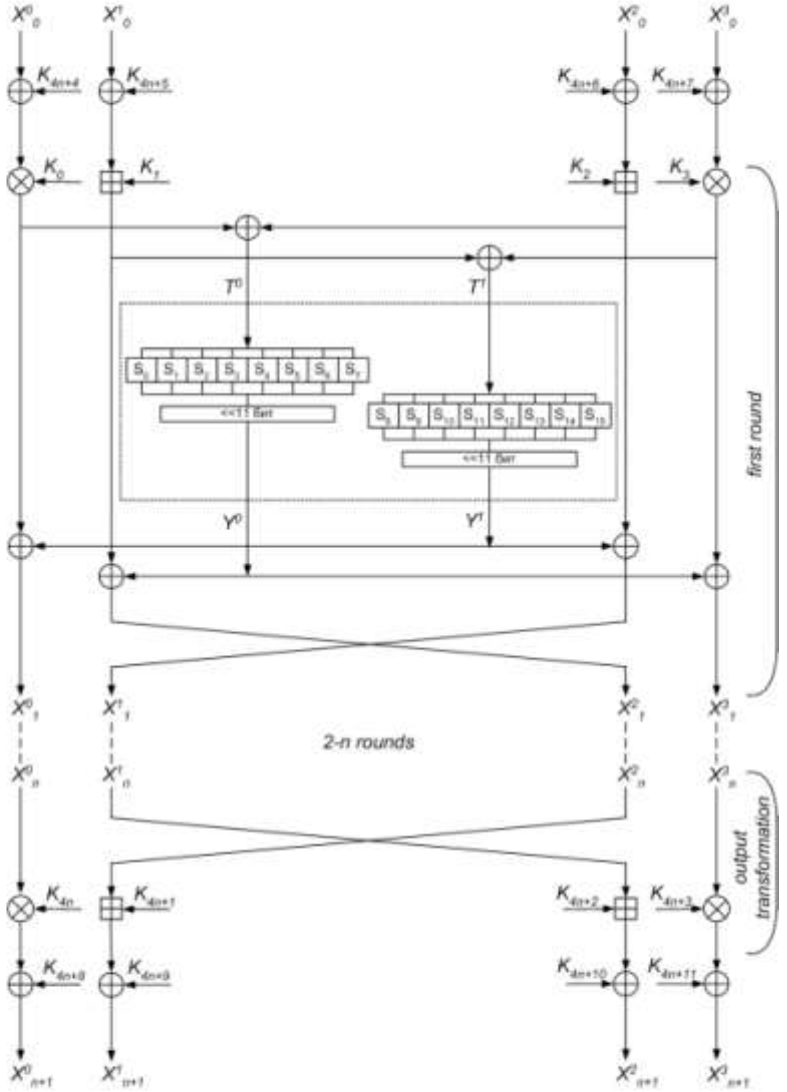

Fig. 1. The Scheme *N*-Rounded Encryption Algorithm GOST28147–89–RFWKIDEA4–2

Consider the round function block encryption algorithm GOST28147–89–RFWKIDEA4–2. First to the 32-bit subblock 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, $Y^1: Y^0 = R^0 \ll 11$, $Y^1 = R^1 \ll 11$. The S-box encryption algorithm shown in Table 1.

Table 1. The S-box of Encryption Algorithm GOST28147–89–RFWKIDEA4–2

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

Consider the encryption process of encryption algorithm GOST28147–89–RFWKIDEA4–2. Initially the 128-bit plaintext X partitioned into subblocks of 32 bits X_0^0, X_0^1, X_0^2, X_0^3, and performs the following steps:

1. subblocks X_0^0, X_0^1, X_0^2, X_0^3 are summed to XOR with corresponding round keys K_{4n+4}, K_{4n+5}, K_{4n+6}, K_{4n+7}: $X_0^i = X_0^i \oplus K_{4n+4+i}$, $j = \overline{0 \ldots 3}$.

2. subblocks X_0^0, X_0^1, X_0^2, X_0^3 respectively, multiplied and summed with the round keys $K_{4(i-1)}$, $K_{4(i-1)+1}$, $K_{4(i-1)+2}$, $K_{4(i-1)+3}$ and calculated 32-bit subblocks T^0, T^1. This step can be represented as follows: $T^0 = (X_{i-1}^0 \cdot K_{4(i-1)}) \oplus (X_{i-1}^2 + K_{4(i-1)+2})$, $T^1 = (X_{i-1}^1 + K_{4(i-1)+1}) \oplus (X_{i-1}^3 \cdot K_{4(i-1)+3})$, $i = 1$.

3. to 32-bit subblocks 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 to 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, $X_i^0 = X_{i-1}^0$, $X_i^1 = X_{i-1}^2$, $X_i^2 = X_{i-1}^1$, $X_i^3 = X_{i-1}^3$, $i = 1$.

6. repeating the steps 2–5 n time, i.e. $i = \overline{2 \ldots n}$, 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. $X_{n+1}^0 = X_n^0 \cdot K_{4n}$, $X_{n+1}^1 = X_n^2 + K_{4n+1}$, $X_{n+1}^2 = X_n^1 + K_{4n+2}$, $X_{n+1}^3 = X_n^3 \cdot K_{4n+3}$.

8. subblocks X_n^0, X_n^1, X_n^2, X_n^3 are summed to XOR with the round keys K_{4n+8}, K_{4n+9}, K_{4n+10}, K_{4n+11}: $X_{n+1}^j = X_{n+1}^j \oplus K_{4n+8+j}$, $j = \overline{0 \ldots 3}$. 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–RFWKIDEA4–2 with encryption and decryption is used the same algorithm, only when decryption calculated inversion round keys depending on the operations and are applied in reverse order. One important task of encryption is key generation.

III. KEY GENERATION ENCRYPTION ALGORITHM GOST28147–89–RFWKIDEA4–2

In the *n*-round encryption algorithm GOST28147–89–RFWKIDEA4–2 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 encoding in Fig.1 instead K_i used the round keys K_i^c, and when decrypting 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, $K_L = \text{0xC5C31537}$. Round keys K_i^c, $i = \overline{Lenght ... 4n+11}$ calculated as follows: $K_i^c = SBox0(K_{i-Lenght}^c) \oplus SBox1(RotWord(K_{i-Lenght+1}^c)) \oplus K_L$. 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, *SubBytes32()*–convert 32-bit subblock 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.

The 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:

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

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

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

Likewise, decryption round keys second, third and *n*-round associated with the encryption round keys as follows:

$$(K_{4(i-1)}^d, K_{4(i-1)+1}^d, K_{4(i-1)+2}^d, K_{4(i-1)+3}^d) = ((K_{4(n-i+1)}^c)^{-1}, -K_{4(n-i+1)+2}^c, -K_{4(n-i+1)+1}^c, (K_{4(n-i+1)+3}^c)^{-1}),\ i = \overline{2...n}. \quad 3$$

Decryption round keys, applied to the first round and after output transformation connected with encryption keys as follows: $K_{4n+4+j}^d = K_{4n+8+j}^c$, $K_{4n+8+j}^d = K_{4n+4+j}^c$, $j = \overline{0...3}$.

For example, if the number of rounds encryption algorithm is 16, then (1) and (2) formula as follows:

$$(K_0^d, K_1^d, K_2^d, K_3^d, K_4^d, K_5^d) = (K_{96}^{c\ -1}, -K_{97}^c, -K_{98}^c, K_{99}^{c\ -1}, K_{94}^c, K_{95}^c)$$

$$(K_{96}^d, K_{97}^d, K_{98}^d, K_{99}^d) = (K_0^{c\ -1}, -K_1^c, -K_2^c, K_3^{c\ -1})$$

Likewise, by formula (3) decryption round key second, third and sixteenth round calculated as follows:

$$(K_6^d, K_7^d, K_8^d, K_9^d, K_{10}^d, K_{11}^d) = (K_{90}^{c^{-1}}, -K_{92}^c, -K_{91}^c, K_{93}^{c^{-1}}, K_{88}^c, K_{89}^c),$$

$$(K_{12}^d, K_{13}^d, K_{14}^d, K_{15}^d, K_{16}^d, K_{17}^d) = (K_{84}^{c^{-1}}, -K_{86}^c, -K_{85}^c, K_{87}^{c^{-1}}, K_{82}^c, K_{83}^c),$$

$$(K_{90}^d, K_{91}^d, K_{92}^d, K_{93}^d, K_{94}^d, K_{95}^d) = (K_6^{c^{-1}}, -K_8^c, -K_7^c, K_9^{c^{-1}}, K_4^c, K_5^c).$$

Likewise, calculated decryption round keys when the number of rounds equal to 8 and 12.

IV. RESULTS

As a result of this study built a new block encryption algorithm called GOST28147–89–RFWKIDEA4–2. This algorithm is based on a network RFWKIDEA4-2 using the round function of GOST 28147-89. Length of block encryption algorithm is 128 bits, the number of rounds and key length 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 the S-box encryption algorithm GOST 28147-89 are secret and used as a long-term key. In Table 2 below describes the options openly declared S-box such as: $\deg$ -degree of algebraic nonlinearity; NL -nonlinearity; λ -resistance to linear cryptanalysis; δ -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 $\deg$ and NL were large, and the values λ, δ, SAC and BIC small.

In block cipher algorithm GOST28147–89–RFWKIDEA4–2 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 GOST 28147-89.

Table2 Parameters of the S-Boxes Algorithm 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

Studies have shown that the speed of the encryption block cipher algorithm GOST28147–89–RFWKIDEA4–2 faster than the GOST 28147-89. Created 16-round algorithm encrypts 1.25 times faster than the 32-round GOST 28147-89.

In this way, built a new block encryption algorithm called GOST28147–89–RFWKIDEA4–2 network-based RFWKIDEA4-2 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

[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