



New encryption algorithm based on network PES8-1 using of the transformations of the encryption algorithm AES

Gulom Tuychiev

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

ABSTRACT

In this article developed a new block encryption algorithm based on network PES8-1 using of the transformations of the encryption algorithm AES, which is called AES-PES8-1. The block's length of this encryption algorithm is 256 bits, the number of rounds are 10, 12 and 14. The advantages of the encryption algorithm AES-PES8-1 are that, when encryption and decryption process used the same algorithm. In addition, the encryption algorithm AES-PES8-1 encrypts faster than AES. Unlike encryption algorithm AES-RFWKPES8-1 [5] in the encryption algorithm AES-PES8-1 in the round function is applied round keys and AddRoundKey() transformation.

Keywords: Advanced Encryption Standard, Feistel network, Lai–Massey scheme, round function, round keys, output transformation, multiplication, addition, multiplicative inverse, additive inverse.

1. INTRODUCTION

In September 1997 the National Institute of Standards and Technology (NIST) issued a public call for proposals for a new block cipher to succeed the Data Encryption Standard (DES) [6]. Out of 15 submitted algorithms the Rijndael cipher by Daemen and Rijmen [1] was chosen to become the new Advanced Encryption Standard (AES) in November 2001 [2]. The Advanced Encryption Standard is a block cipher with a fixed block length of 128 bits. It supports three different key lengths: 128 bits, 192 bits, and 256 bits. Encrypting a 128-bit block means transforming it in n rounds into a 128-bit output block. The number of rounds n depends on the key length: $n = 10$ for 128-bit keys, $n = 12$ for 192-bit keys, and $n = 14$ for 256-bit keys. The 16-byte input block $(t_0, t_1, \dots, t_{15})$ which is transformed during encryption is usually written as a 4x4 byte matrix, the called AES *State*.

t_0	t_4	t_8	t_{12}
t_1	t_5	t_9	t_{13}
t_2	t_6	t_{10}	t_{14}
t_3	t_7	t_{11}	t_{15}

The structure of each round of AES can be reduced to four basic transformations occurring to the elements of the *State*.

Each round consists in applying successively to the *State* the SubBytes(), ShiftRows(), MixColumns() and AddRoundKey() transformations. The first round does the same with an extra AddRoundKey() at the beginning whereas the last round excludes the MixColumns() transformation.

The SubBytes() transformation is a nonlinear byte substitution that operates independently on each byte of the *State* using a substitution table (S-box). Figure 1 illustrates the SubBytes() transformation on the *State*.

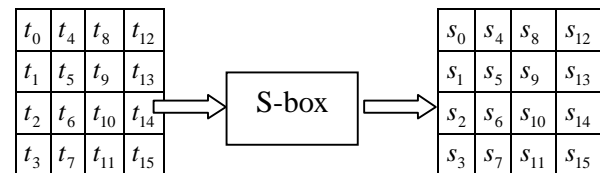


Figure 1: SubBytes() transformation

In the ShiftRows() transformation operates on the rows of the *State*; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. Figure 2 illustrates the ShiftRows() transformation.

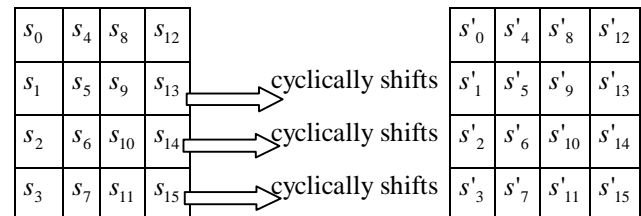


Figure 2: ShiftRows() transformation.

The MixColumns() transformation operates on the *State* column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials over $GF(2^8)$ and multiplied modulo $x^4 + 1$ with a fixed polynomial $a(x)$, given by $a(x) = 3x^2 + x^2 + x + 2$. Let $p = a(x) \otimes s'$:

$$\begin{bmatrix} p_{4i} \\ p_{4i+1} \\ p_{4i+2} \\ p_{4i+3} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s'_{4i} \\ s'_{4i+1} \\ s'_{4i+2} \\ s'_{4i+3} \end{bmatrix}, i = \overline{0..3}$$

As a result of this multiplication, the four bytes in a column are replaced by the following:

$$\begin{aligned}
 y_{4i} &= ((\{02\} \bullet s'_{4i}) \oplus (\{03\} \bullet s'_{4i+1}) \oplus s'_{4i+2} \oplus s'_{4i+3}) \\
 y_{4i+1} &= s'_{4i} \oplus ((\{02\} \bullet s'_{4i+1}) \oplus (\{03\} \bullet s'_{4i+2}) \oplus s'_{4i+3}) \\
 y_{4i+2} &= s'_{4i} \oplus s'_{4i+1} \oplus ((\{02\} \bullet s'_{4i+2}) \oplus (\{03\} \bullet s'_{4i+3}) \\
 y_{4i+4} &= ((\{03\} \bullet s'_{4i}) \oplus s'_{4i+1} \oplus s'_{4i+2} \oplus (\{02\} \bullet s'_{4i+3}) .
 \end{aligned}$$

Figure 3 illustrates the MixColumns() transformation.

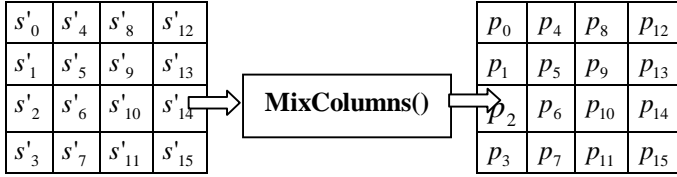


Figure 3: MixColumns() transformation.

In the AddRoundKey() transformation, a round key is added to the State by a simple bitwise XOR operation. Figure 4 illustrates the AddRoundKey() transformation.

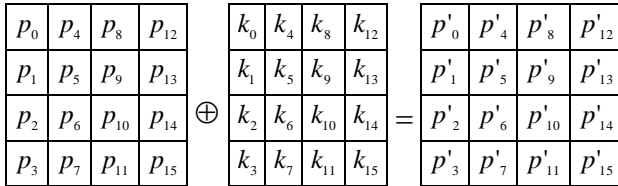


Figure 4. AddRoundKey() transformation

Description networks PES8-1 and RFWKPES8-1 given in [3, 4] and, similarly as in the Feistel network, when it encryption and decryption using the same algorithm. In the network used one round function having four input and output blocks and as the round function can use any transformation. Unlike network RFWKPES8-1 in the network PES8-1 in the round function is applied round keys.

In this paper developed block encryption algorithm AES-PES8-1 based network PES8-1 using transformation of the encryption algorithm AES. The length of block of the encryption algorithm AES-PES8-1 is 256 bits, the number of rounds n equal to 10, 12, 14 and the length of key is variable from 256 bits to 1024 bits in steps 128 bits, i.e. key length is equal to 256, 384, 512, 640, 768, 896 and 1024 bits.

2. THE ENCRYPTION ALGORITHM AES-PES8-1.

2.1 The structure of the encryption algorithm AES-PES8-1.

In the encryption algorithm AES-PES8-1 as the round function used SubBytes(), ShiftRows(), MixColumns(), AddRoundKey() transformation of the encryption algorithm AES. The scheme n -rounded encryption algorithm AES-PES8-1 shown in Figure 5, and the length of subblocks X^0, X^1, \dots, X^7 , length of round keys $K_{9(i-1)}, K_{9(i-1)+1}, \dots,$

$K_{9(i-1)+7}, i = \overline{1..n+1}$ and $K_{9n+8}, K_{9n+9}, \dots, K_{9n+23}$ are equal to 32-bits.

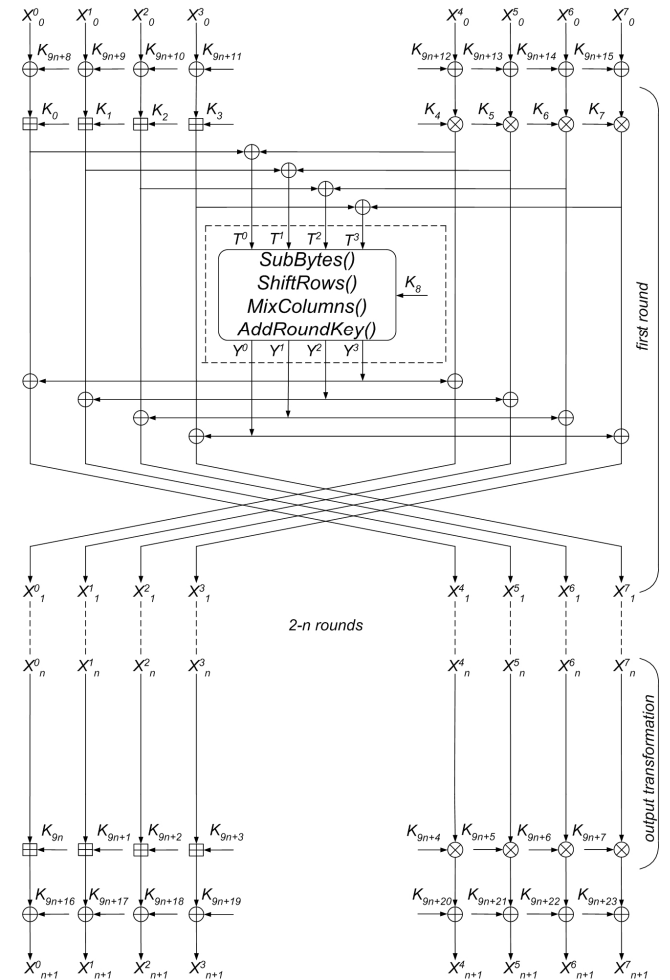


Figure 5. The scheme n -rounded encryption algorithm AES-PES8-1

Consider the round function of the encryption algorithm AES-PES8-1. Initially 32-bit subblocks T^0, T^1, T^2, T^3 , are partitioned into 8-bit subblocks, i.e., on bytes:

$$\begin{aligned}
 t_0 &= sb_0(T^0), t_1 = sb_1(T^0), t_2 = sb_2(T^0), t_3 = sb_3(T^0), \\
 t_4 &= sb_0(T^1), t_5 = sb_1(T^1), t_6 = sb_2(T^1), t_7 = sb_3(T^1), \\
 t_8 &= sb_0(T^2), t_9 = sb_1(T^2), t_{10} = sb_2(T^2), \\
 t_{11} &= sb_3(T^2), t_{12} = sb_0(T^3), t_{13} = sb_1(T^3), \\
 t_{14} &= sb_2(T^3), t_{15} = sb_3(T^3), \text{ here } sb_0(X) = x_0x_1\dots x_7, \\
 sb_1(X) &= x_8x_9\dots x_{15}, sb_2(X) = x_{16}x_{17}\dots x_{23}, \\
 sb_3(X) &= x_{24}x_{25}\dots x_{31} \text{ and } X = x_0x_1\dots x_{31}. \text{ After which the } 8\text{-bit subblocks } t_0, t_1, \dots, t_{15} \text{ are written into the array } State \text{ and are executed the above transformations } SubBytes(), ShiftRows(), MixColumns(), AddRoundKey().
 \end{aligned}$$

In AddRoundKey () transformation as the key k_0, k_1, \dots, k_{15} selected 128-bit round key $K_{9(i-1)+8}$. Initially the key $K_{9(i-1)+8}$ is partitioned into 32-bit keys $K_{9(i-1)+8}^0, K_{9(i-1)+8}^1, \dots, K_{9(i-1)+8}^3$. These keys are partitioned into 8-bit subblocks as follows: $k_0 = sb_0(K_{9(i-1)+8}^0), k_1 = sb_1(K_{9(i-1)+8}^0), k_2 = sb_2(K_{9(i-1)+8}^0), k_3 = sb_3(K_{9(i-1)+8}^0), k_4 = sb_0(K_{9(i-1)+8}^1), k_5 = sb_1(K_{9(i-1)+8}^1), k_6 = sb_2(K_{9(i-1)+8}^1), k_7 = sb_3(K_{9(i-1)+8}^1), k_8 = sb_0(K_{9(i-1)+8}^2), k_9 = sb_1(K_{9(i-1)+8}^2), k_{10} = sb_2(K_{9(i-1)+8}^2), k_{11} = sb_3(K_{9(i-1)+8}^2), k_{12} = sb_0(K_{9(i-1)+8}^3), k_{13} = sb_1(K_{9(i-1)+8}^3), k_{14} = sb_2(K_{9(i-1)+8}^3), k_{15} = sb_3(K_{9(i-1)+8}^3).$

After the AddRoundKey() transformation we obtain 8-bits subblocks $p'_0, p'_1, \dots, p'_{15}$. The resulting 8-bit subblocks are writes on a 32-bit subblocks Y^0, Y^1, Y^2, Y^3 as follows: $Y^0 = p'_0 \parallel p'_1 \parallel p'_2 \parallel p'_3, Y^1 = p'_4 \parallel p'_5 \parallel p'_6 \parallel p'_7, Y^2 = p'_8 \parallel p'_9 \parallel p'_{10} \parallel p'_{11}, Y^3 = p'_{12} \parallel p'_{13} \parallel p'_{14} \parallel p'_{15}.$

The S-box SubBytes() transformation shown in Table 1 and is the only nonlinear transformation. The length of the input and output blocks S-box is eight bits. For example, if the input value the S-box is equal to 0xE7, then the output value is equal 0xCB, ie selected elements of intersection row 0xE and column 0x7.

Table 1: The S-box of encryption algorithm AES-RFWKPES8-1

0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF	
0x0	0x7F	0xFF	0xCE	0x10	0xAD	0x34	0xA1	0xF4	0x69	0x8C	0xEA	0x7C	0xF7	0xF9	0xD9	0x9D
0x1	0x50	0x53	0x2A	0xAC	0xE6	0xAB	0x78	0x84	0x65	0x83	0xC1	0x4C	0x80	0x7D	0x08	0x32
0x2	0x23	0x09	0x24	0xF1	0xD5	0xCD	0x6B	0xEC	0xFC	0x2E	0x66	0xD7	0x72	0x5D	0x3A	0x3D
0x3	0x4A	0xDD	0x37	0x48	0xB0	0xE2	0x19	0x3C	0x30	0xD2	0x7A	0x89	0x91	0xA2	0xE5	0xE7
0x4	0xC7	0x00	0x93	0x99	0xCA	0xF2	0xD1	0xE8	0x9A	0x58	0xAA	0xB6	0x55	0xDA	0xE9	0xEF
0x5	0xC9	0x45	0xDF	0x2D	0x4D	0x5A	0x9F	0x82	0x67	0x03	0x3B	0x71	0xF5	0x49	0xF8	0x96
0x6	0x16	0x04	0x8A	0xB5	0xED	0x1A	0x13	0xBC	0x51	0x9C	0xF6	0x85	0xB3	0xB7	0xFA	0xA3
0x7	0xE0	0x92	0x97	0xDE	0x77	0xD3	0x22	0x42	0x12	0x70	0x74	0x1E	0xFB	0x20	0xFE	0x3F
0x8	0xBF	0x01	0x81	0x7E	0x17	0xB1	0x02	0x14	0xA7	0x07	0xD6	0xE4	0x90	0x3E	0xE3	0x5B
0x9	0x05	0x29	0x33	0xC8	0x64	0x2F	0x5E	0x31	0x2B	0x57	0x87	0x6E	0xE1	0x44	0xEE	0x27
0xA	0xA0	0x40	0x0B	0xBD	0x8F	0x63	0xD8	0x6D	0x1B	0x6C	0x36	0x2C	0x0D	0x98	0x35	0x39
0xB	0x4F	0x79	0x86	0x88	0xF7	0xD0	0x60	0xF0	0xDB	0x0F	0x11	0xC2	0xC3	0x43	0x1F	0x94
0xC	0xAE	0x6A	0x8B	0xBA	0x25	0x54	0x59	0x8E	0xEB	0x9B	0xB4	0x5F	0xA6	0x41	0x4B	0x18
0xD	0x21	0x9E	0x0A	0xB9	0xDC	0xA5	0x38	0x28	0x56	0xAF	0x5C	0xD4	0xC6	0x61	0x76	0xA9
0xE	0xF3	0xCF	0x15	0xA8	0x1D	0x0E	0x8D	0xCB	0x6F	0x0C	0x95	0xB8	0xC5	0x75	0x06	0xB2
0xF	0xA4	0xBB	0x62	0x52	0x68	0x1C	0xBE	0x4E	0xC4	0x46	0xC0	0x7B	0xCC	0x26	0xFD	0x47

Consider the encryption process of encryption algorithm AES-PES8-1. Initially the 256-bit plaintext X partitioned into subblocks of 32 bits $X_0^0, X_0^1, \dots, X_0^7$, and performs the following steps:

- 1) subblocks $X_0^0, X_0^1, \dots, X_0^7$ summed by XOR respectively with round key $K_{9n+8}, K_{9n+9}, \dots, K_{9n+15}$: $X_0^j = X_0^j \oplus K_{9n+8+j}, j = \overline{0..7}.$
- 2) subblocks $X_0^0, X_0^1, \dots, X_0^7$ multiplied and summed respectively with the round key $K_{9(i-1)}, K_{9(i-1)+1}, \dots, K_{9(i-1)+7}$

and calculated 32-bit subblocks T^0, T^1, T^2, T^3 . This step can be represented as follows:

$$T^0 = (X_{i-1}^0 + K_{9(i-1)}) \oplus (X_{i-1}^4 \cdot K_{9(i-1)+4}),$$

$$T^1 = (X_{i-1}^1 + K_{9(i-1)+1}) \oplus (X_{i-1}^5 \cdot K_{9(i-1)+5}),$$

$$T^2 = (X_{i-1}^2 + K_{9(i-1)+2}) \oplus (X_{i-1}^6 \cdot K_{9(i-1)+6}),$$

$$T^3 = (X_{i-1}^3 + K_{9(i-1)+3}) \oplus (X_{i-1}^7 \cdot K_{9(i-1)+7}), i = 1$$

3) subblocks T^0, T^1, T^2, T^3 is split into 8 bit subblocks t_0, t_1, \dots, t_{15} and performed SubBytes(), ShiftRows(), MixColumns(), AddRoundKey() transformations. Output subblocks of the round function of the encryption algorithm are Y^0, Y^1, Y^2, Y^3 .

4) subblocks Y^0, Y^1, Y^2, Y^3 are summed to XOR with subblocks $X_{i-1}^0, X_{i-1}^1, \dots, X_{i-1}^7$, i.e. $X_{i-1}^j = X_{i-1}^j \oplus Y_{3-j}, X_{i-1}^{j+4} = X_{i-1}^{j+4} \oplus Y_{3-j}, j = \overline{0..3}, i = 1.$

5) at the end of the round subblocks are swapped, i.e., $X_i^j = X_{i-1}^{j+4}, X_i^{j+4} = X_{i-1}^j, j = \overline{0..3}, i = 1.$

6) repeating steps 2-5 n times, i.e., $i = \overline{2..n}$ we obtain subblocks $X_n^0, X_n^1, \dots, X_n^7$.

7) in output transformation round keys are multiplied and summed into subblocks, i.e. $X_{n+1}^0 = X_n^0 + K_{9n}, X_{n+1}^1 = X_n^1 + K_{9n+1}, X_{n+1}^2 = X_n^2 + K_{9n+2}, X_{n+1}^3 = X_n^3 + K_{9n+3}, X_{n+1}^4 = X_n^4 \cdot K_{9n+4}, X_{n+1}^5 = X_n^5 \cdot K_{9n+5}, X_{n+1}^6 = X_n^6 \cdot K_{9n+6}, X_{n+1}^7 = X_n^7 \cdot K_{9n+7}.$

8) subblocks $X_{n+1}^0, X_{n+1}^1, \dots, X_{n+1}^7$ are summed to XOR with the round key $K_{9n+16}, K_{9n+17}, \dots, K_{9n+23}$: $X_{n+1}^j = X_{n+1}^j \oplus K_{9n+16+j}, j = \overline{0..7}.$ As cipher text plaintext X receives the combined 32-bit subblocks $X_{n+1}^0 \parallel X_{n+1}^1 \parallel \dots \parallel X_{n+1}^7.$

2.2 Key generation of the encryption algorithm AES-RFWKPES8-1.

In the n -round encryption algorithm AES-PES8-1 in each round we applied eight round keys of the 32 bits and output transformation eight round keys of 32 bits. In addition, before the first round and after the output transformation we used eight round keys of 32 bits. Total number of 32-bit round key is $8n + 24$, and 128-bit round key is equal to n . If 128-bit round key convert to four 32-bit keys, the total number of 32 bit keys equal to $12n + 24$. In Figure 5 encryption used encryption round keys K_i^c instead of K_i , while decryption used decryption round keys K_i^d .

When generating round keys like the AES encryption algorithm uses an array Rcon: Rcon=[0x00000001, 0x00000002, 0x00000004, 0x00000008, 0x00000010, 0x00000020, 0x00000040, 0x00000080, 0x00000100, 0x00000200, 0x00000400, 0x00000800, 0x00001000, 0x00002000, 0x00004000, 0x00008000, 0x00010000, 0x00020000, 0x00040000, 0x00080000, 0x00100000, 0x00200000, 0x00400000, 0x00800000, 0x01000000, 0x02000000, 0x04000000, 0x08000000, 0x10000000, 0x20000000, 0x40000000, 0x80000000].

The key encryption algorithm K of length l ($256 \leq l \leq 1024$) bits is divided into 32-bit round keys $K_0^c, K_1^c, \dots, K_{Lenght-1}^c$, $Lenght = l/32$, here $K = \{k_0, k_1, \dots, k_{l-1}\}$, $K_0^c = \{k_0, k_1, \dots, k_{31}\}$, $K_1^c = \{k_{32}, k_{33}, \dots, k_{63}\}$, \dots , $K_{Lenght-1}^c = \{k_{l-32}, k_{l-31}, \dots, k_{l-1}\}$ and $K = K_0^c \parallel K_1^c \parallel \dots \parallel K_{Lenght-1}^c$. Then we calculate $K_L = K_0^c \oplus K_1^c \oplus \dots \oplus K_{Lenght-1}^c$. If $K_L = 0$ then K_L is chosen as 0xC5C31537, i.e. $K_L = 0xC5C31537$.

When generating a round key K_i^c , $i = Lenght \dots 12n + 23$, we used transformation $SubBytes32()$ and $RotWord32()$, here $SubBytes32()$ -is transformation 32-bit subblock into S-box and $SubBytes32(X) = S(sb_0(X)) \parallel S(sb_1(X)) \parallel S(sb_2(X)) \parallel S(sb_3(X))$, $RotWord32()$ -cyclic shift to the left of 1 bit of the 32 bit subblock. When the condition $i \bmod 3 = 1$ is true, then the round keys are computed as $K_i^c = SubBytes32(K_{i-Lenght+1}^c) \wedge SubBytes32(RotWord32(K_{i-Lenght}^c)) \wedge Rcon[i \bmod 32] \wedge K_L$, otherwise $K_i^c = SubBytes32(K_{i-Lenght}^c) \wedge SubBytes32(K_{i-Lenght+1}^c) \wedge K_L$. After each round key generation the value K_L is cyclic shift to the left by 1 bit.

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

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

Likewise, the decryption round keys of the first, second, third, and n -round associates with the encryption round keys as follows:

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

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

Encryption round key K_i^c associated with $K_i^{c'}$ follows:

$$K_{9i+j}^c = K_{12i+j}^c, j = 0 \dots 7, K_{9i+8}^c = K_{12i+8}^c \parallel K_{12i+9}^c \parallel K_{12i+10}^c \parallel K_{12i+11}^c, K_{9n+j}^c = K_{12n+j}^c, j = 0 \dots 7, K_{9n+8+j}^c = K_{12n+8+j}^c, j = 0 \dots 15.$$

Likewise, decryption round key K_i^d associated with $K_i^{d'}$ follows:

$$K_{9i+j}^d = K_{12i+j}^{d'}, j = 0 \dots 7, K_{9i+8}^d = K_{12i+8}^{d'} \parallel K_{12i+9}^{d'} \parallel K_{12i+10}^{d'} \parallel K_{12i+11}^{d'}, K_{9n+j}^d = K_{12n+j}^{d'}, j = 0 \dots 7, K_{9n+8+j}^d = K_{12n+8+j}^{d'}, j = 0 \dots 15.$$

3. RESULTS

Using $SubBytes()$, $ShiftRows()$, $MixColumns()$, $AddRoundKey()$ transformation of AES encryption algorithm as the round function network PES8-1 is developed block cipher algorithm AES-PES8-1. In the encryption algorithm AES-RFWKPES8-1 [5] as the round function applied $SubBytes()$, $ShiftRows()$, $MixColumns()$ transformation. In the algorithm, the number of rounds of encryption and key length is variable and the user can select the number of rounds and the key length in dependence of the degree of secrecy of information and speed encryption.

As in the encryption algorithms based on the Feistel network, the advantages of the encryption algorithm AES-PES8-1 are that, when encryption and decryption using the same algorithm and as a round function can be any transformation, including one-way function. When decryption the encryption algorithm AES-PES8-1 encryption round keys are used in reverse order, thus on the basis of operations necessary to compute the inverse. For example, if the round key is multiplied by the sub-block, while decryption is necessary to calculate the multiplicative inverse, if summarized, it is necessary to calculate the additive inverse.

It is known that the resistance of AES encryption algorithm is closely associated with resistance S-box, applied in the algorithm. In the S-box encryption algorithm AES algebraic degree of nonlinearity $deg = 7$, nonlinearity $NL = 112$, resistance to linear cryptanalysis $\lambda = 32/256$, resistance to differential cryptanalysis $\delta = 4/256$, strict avalanche criterion $SAC = 8$, bit independence criterion $BIC = 8$.

In the encryption algorithm AES-PES8-1 resistance S-box is equal to resistance S-box encryption algorithm AES, i.e., $\text{deg} = 7$, $NL = 112$, $\lambda = 32/256$, $\delta = 4/256$, $\text{SAC}=\text{BIC}=8$.

Research indicates that the speed of the encryption algorithm AES-PES8-1 higher than AES. 14 rounder encryption algorithm AES-PES8-1 1.17 times faster than a 14 rounder encrypts AES.

4. CONCLUSIONS

It is known that as a network-based algorithms Feystel the resistance algorithm based on network PES8-1 closely associated with resistance round function. Therefore, selecting the transformations SubBytes(), ShiftRows(), MixColumns(), AddRoundKey() of the encryption algorithm AES, based on round function network PES8-1 we developed relatively resistant encryption algorithm.

REFERENCES

1. Daeman J., Rijmen V. **AES proposal: Rijndael, version 2, 1999.** <http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf>
2. **National Institute of Standards and Technology. Announcing the Advanced Encryption Standard (AES), 2001.** *Federal Information Processing Standards Publication 197*, <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.
3. Tuychiev G. **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, – Tashkent, 2014, pp. 32–36
4. Tuychiev G. **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, – Tashkent, 2014, pp. 28–32.
5. 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
6. **U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology. Data Encryption Standard (DES), 1979.** *Federal Information Processing Standards Publication 46-3*, <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>.