New Encryption Algorithm Based on Network RFWKIDEA8-1 Using Transformation of AES Encryption Algorithm

GULOM TUYCHIEV

National University of Uzbekistan, Republic of Uzbekistan, Tashkent

E-mail: blasterjon@gmail.com

ABSTRACT

In this article we developed a new block encryption algorithm based on network RFWKIDEA8-1 using of the transformations of the encryption algorithm AES, which is called AES-RFWKIDEA8-1. The block’s length of this encryption algorithm is 256 bits; the numbers of rounds are 10, 12 and 14. The advantages of the encryption algorithm AES-RFWKIDEA8-1 are that, when encryption and decryption process used the same algorithm. In addition, the AES-RFWKIDEA8-1 encryption algorithm encrypts faster than AES.

Keywords: Advanced Encryption Standard, Feystel Network, Lai–Massey Scheme, Round Function, Round keys, Output Transformation, 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) [4]. 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 \( (t_0, t_1, \ldots, t_{15}) \) which is transformed during encryption is usually written as a 4x4 byte matrix, the called AES State.

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.

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.
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 $a(x) = 3x^3 + x^2 + x + 2$. Let $p = a(x) \otimes s^i$:

\[
\begin{bmatrix}
  p_{4i} \\
  p_{4i+1} \\
  p_{4i+2} \\
  p_{4i+3}
\end{bmatrix} =
\begin{bmatrix}
  0 & 2 & 0 & 3 & 0 & 1 & 0 & 1 \\
  0 & 1 & 2 & 0 & 3 & 0 & 1 & 0 \\
  0 & 1 & 0 & 2 & 0 & 3 & 0 & 1 \\
  0 & 3 & 0 & 1 & 0 & 0 & 2 & 1
\end{bmatrix}
\begin{bmatrix}
  s'_{4i} \\
  s'_{4i+1} \\
  s'_{4i+2} \\
  s'_{4i+3}
\end{bmatrix}, \quad i = 0...3
\]

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

\[
\begin{align*}
  y_{4i} &= (\{02\} \cdot s'_{4i}) \oplus (\{03\} \cdot s'_{4i+1}) \oplus (\{01\} \cdot s'_{4i+2}) \oplus (\{03\} \cdot s'_{4i+3}) \\
  y_{4i+1} &= s'_{4i} \oplus (\{02\} \cdot s'_{4i+1}) \oplus (\{03\} \cdot s'_{4i+2}) \oplus (\{03\} \cdot s'_{4i+3}) \\
  y_{4i+2} &= s'_{4i} \oplus (\{02\} \cdot s'_{4i+1}) \oplus (\{03\} \cdot s'_{4i+2}) \oplus (\{03\} \cdot s'_{4i+3}) \\
  y_{4i+3} &= (\{03\} \cdot s'_{4i}) \oplus s'_{4i+1} \oplus (\{02\} \cdot s'_{4i+2}) \oplus (\{02\} \cdot s'_{4i+3}).
\end{align*}
\]

Figure 3 illustrates the MixColumns() transformation.

2 THE STRUCTURE OF THE ENCRYPTION ALGORITHM AES-RFKIDEA8-1

In the encryption algorithm AES-RFKIDEA8-1 as the round function used SubBytes(), ShiftRows(), MixColumns() transformation of the encryption algorithm AES. The scheme $n$-rounded encryption algorithm AES-RFKIDEA8-1 shown in Figure 4, and the length of subblocks $X^0$, $X^1$, ..., $X^7$, length of round keys $K_{0(i-1)}$, $K_{0(i-1)+1}$, ..., $K_{8(i-3)+n+1}$, $i = 1...n+1$ and $K_{8n+8}$, $K_{8n+9}$, ..., $K_{8n+32}$ are equal to 32 bits.
subblocks $T^0$, $T^1$, $T^2$, $T^3$, are partitioned into 8-bit subblocks, i.e., on bytes:

$t_0 = s_b(t_0)$, $t_1 = s_b(t^0)$, $t_2 = s_b(t_0)$, $t_3 = s_b(t^0)$, $t_4 = s_b(t)$, $t_5 = s_b(t^0)$, $t_6 = s_b(t^0)$, $t_7 = s_b(t)$, $t_8 = s_b(t^0)$, $t_9 = s_b(t^0)$, $t_{10} = s_b(t^0)$, $t_{11} = s_b(t^0)$, $t_{12} = s_b(t)$. Here $s_b(X) = x_0x_1...x_7$, $s_b(X) = x_8x_9...x_{15}$, $s_b(X) = x_{16}x_{17}...x_{23}$, $s_b(X) = x_{24}x_{25}...x_{31}$ and $X = x_0x_1...x_{31}$. After which the 8-bit subblocks $t_0$, $t_1$, ..., $t_{15}$ are written into the array State and are executed as the above transformations SubBytes(), ShiftRows(), MixColumns().

After the MixColumns() transformation we obtain 8-bits subblocks $p_0$, $p_1$, ..., $p_{15}$. 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 0xFE, then the output value is equal 0xFF, i.e. selected elements of intersection row 0xEF and column 0x7.

### Table 1: The S-box of encryption algorithm AES-RFJKIDEA8-1

<table>
<thead>
<tr>
<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>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>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>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>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>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>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>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>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>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>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>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>0xF</td>
<td>0x9</td>
<td>0xE</td>
<td>0x6</td>
<td>0x7</td>
<td>0x5</td>
<td>0x4</td>
<td>0x3</td>
<td>0x2</td>
<td>0x1</td>
<td>0x0</td>
<td>0x8</td>
<td>0x9</td>
<td>0x6</td>
<td>0x7</td>
<td>0x5</td>
</tr>
</tbody>
</table>

Consider the encryption process of encryption algorithm AES-RFJKIDEA8-1. Initially the 256-bit plaintext $X$ partitioned into subblocks of 32-bits $X^0_0$, $X^1_0$, ..., $X^7_0$ and performs the following steps:

1) subblocks $X^0_0$, $X^1_0$, ..., $X^7_0$ summed by XOR with round key $K_{h(0,i)}$, $K_{h(0,i)} = X^j_0 \oplus K_{h(0,j)}$, $i = 0...7$

2) subblocks $X^0_0$, $X^1_0$, ..., $X^7_0$ multiplied and summed respectively with the round key $K_{h(1,i)}$, $K_{h(1,i)} = K_{h(1,i)} \oplus K_{h(1,j)}$, $i = 0...7$ calculated 32-bit subblocks $T^0$, $T^1$, $T^2$, $T^3$. This step can be represented as follows:

$T^0 = (X^0_{i=1} \oplus K_{h(1,i)}) \oplus (X^5_{i=1} \oplus K_{h(1,i)})$, $T^1 = (X^1_{i=1} \oplus K_{h(1,i)}) \oplus (X^5_{i=1} \oplus K_{h(1,i)})$, $T^2 = (X^2_{i=1} \oplus K_{h(1,i)}) \oplus (X^5_{i=1} \oplus K_{h(1,i)})$, $T^3 = (X^3_{i=1} \oplus K_{h(1,i)}) \oplus (X^5_{i=1} \oplus K_{h(1,i)})$, $i = 1$

3) subblocks $T^0$, $T^1$, $T^2$, $T^3$ is split into 8-bit subblocks $t_0$, $t_1$, ..., $t_{15}$ 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 $X^0_0$, $X^1_0$, ..., $X^7_0$, i.e. $X^0_{i=1} = X^0_{i=1} \oplus Y_{i=1}$, $X^4_{i=1} = X^4_{i=1} \oplus Y_{i=1}$, $j = 0...3$.

5) at the end of the round subblocks $X^0_0$ and $X^4_0$, $X^2_0$ and $X^5_0$, $X^3_0$ and $X^4_0$ swapped, $X^0_0$ and $X^3_0$ does not change,
6) repeating steps 2-5  \(n\) times, i.e.,  \(i = \frac{2}{\sqrt{n}}\) we obtain subblocks \(X_0, X_1, \ldots, X_{63}\).

7) in output transformation round keys are multiplied and summed into subblocks, i.e.
\[
\begin{align*}
X_{n+1} &= X_0 + K_n, \\
X_{n+2} &= X_1 + K_{n+1}, \\
X_{n+3} &= X_2 + K_{n+2}, \\
X_{n+4} &= X_3 + K_{n+3}, \\
X_{n+5} &= X_4 + K_{n+4}, \\
X_{n+6} &= X_5 + K_{n+5}.
\end{align*}
\]

8) subblocks \(X_0, X_1, \ldots, X_{63}\) are summed to XOR with the round key \(K_{n+6}, K_{n+7}, \ldots\), \(K_{n+23} : X_{n+1} = X_{n+1} \oplus K_{n+23}\), \(j = 0, \ldots, 7\). As cipher text plaintext \(X\) receives the combined 32-bit subblocks \(X_0 \| X_1 \| \ldots \| X_{63}\).

3 KEY GENERATION OF THE ENCRYPTION ALGORITHM AES-RFWKIDEA8-1

In the 8-round encryption algorithm AES-RFWAREAL8-1 in each round applied of eight round keys to 32 bits and output conversion eight round keys of 32 bits. In addition, before the first round and after the output transformations is used eight round keys of 32 bits. Total number of 32-bit round key is equal to \(8n + 24\). When encoding in Figure 4 is used instead of \(K\) encryption round keys \(K_i\), while decryption round decryption key \(K_i\).

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

The key encryption algorithm \(K\) of length \(l = 256\) is divided into 32-bit round keys \(K_0, K_1, \ldots, K_{l-1}\), \(l = 1 / 32\), here
\[
K = \{k_0, k_1, \ldots, k_{l-1}\}, \\
K' = \{k_0, k_1, \ldots, k_{l-1}\}, \\
K'' = \{k_0, k_1, \ldots, k_{l-1}\}, \\
K''' = \{k_0, k_1, \ldots, k_{l-1}\}.
\]
and \(K = K_0 \| K_1 \| \ldots \| K_{l-1}\). After which calculated \(K_j = K_j \oplus K_j \oplus \ldots \oplus K_{j+63}\). If \(K = 0\) then \(K\) is chosen as \(0xC5C31537\), i.e. \(K = 0xC5C31537\). When generating a round key \(K_j\), \(i = \text{Lenght} \cdot 8n + 23\), we used transformation SubBytes() and RotWord() here.

For \(K_0\text{SubBytes}(X), \text{RotWord}(X)\text{SubBytes}(X)\text{SubBytes}(X)\) is transformation 32-bit subblock into S-box and SubBytes(X) = \(S(sb(X)) \| S(sb(X))\).

4 RESULTS

Using the transformations SubBytes(), ShiftRows(), MixColumns() of the encryption algorithm AES as the round transformation network.
RFWKIDEA8-1 we developed block cipher algorithm AES-RFWKPES8-1. In the algorithm, the number of rounds of encryption and key’s length is variable and the user can select the number of rounds and the key’s 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 RFWKIDEA8-1 are that, when encryption and decryption process used the same algorithm. In the encryption algorithm RFWKIDEA8-1 in decryption process 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 subblock, while decryption is 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’s 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-RFWKIDEA8-1 resistance S-box is equal to resistance S-box’s encryption algorithm AES, i.e., $\deg = 7$, $NL = 112$, $\lambda = 32/256$, $\delta = 4/256$, $SAC = BIC = 8$.

Research indicates that the speed of the encryption algorithm AES-RFWKIDEA8-1 is faster than AES. The encryption speed of the 14 rounds encryption algorithm AES-RFWKIDEA8-1 1.25 times faster than the 14 rounds encryption algorithm AES.

5 CONCLUSIONS

It is known that as a network-based algorithms Feistel the resistance algorithm based on network RFWKIDEA8-1 closely associated with resistance round function. Therefore, selecting the transformations SubBytes(), ShiftRows(), MixColumns() of the encryption algorithm AES, based on round function network RFWKIDEA8-1 we developed relatively resistant encryption algorithm.

7 REFERENCES