A Modified Feistel Cipher involving a key as a multiplicant on both the sides of the Plaintext matrix and supplemented with Mixing Permutation and XOR Operation

1V.U.K. Sastry, 2K. Anup Kumar
1Director School of Computer Science and Informatics, Dean(R & D), Department of Computer Science and Engineering, Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad, 501301, Andhra Pradesh, India.
2Associate Professor, Department of Computer Science and Engineering, Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad, 501301, Andhra Pradesh, India.

1vuksastry@rediffmail.com and 2k_anupkumar@yahoo.com

Abstract: In this paper, we have developed a block cipher by offering a modification to the classical Feistel cipher. Unlike in the case of the classical Feistel cipher wherein we have a binary string as a plaintext, here we have taken the plaintext as a matrix, which is divided into a pair of matrices. One of these matrices is multiplied with the key matrix on both the sides. The process of encryption is supplemented with a pair of functions called Mix ( ) and Permute ( ). In addition to these two, we have used XOR operation. The avalanche effect and the cryptanalysis indicate that cipher is a strong one.

Key words: Encryption, Decryption, Key matrix, Mix, Permute and XOR.

1. Introduction

In the development of symmetric block ciphers, Feistel cipher [1] plays a prominent role. The basic features of the block cipher are confusion and diffusion. In diffusion, the structure of the plaintext is thoroughly dissipated before it becomes the cipher text. In confusion, the relationship between the cipher text and the key are made as complex as possible, so that the key can never be found by any means.

In the classical Feistel cipher [2], the string of the plaintext, which is of length 64 binary bits, is divided into two equal halves. In the iteration process, operations are carried out on the left and right halves by including a key, involving a XOR operation and interchanging the left half and the right half as we proceed through various steps of the iteration process. The relations describing the encryption of the cipher can be written in the form

\[ P_i = Q_{i-1}, \]

\[ Q_i = P_{i-1} \oplus F(Q_{i-1}, K_i), \]

Where, \( P_i \) and \( Q_i \) are the binary strings which correspond to the left and right halves of the plaintext (in the \( i \) th round of the iteration) respectively. Here, \( K_i \) is the key in the \( i \) th round, and ‘F’ is a function of \( Q_{i-1} \) and \( K_i \). The relations governing the decryption can readily be written in the form

\[ Q_{i-1} = P_i, \]

\[ P_{i-1} = Q_i \oplus F(P_i, K_i). \]

In the literature of Cryptography, it is clearly seen that most of the popular ciphers [2-5] are developed by using the basic principles of the Feistel cipher.

In a recent investigation, Sastry et al. [6-7] have developed a pair of variants of Feistel cipher. In these ciphers they have taken \( P_i \) and \( Q_i \) (the left half and the right half) as matrices instead of binary strings. In the first one, they have developed a modified Feistel cipher involving modular arithmetic and a key on both the sides of the plaintext matrix. They have included the XOR operation in this analysis. On the other hand, in the
second one, they have introduced modular arithmetic addition.

In the present analysis, our objective is to reconsider the modified Feistel cipher involving XOR operation and to supplement it with the functions namely mixing and permutation so that the strength of the cipher increases significantly.

In what follows, we mention the plan of the paper. In section 2, we introduce the development of the cipher, and present the flowcharts and the algorithms describing the cipher. Section 3 deals with the illustration of the cipher, and exhibits the avalanche effect. We have discussed the cryptanalysis in section 4. Finally, we have mentioned the details of the computations and drawn conclusions in section 5.

2. Development of the cipher

Let us consider a plaintext \( P \) containing \( 2m^2 \) characters. On using EBCIDIC code, this can be represented in the form of a pair of square matrices \( P_0 \) and \( Q_0 \) wherein each one is of size ‘m’. Let ‘K’ be the key matrix whose size is \( mxm \). Here it is to be noted that all the elements of \( P_0, Q_0 \) and \( K \) are lying in \([0,255]\).

In this analysis, the process of encryption is governed by the relations

\[
\begin{align*}
P_i &= Q_{i-1}, \\
Q_i &= (P_{i-1} \oplus F(Q_{i-1}, K)) \mod N, \text{ for } i=1 \text{ to } n, (2.1)
\end{align*}
\]

In which

\[F(Q_{i-1}, K) = K Q_{i-1} K\]

These relations are supplemented with the functions (1) Mix ( ) and (2) Permute ( ).

The details of the processes involved in these functions are mentioned later.

The process of decryption can be described by the relations

\[
\begin{align*}
Q_{i-1} &= P_i, \\
P_{i-1} &= (Q_i \oplus F(P_i, K)) \mod N, \text{ for } i = n \text{ to } 1
\end{align*}
\]

Where, \( F(P_i, K) = K P_i K \) (2.2)

Along with these relations, we use the functions IMix ( ) and IPermute ( ), where these functions denote the reverse process of Mix ( ) and Permute ( ) respectively.

In (2.1) and (2.2), ‘N’ is a positive integer which is chosen appropriately. Here we take \( N=256 \).

The flowcharts depicting the encryption and the decryption are given in Fig.1 and Fig.2 respectively.
In the flowchart for encryption (Fig. 1), for the sake of elegance, we have replaced the functions Mix ( ) and Permute ( ) by M ( ) and Φ ( ) respectively. The reverse processes represented by IMix ( ) and IPermute ( ), arising in decryption, are denoted by IM ( ) and Φ ( ).

The algorithms corresponding to the encryption and the decryption are given below.

Algorithm for Encryption

1. Read P, K, n and N
2. $P_{n0} =$ Left half of P.
   $Q_{n0} =$ Right half of P.
3. for i = n to 1 begin
   $P_i = IΦ (P_{i-1})$
   $Q_i = IΦ (Q_{i-1})$
   $P_{i-1} = M (P_i)$
   $Q_{i-1} = M (Q_i)$
   $P_i = F = K (Q_{i-1} K)$
   $Q_i = (P_{i-1} \oplus F) \mod N$
   $P_{i-1} = Φ (P_i)$
   $Q_{i-1} = Φ (Q_i)$
end
4. $C = P_n \parallel Q_n$ /* \parallel represents concatenation */
5. Write(C)

Algorithm for Decryption:

1. Read C, K, n and N.
2. $P_{n0} =$ Left half of C
   $Q_{n0} =$ Right half of C.
3. for i = n to 1 begin
   $P_{i-1} = Q_i \mod N$
   $P_i = IΦ (P_{i-1})$
   $Q_i = IΦ (Q_{i-1})$
   $P_i = M (P_{i-1})$
   $Q_i = M (Q_{i-1})$
   $Q_{i-1} = P_{i-1}$
   $F = K P_{i-1} K$
   $P_{i-1} = (Q_i \oplus F) \mod N$
end
4. $P = P_0 \parallel Q_0$ /* \parallel represents concatenation */
Let us now see the development of the functions (1) Mix ( ) and (2) Permute ( ).

Consider the left half or the right half of the plaintext matrix $P$ as it undergoes the iteration process. Let it be denoted by the square matrix $T$, which is of size ‘m’. This matrix is written in the form

$$T = [T_{ij}], \quad i = 1 \text{ to } m, \quad j = 1 \text{ to } m.$$

On representing each element of the matrix $T$ in its binary form we have

$$T_{111} T_{112} \ldots T_{118} \ldots \ldots \ldots \ldots \ldots T_{1m1} T_{1m2} \ldots T_{1m8}$$

This has $m$ rows and $8m$ columns. On focusing our attention on the first column and taking 8 bits into consideration, we can represent them as a decimal number. This can be placed as the first row first column element of a new matrix. Similarly, we take eight more bits of the first column (if available) and form a decimal number then this can be placed as the first row second column element of the resulting matrix. This process is repeated till we complete all the elements of a column, and all the columns of the above matrix, one after the other, and finally obtain the resulting matrix by placing the elements in a row wise order.

Consider a typical example taking $m=8$. We get one decimal number from the first column. Similarly we have one more from the second column. Finally as we continue the process, we get a square matrix of size 8. This is the process of mixing.

Let us now consider the development of the function Permute ( ). Let $K$ be the key matrix given by

$$K = [ K_{ij} ], \quad i = 1 \text{ to } m, \quad j = 1 \text{ to } m.$$  

This matrix contains $m^2$ decimal numbers. We arrange these numbers in ascending order and reckon the position of each number by looking at the elements of the matrix in a row wise manner. Let the numbers representing the positions be denoted by $L_i, \quad i = 1 \text{ to } m^2$. This representation can be written in the form of a table given below.

<table>
<thead>
<tr>
<th>$L_1$</th>
<th>$L_2$</th>
<th>$L_3$</th>
<th>$L_4$</th>
<th>$\ldots$</th>
<th>$L_i$</th>
<th>$\ldots$</th>
<th>$L_{m^2}$</th>
</tr>
</thead>
</table>

Table 1. Correspondence between the elements of the matrix and their positions.

Consider the plaintext matrix $T$ having $m$ rows and $m$ columns. As we have seen earlier in mixing, $T$ can be written in the form a matrix containing $m$ rows and $8m$ columns. In this matrix, we go on interchanging $i$ th column with $L_i$ th column, till we exhaust all possible such interchanges. Nevertheless, for simplicity, we remember the following rules in carrying out the interchanging process.

1. We do not interchange the columns if $i$ is equal to $L_i$.
2. If an interchange is already made between two columns, by considering the corresponding numbers, no such change will be carried out further.

After carrying out all these interchanges, we can convert the resulting plaintext matrix into its decimal form. Thus we get back an $m \times m$ matrix. This is the process of permutation.

In order to have a clear insight into the permutation process let us consider a simple example.

Let

$$K = \begin{bmatrix} 64 & 196 & 13 & 73 \\ 56 & 48 & 102 & 32 \\ 02 & 94 & 83 & 65 \\ 36 & 67 & 87 & 250 \end{bmatrix}$$ (2.1)

On arranging the numbers in the key in ascending order, we have

$$02 , 13 , 32 , 36 , 48 , 56 , 64 , 65 , 73 , 83 , 87 , 94 , 102 , 196 , 250.$$  

These numbers can be written in the form of Table 1 as shown below.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>3</td>
<td>8</td>
<td>13</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>12</td>
<td>14</td>
<td>4</td>
<td>11</td>
<td>15</td>
<td>10</td>
<td>7</td>
<td>2</td>
<td>16</td>
</tr>
</tbody>
</table>
Let us now consider, a plaintext represented in the form a matrix $T$ having 4 rows and 4 columns. This is given by

$$
T = \begin{bmatrix}
72 & 101 & 108 & 108 \\
111 & 32 & 102 & 114 \\
116 & 104 & 101 & 32 \\
112 & 114 & 111 & 98
\end{bmatrix}
$$

(2.2)

On representing each element of (2.2) in its binary form, and writing the binary bits in horizontal manner, we get a matrix $T_b$ containing 4 rows and 32 columns. This is given by

$$
T_b = \begin{bmatrix}
01001000 & 0100101 & 01101000 & 01101000 \\
0110111 & 00100000 & 01100110 & 01100110 \\
01110100 & 01101000 & 01100101 & 00100000 \\
01110000 & 01100110 & 0110111 & 01100010
\end{bmatrix}
$$

(2.3)

We focus our attention on the first sixteen columns of (2.3). On making use of the table, we interchange the columns 1 and 9, 2 and 3, 4 and 13, 5 and 6 etc. till we exhaust all the first 16 columns of $T_b$. The same procedure will be carried out on the next 16 columns. In view of the basic principles 1 and 2 of the permutation process, here, we have not interchanged the columns 3 and 8, 6 and 5, 11 and 11 and 16 and 16. This stipulation is applied in the second part also. Now we convert the binary bits into the decimal numbers by considering bits in a row wise manner. Thus, we get a 4x4 matrix which is obtained in the form

$$
T = \begin{bmatrix}
38 & 229 & 124 & 100 \\
110 & 48 & 107 & 98 \\
120 & 104 & 104 & 96 \\
97 & 106 & 110 & 114
\end{bmatrix}
$$

(2.4)

On the other hand, it is interesting to note that, if $K$ and $T$ are square matrices of size 8 then the matrix obtained by converting $T$ into its binary form (consisting of 8 rows and 64 columns) can be handled as a single piece.

### 3. Illustration of Cipher

Consider the plaintext given below.

Hello friend, Environment is to be protected with utmost care. The problem of floods has been found to be a severe one since several centuries. No one is able to overcome this. The study of environmental science and technology has to be undertaken by all of us with sincerity and devotion. If we can bombard the clouds and drive them out into Bay of Bengal and protect the farmers from the calamity of floods. Make a sincere attempt. 

(3.1)

Let us focus our attention on the first 128 characters of the plaintext. This is given by

Hello friend, Environment is to be protected with utmost care. The problem of floods has become severe since several centuries. No

(3.2)

On applying the EBCDIC code we get,

$$
\begin{bmatrix}
\end{bmatrix}
$$

(3.3)
Illustration of Key Based Random Decomposition and Interlacing during Encryption.

Fig. 01.

(3.3) can be written in the form

\[
P_b = \begin{bmatrix}
72 & 101 & 108 & 111 & 32 & 102 & 114 \\
105 & 114 & 111 & 110 & 109 & 101 & 116 \\
101 & 32 & 112 & 114 & 111 & 116 & 101 & 99 \\
32 & 105 & 115 & 32 & 116 & 111 & 32 & 98 \\
101 & 32 & 112 & 114 & 111 & 116 & 101 & 99 \\
116 & 101 & 100 & 32 & 119 & 110 & 111 & 116 & 104 \\
99 & 97 & 114 & 101 & 46 & 32 & 84 & 104
\end{bmatrix} \quad (3.4) \quad Q_b = \begin{bmatrix}
101 & 32 & 112 & 114 & 111 & 98 & 108 & 101 \\
109 & 32 & 111 & 102 & 32 & 102 & 108 & 111 \\
111 & 100 & 115 & 32 & 104 & 97 & 115 & 32 \\
98 & 101 & 101 & 110 & 32 & 102 & 111 & 117 \\
110 & 100 & 32 & 116 & 111 & 32 & 98 & 101 \\
32 & 97 & 32 & 115 & 101 & 118 & 101 & 114 \\
101 & 32 & 111 & 110 & 101 & 32 & 115 & 105 \\
110 & 99 & 101 & 32 & 115 & 101 & 118 & 101
\end{bmatrix} \quad (3.5)
\]

Let us take the key in the form

\[
K = \begin{bmatrix}
21 & 134 & 52 & 6 & 9 & 22 & 11 & 58 \\
71 & 12 & 3 & 61 & 101 & 199 & 143 & 7 \\
53 & 100 & 4 & 153 & 254 & 91 & 36 & 171 \\
4 & 55 & 181 & 200 & 89 & 112 & 13 & 1 \\
63 & 74 & 111 & 222 & 5 & 238 & 192 & 4 \\
123 & 199 & 255 & 0 & 125 & 77 & 196 & 28 \\
32 & 89 & 35 & 26 & 78 & 201 & 164 & 15 \\
92 & 163 & 194 & 68 & 150 & 230 & 8 & 45
\end{bmatrix} \quad (3.6)
\]

On using (3.4) to (3.6) and the functions Mix( ) and Permute( ), we adopt the encryption algorithm, mentioned in section 2, and obtain the cipher text C given by

\[
C = \begin{bmatrix}
182 & 252 & 139 & 132 & 216 & 93 & 93 & 78 & 44 & 249 & 255 & 254 & 224 & 204 & 185 & 225 \\
92 & 237 & 203 & 183 & 209 & 126 & 218 & 40 & 0 & 59 & 89 & 10 & 2 & 90 & 67 & 19 \\
189 & 87 & 25 & 21 & 88 & 249 & 182 & 202 & 0 & 16 & 81 & 0 & 17 & 89 & 24 & 88 \\
119 & 120 & 248 & 236 & 02 & 91 & 236 & 86 & 88 & 102 & 37 & 23 & 86 & 10 & 118 & 106 \\
175 & 223 & 181 & 38 & 73 & 162 & 91 & 128 & 24 & 64 & 8 & 80 & 88 & 8 & 0 & 88 \\
254 & 139 & 9 & 33 & 129 & 60 & 208 & 215 & 197 & 32 & 160 & 67 & 162 & 224 & 7 & 224 \\
75 & 244 & 207 & 67 & 90 & 64 & 10 & 176 & 86 & 42 & 131 & 26 & 254 & 217 & 70 & 174
\end{bmatrix} \quad (3.7)
\]
On applying the decryption algorithm with the inputs (3.6), (3.7) and the functions IMix ( ) and IPermute ( ), we get back the original plaintext given by (3.2).

Let us now examine the avalanche effect which indicates the strength of the cipher in a qualitative manner. To carry out this one, we change the first character of the plaintext (3.2) from H to I. As the EBCDIC codes of these characters are 72 and 73, we have a change of one bit in the plaintext. On using the modified plaintext and the key (3.6), we apply the encryption algorithm, given in section 2, we get the new cipher text C in the form

$$
\begin{bmatrix}
32 & 117 & 03 & 183 & 245 & 129 & 132 & 197 & 101 & 26 & 222 & 114 & 28 & 9 & 176 & 75 \\
139 & 67 & 165 & 154 & 88 & 137 & 153 & 219 & 216 & 171 & 254 & 20 & 82 & 64 & 35 & 70 \\
11 & 190 & 82 & 189 & 2 & 91 & 200 & 216 & 217 & 100 & 144 & 182 & 197 & 124 & 15 & 3 \\
175 & 53 & 180 & 144 & 73 & 215 & 192 & 15 & 21 & 128 & 12 & 196 & 87 & 59 & 60 & 188 \\
224 & 13 & 19 & 33 & 119 & 19 & 86 & 15 & 93 & 72 & 170 & 87 & 111 & 121 & 212 & 74 \\
15 & 211 & 200 & 156 & 90 & 44 & 67 & 16 & 86 & 92 & 121 & 226 & 57 & 29 & 96 & 205
\end{bmatrix}
$$

Let $$C = (3.8)$$.

On comparing (3.7) and (3.8) in their binary form, we find that they differ by 518 bits (out of 1024 bits). This shows that, the cipher is expected to be a strong one.

Let us now consider a one bit change in the key K given by (3.6). To this end, we change the 5 th row 4 th column element 222 to 223. This leads to a change of one binary bit in the key. Now on using the original plaintext (3.3) and the modified key, as we have obtained here, we apply the encryption algorithm and obtain the cipher text C in the form

$$
\begin{bmatrix}
122 & 52 & 130 & 172 & 211 & 11 & 36 & 128 & 52 & 184 & 250 & 196 & 204 & 51 & 5 & 222 \\
100 & 62 & 03 & 192 & 180 & 172 & 200 & 216 & 250 & 281 & 72 & 86 & 94 & 22 & 221 & 23 \\
27 & 92 & 131 & 169 & 32 & 78 & 101 & 112 & 113 & 114 & 115 & 120 & 156 & 198 & 112 & 36 \\
152 & 169 & 121 & 182 & 29 & 121 & 233 & 208 & 17 & 69 & 84 & 245 & 197 & 6 & 171 & 24 \\
174 & 60 & 53 & 77 & 157 & 154 & 173 & 189 & 156 & 21 & 91 & 82 & 5 & 12 & 173 & 20
\end{bmatrix}
$$

Let $$C = (3.9)$$. 

On comparing (3.7) and (3.9) in their binary form, we find that they differ by 950 bits (out of 1024 bits). This shows that, the cipher is expected to be a strong one.
After converting (3.7) and (3.9) into their binary form and comparing them, we notice that they differ by 511 bits (out of 1024 bits). This result also firmly indicate that the cipher is a potential one.

4. Cryptanalysis

In the literature of Cryptography [1], Some of the methods which are used for breaking ciphers are

1. Cipher text only (Brute Force) attack.
2. Known Plaintext attack.
3. Chosen Plaintext attack.
4. Chosen Cipher text attack.

In all these attacks, we assume that the encryption algorithm and the cipher text are known to the attacker.

Let us now consider the brute force attack. In this analysis, the key is of size 512 bits. Thus the size of the key space is

\[ 2^{512} = (2^{10})^{51.2} \approx (10)^{153.6} \]

If the time required for the execution of the encryption algorithm with one value of the key is \( 10^{-7} \) seconds, then the total time required for execution with all the possible keys in the key space is

\[ 10^{153.6} \times 10^{-7} \approx 10^{153.6} \times 3.12 \times 10^{15} = 3.12 \times 10^{138.6} \text{ years} \]

As this is a staggering amount of time, it is impossible to break the cipher by applying this attack.

We now discuss the known plaintext attack. In this attack, we conventionally know as many pairs of plaintext and ciphertext as we require. In this analysis, in each round of the iteration process, as the plaintext is multiplied by the key on both the sides, and it is subjected to several operations caused by the mod operation, the mix ( ) and the permute ( ) functions, it is not possible for obtaining K or a function of K in terms the other elements such as the plaintext or the ciphertext. In the light of these factors, we conclude that it is not possible to break the cipher by the known plaintext attack.

The last two attacks namely, the chosen plaintext attack and the chosen ciphertext attack are somewhat complex and they are seldom applied in cryptography. We do not find any scope for the application of these attacks in the case of the cipher under consideration.

In the light of the afore mentioned discussion, we conclude that it is not possible to break the cipher by any one of the above attacks.

5. Computations and Conclusions

In this paper, we have devoted our attention to the study of a modified Feistel cipher which includes a key on both the sides of plaintext. In this process, we have made use of XOR operation and a pair of functions called mix ( ) and permute ( ) for thoroughly mixing and permuting the binary bits of the plaintext and the key(in each round of the iteration process) before the result assumes the form of the cipher text.

The programs required for encryption and decryption are written in C Language.

The plaintext (3.1) is divided into 4 blocks. We have 128 characters in the first three blocks and 8 characters in the last block. Thus we append 120 blanks to make it a block of 128 characters. On encrypting the whole plaintext (3.1), we get the following cipher text.
In this cipher, we conclude that, the functions mix and permute play a prominent role in the development of the cipher text, and they strengthen the cipher in various ways. From this analysis we also conclude that, the cipher is a strong one and it can be applied safely for the security of information.

References:


Authors profile:

Dr. V. U. K. Sastry is presently working as Professor in the Dept. of Computer Science and Engineering (CSE), Director (SCSI), Dean (R & D), SreeNidhi Institute of Science and Technology (SNIST), Hyderabad, India. He was Formerly Professor in IIT, Kharagpur, India and worked in IIT, Kharagpur during 1963 – 1998. He guided 12 PhDs, and published more than 40 research papers in various international journals. His research interests are Network Security & Cryptography, Image Processing, Data Mining and Genetic Algorithms.

Mr. K. Anup Kumar is presently working as an Associate Professor in the Department of Computer Science and Engineering, SNIST, Hyderabad India. He obtained his B.Tech (CSE) degree from JNTU Hyderabad and his M.Tech (CSE) from Osmania university, Hyderabad. He is now pursuing his PhD from JNTU, Hyderabad, India, under the supervision of Dr. V.U.K. Sastry in the area of Information Security and Cryptography. He has 10 years of teaching experience and his interest in research area includes, Cryptography, Steganography and Parallel Processing Systems.