A Block Cipher Obtained By Blending Modified Feistel Cipher And Advanced Hill Cipher Involving A Single Key Matrix

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

Abstract

In this investigation, we have developed a block cipher which includes the basic ideas of the modified Feistel cipher and the advanced Hill cipher. In the advanced Hill cipher, as the modular arithmetic inverse of a matrix is the same as the matrix itself, the computations involved in the development of the inverse of the matrix are reduced remarkably. The cryptanalysis carried out in this investigation clearly shows that the strength of the cipher is quite significant as the system of equations occurring in the encryption process are nonlinear, and the supporting functions such as Shift ( ) and Mix ( ) are causing confusion and diffusion in each round of the iteration process.

Key words: Encryption, Decryption, Key matrix, Shift, Mix, XOR Operation.

1. Introduction

In the recent investigations [1-8], we have developed several block ciphers by generalizing the classical Feistel cipher wherein we have considered a plaintext which can be represented in the form of a pair of matrices instead of a pair of binary strings that was used in the case of classical Feistel cipher.

In this development, we have used a key on both the sides of a portion of the plaintext matrix, and made use of iteration. In the iteration process, we have included the features, namely, mixing, permutation, and XOR operation, blending and shuffling. In this, we have seen that the strength of the cipher enhances quite significantly as all the three features, involved in the iteration process thoroughly modify the plaintext before it becomes the cipher text. The avalanche effect and the cryptanalysis discussed in this analysis effectively indicate that the cipher is a strong one.

In a recent investigation, Bibhudenra Acharya et al. [9] have developed Advanced Hill Cipher using an involutory matrix. In the present paper, our objective is to develop the modified Feistel cipher using the features of Advanced Hill cipher.

The basic equations governing the encryption and the decryption of this cipher are given by

$$\left.\begin{array}{l} P_i = (K\, Q_{i-1}\, K) \bmod N \\ Q_i = P_{i-1} \oplus P_i \end{array}\right\} \quad i = 1 \text{ to } n. \qquad (1.1)$$

and

$$\left.\begin{array}{l} Q_{i-1} = (K\, P_i\, K) \bmod N, \\ P_{i-1} = Q_i \oplus P_i \end{array}\right\} \quad i = n \text{ to } 1. \qquad (1.2)$$

where, P_i and Q_i are the plaintext matrices at the i^{th} stage of the iteration, K the involutory key matrix and N is a positive integer chosen appropriately. Here n denotes the number of iterations. In the development of this cipher, we have utilized the functions Mix ( ) and Shift ( ), and XOR operation.

In what follows, we present the plan of the paper. In section 2, we discuss the development of the involutory matrix. In section 3, we discuss the development of the cipher and present the flowcharts and the algorithms describing the cipher. Section 4 is devoted to an illustration of the cipher, and in this we have determined the avalanche effect. We have examined the cryptanalysis in section 5. Finally in section 6, we have given the details of the computations and arrived at the conclusions.

2. Development of the Involutory Matrix

An involutory matrix is a square matrix whose inverse is same as the original matrix.

Let $A = [a_{ij}]$ be a square matrix of size n.
Let it be denoted as

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \qquad (2.1)$$

Where all the sub matrices A_{11}, A_{12}, A_{21} and A_{22} are square matrices of size n/2.

As the modular arithmetic inverse of an involutory matrix is governed by the relations

$$(A\, A^{-1}) \bmod N = I, \qquad (2.2)$$

and $A = A^{-1}$ (2.3)

Thus we have, $A^2 \bmod N = I$ (2.4)

Here I is an identity matrix and N is any non zero positive integer chosen appropriately.

From (2.1) and (2.4) we get,

$A_{22} \bmod N = - A_{11} \bmod N$, (2.5)

$A_{12} = [\, d\,(I - A_{11})] \bmod N$ (2.6)

$A_{21} = [\, w\,(I + A_{11})\,] \bmod N$ (2.7)

where $(dw) \bmod N = 1$ (2.8)

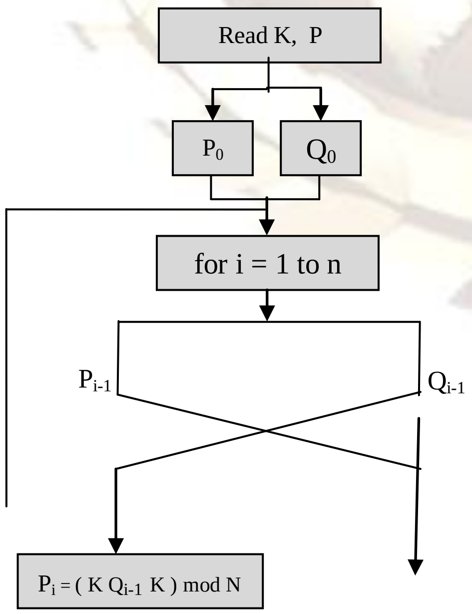
Given A_{11}, on selecting d, where d lies in the interval $0 < d < N$, firstly we obtain w from (2.8), then we determine A_{22}, A_{12} and A_{21} by satisfying the relations (2.5) to (2.7).

If A_{11} is the key, a square matrix of size n/2, then we get the involutory matrix A of size n.

1. Development of the Cipher

Consider a plaintext P consisting of $2m^2$ characters. On employing the EBCIDIC code, the plaintext can be written in the form of a pair of square matrices P_0 and Q_0, wherein, each one is of size m. Let us consider a key matrix K, whose size is mxm.

In this cipher, the encryption and the decryption are governed by the relations (1.1) and (1.2) respectively. In what follows we present the flow charts and the algorithms describing the encryption and the decryption processes.



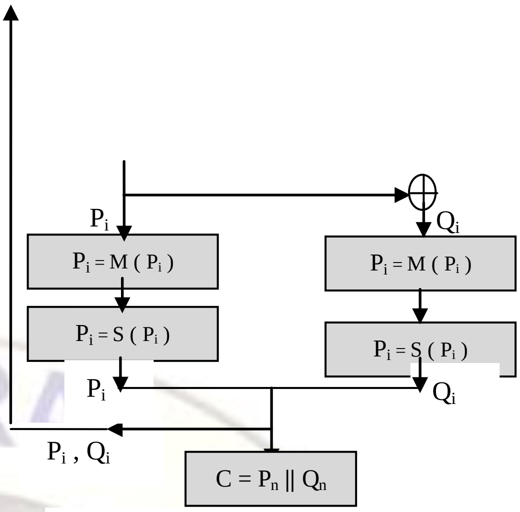


Fig 1. The process of Encryption

In the matrices P_i and Q_i, all the numbers are lying in the interval [0-255]. Here K is the involutory matrix of the key matrix, in this analysis, we have taken n =16.

In the flowchart, for the sake of elegance, we have written the functions Mix ( ) and Shift ( ), arising in encryption, as M ( ) and S ( ) respectively. The reverse processes represented by IMix ( ) and IShift ( ), arising in decryption, are denoted by IM ( ) and IS ( ).

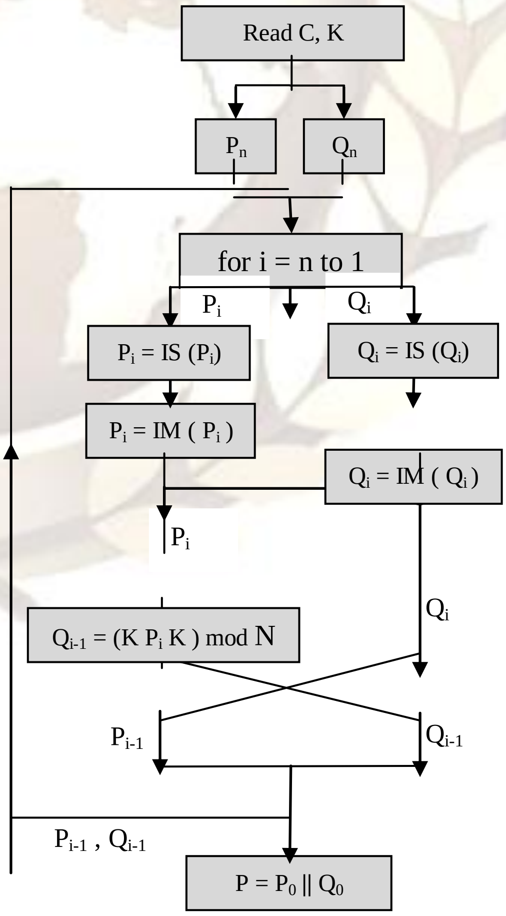


Fig 2. The process of Decryption

The processes of encryption and decryption depicted in the flow charts are described by the algorithms given below.

Algorithm for Encryption

1.Read P, A, n and N.
2.K = Involute (A)
3. P_0 = Left half of P.
 Q_0 = Right half of P.
4. for i = 1 to n
begin
 $P_i = (K\, Q_{i-1}\, K) \bmod N$
 $Q_i = P_{i-1} \oplus P_i$
 $P_i = M (P_i)$
 $P_i = S (P_i)$
 $Q_i = M (Q_i)$
 $Q_i = S (Q_i)$
end
5. $C = P_n \| Q_n$ /* $\|$ represents concatenation */
6. Write(C)

Algorithm for Decryption

1.Read C, A, n and N.
2.K = Involute (A)
3. P_n = Left half of C
 Q_n = Right half of C
4. for i = n to 1
begin
 $P_i = IS (P_i)$
 $P_i = IM (P_i)$
 $Q_i = IS (Q_i)$
 $Q_i = IM (Q_i)$
 $Q_{i-1} = (K\, P_i\, K) \bmod N$
 $P_{i-1} = Q_i \oplus P_i$
end
5. $P = P_0 \| Q_0$ /* $\|$ represents concatenation */
6. Write (P)

Let us now see how the functions (1) Mix ( ) and (2) Shift ( ) can be developed.

In the iteration process, we come across a pair of square matrices of size m. Let us suppose that a square matrix of size m denoted by P_i can be written in the form

$$P_i = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots\cdots\cdots & p_{1m} \\ p_{21} & p_{22} & p_{23} & \cdots\cdots\cdots & p_{2m} \\ p_{31} & p_{32} & p_{33} & \cdots\cdots\cdots & p_{3m} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ p_{m1} & p_{m2} & p_{m3} & \cdots\cdots\cdots & p_{mm} \end{bmatrix} \quad (3.1)$$

On converting each element of the matrix in its 8 bit binary form we get a matrix of size mx8m.

$$P = \begin{bmatrix} p_{111}p_{112}\ldots p_{118} & p_{121}p_{122}\ldots p_{128}\ldots\ldots\ldots & p_{1m1}p_{1m2}\ldots p_{1m8} \\ p_{211}p_{212}\ldots p_{218} & p_{221}p_{222}\ldots p_{228}\ldots\ldots\ldots & p_{2m1}p_{2m2}\ldots p_{2m8} \\ p_{311}p_{312}\ldots p_{318} & p_{321}p_{322}\ldots p_{328}\ldots\ldots\ldots & p_{3m1}p_{3m2}\ldots p_{3m8} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ p_{m11}p_{m12}\ldots p_{m18} & p_{m21}p_{m22}\ldots p_{m28}\ldots\ldots & p_{mm1}p_{mm2}\ldots p_{mm8} \end{bmatrix} \quad (3.2)$$

Here each column contains m binary bits. Now to perform the function Shift ( ) on plaintext P_i, we offer a 4 bit down ward circular shift in each column, during the iteration process. This process is known as shifting. Here, it may be noted that the function IShift ( ), which is the reverse process of Shift ( ) can be readily obtained by giving a 4 bit upward circular shift in every column of the ciphertext C during the iteration process of decryption.

Let us now consider the function Mix ( ), consider P_i as the matrix represented in (3.2) obtained during the iteration process. This matrix is of size mx8m, where each row contains 8m binary bits. On concatenating the bit of 1st column to bit of 8th column, we get a decimal number,on concatenating the bit of 9 column to bit 16 column, we get another decimal number and by continuing this process till we exhaust all rows and columns, we get the decimal numbers to be arranged in row wise manner to get square matrix of size m. This is the process involved in the function Mix ( ). Thus we have the new plaintext obtained after the function Mix( ). IMix ( ) is the reverse process of Mix ( ) which is used during the process of decryption.

2. Illustration of the Cipher

Consider the plaintext given below.

India is a country well known for its great culture, literature etc. Ramakrishna paramahamsa, Vivekananda, Gandhi, these were the great prople. Geetha in Mahabharata is a very great epic. Though it was well known in the past, today India is having crores of liquor shops. Each politician is supporting liquor business. Even the government is getting crores in the form of central exercise tax. I have really been in the country and I have visited every corner. Today, nothing is great about India. One party is fighting with the other. All people are greatly in respect of power. I do not know how Jesus will save them. (4.1)

Let us focus our attention on the first 128 characters of the plaintext. This can be seen as

India is a country well known for its great culture, literature etc. Ramakrishna paramahamsa, Vivekananda, Gandhi, these were th (4.2)

On converting each character of the plaintext in (4.2) into its equivalent EBCIDIC code, we get a plaintext matrix P of size 8x16.

$$\begin{bmatrix} 73 & 110 & 100 & 105 & 97 & 32 & 105 & 115 & 32 & 97 & 32 & 99 & 111 & 117 & 110 & 116 \\ 114 & 121 & 32 & 119 & 101 & 108 & 108 & 32 & 107 & 110 & 111 & 119 & 110 & 32 & 102 & 111 \\ 114 & 32 & 105 & 116 & 115 & 32 & 103 & 114 & 101 & 97 & 116 & 32 & 99 & 117 & 108 & 116 \\ 117 & 114 & 101 & 44 & 32 & 108 & 105 & 116 & 101 & 114 & 97 & 116 & 117 & 114 & 101 & 32 \\ 101 & 116 & 99 & 46 & 32 & 82 & 97 & 109 & 97 & 107 & 114 & 105 & 115 & 104 & 110 & 97 \\ 32 & 112 & 97 & 114 & 97 & 109 & 97 & 104 & 97 & 109 & 115 & 97 & 44 & 32 & 86 & 105 \\ 118 & 101 & 107 & 97 & 110 & 97 & 110 & 100 & 97 & 44 & 32 & 71 & 97 & 110 & 100 & 104 \\ 105 & 44 & 32 & 116 & 104 & 101 & 115 & 101 & 32 & 119 & 101 & 114 & 101 & 32 & 116 & 104 \end{bmatrix} \quad (4.3)$$

Now from (3.3), consider the left half as a matrix represented by P_0, and the right half as a matrix represented by Q_0. Thus we have P_0 and Q_0 as

$$P_0 = \begin{bmatrix} 73 & 110 & 100 & 105 & 97 & 32 & 105 & 115 \\ 114 & 121 & 32 & 119 & 101 & 108 & 108 & 32 \\ 114 & 32 & 105 & 116 & 115 & 32 & 103 & 114 \\ 117 & 114 & 101 & 44 & 32 & 108 & 105 & 116 \\ 101 & 116 & 99 & 46 & 32 & 82 & 97 & 109 \\ 32 & 112 & 97 & 114 & 97 & 109 & 97 & 104 \\ 118 & 101 & 107 & 97 & 110 & 97 & 110 & 100 \\ 105 & 44 & 32 & 116 & 104 & 101 & 115 & 101 \end{bmatrix} \quad (4.4)$$

and

$$Q_0 = \begin{bmatrix} 32 & 97 & 32 & 99 & 111 & 117 & 110 & 116 \\ 107 & 110 & 111 & 119 & 110 & 32 & 102 & 111 \\ 101 & 97 & 116 & 32 & 99 & 117 & 108 & 116 \\ 101 & 114 & 97 & 116 & 117 & 114 & 101 & 32 \\ 97 & 107 & 114 & 105 & 115 & 104 & [illegible] & [illegible] \\ 97 & 109 & 115 & 97 & 44 & 32 & 86 & 105 \\ 97 & 44 & 32 & 71 & 97 & 110 & 100 & 104 \\ 32 & 119 & 101 & 114 & 101 & 32 & 116 & 104 \end{bmatrix} \quad (4.5)$$

Let us take a key matrix denoted by A of size 4x4 as

$$A = \begin{bmatrix} 69 & 124 & 27 & 167 \\ 135 & 79 & [illegible] & [illegible] \\ 248 & 199 & 209 & 75 \\ 239 & 45 & 255 & 92 \end{bmatrix} \quad (4.6)$$

On using the relations (2.5) to (2.7) and taking d =99, we get the involutory matrix of key matrix A as K, whose size is 8x8.
Thus we have the key K as

$$K = \begin{bmatrix} 69 & 124 & 27 & 167 & 180 & 12 & 143 & 107 \\ 135 & 79 & 99 & 111 & 203 & 214 & 183 & 19 \\ 248 & 199 & 209 & 75 & 24 & 11 & 144 & 255 \\ 239 & 45 & 255 & 92 & 147 & 153 & 99 & 207 \\ 130 & 84 & 233 & 237 & 187 & 132 & 229 & 89 \\ 141 & 112 & 1 & 133 & 121 & 177 & 157 & 145 \\ 168 & 77 & 134 & 249 & 8 & 57 & 47 & 181 \\ 5 & 47 & 181 & 63 & 17 & 211 & 1 & 164 \end{bmatrix} \quad (4.7)$$

On using (4.4), (4.5), (4.7) and the functions Shift ( ) and Mix ( ), we adopt the encryption algorithm given in section 3 and obtain the ciphertext C as

$$C=\begin{bmatrix}
28 & 188 & 242 & 196 & 4 & 152 & 196 & 240 & 246 & 144 & 220 & 218 & 18 & 68 & 132 & 224 \\
58 & 110 & 90 & 220 & 182 & 222 & 22 & 146 & 102 & 108 & 30 & 8 & 44 & 84 & 154 & 100 \\
200 & 58 & 122 & 230 & 136 & 10 & 48 & 140 & 172 & 236 & 106 & 176 & 182 & 172 & 138 & 74 \\
140 & 230 & 146 & 62 & 190 & 54 & 42 & 54 & 210 & 230 & 44 & 254 & 210 & 88 & 170 & 52 \\
36 & 176 & 86 & 104 & 144 & 118 & 246 & 206 & 16 & 22 & 98 & 18 & 194 & 218 & 70 & 114 \\
110 & 94 & 150 & 150 & 48 & 236 & 164 & 108 & 6 & 206 & 70 & 114 & 96 & 194 & 166 & 120 \\
182 & 216 & 250 & 66 & 4 & 56 & 202 & 118 & 184 & 38 & 248 & 110 & 168 & 182 & 100 & 138 \\
122 & 206 & 126 & 250 & 150 & 62 & 124 & 54 & 110 & 102 & 100 & 124 & 152 & 128 & 122 & 186
\end{bmatrix} \quad (4.8)$$

On applying the decryption algorithm given in section 3 and by using the inputs (4.7) , (4.8), along with the functions IShift ( ) and IMix ( ), we get back the original plaintext given in (4.3).

Let us now examine the avalanche effect which indicates the strength of the cipher in a qualitative manner. To carry out this one, we first consider a one bit change in the plaintext (4.3), this can be done by changing the first row first column element in (4.3) from 73 to 72. By applying the encryption algorithm given in section 3, on the modified plaintext and by keeping the key in (4.7) as it is, we get the cipher text as

$$C=\begin{bmatrix}
38 & 238 & 178 & 202 & 248 & 222 & 182 & 86 & 188 & 222 & 116 & 122 & 156 & 166 & 178 & 236 \\
16 & 146 & 104 & 90 & 188 & 164 & 136 & 6 & 186 & 108 & 152 & 44 & 170 & 126 & 252 & 64 \\
196 & 88 & 182 & 248 & 242 & 28 & 228 & 146 & 50 & 2 & 204 & 220 & 152 & 212 & 174 & 166 \\
64 & 212 & 198 & 48 & 204 & 138 & 186 & 28 & 204 & 190 & 12 & 156 & 150 & 60 & 48 & 16 \\
216 & 58 & 124 & 200 & 88 & 114 & 82 & 86 & 138 & 114 & 218 & 122 & 98 & 72 & 224 & 236 \\
244 & 184 & 58 & 120 & 14 & 212 & 176 & 146 & 182 & 142 & 58 & 108 & 102 & 56 & 44 & 144 \\
24 & 52 & 8 & 36 & 150 & 172 & 204 & 224 & 222 & 220 & 242 & 70 & 210 & 88 & 184 & 20 \\
176 & 190 & 138 & 48 & 140 & 170 & 114 & 110 & 164 & 98 & 234 & 214 & 146 & 190 & 120 & 202
\end{bmatrix} \quad (4.9)$$

Now on comparing (4.8) and (4.9) in their binary form, we notice that they differ by 518 bits out of 1024 bits. This shows that the cipher has good strength.

Now let us consider a one bit change in the key. This can be carried out by converting the first row first column element of (4.7) from 69 to 70. On adopting the encryption algorithm given in section 3 along with the modified key, and by keeping the plaintext in (4.3) as it is, we get the ciphertext as

$$\begin{bmatrix}
120 & 122 & 110 & 0 & 218 & 122 & 222 & 16 & 22 & 72 & 238 & 72 & 98 & 248 & 58 & 104 \\
240 & 44 & 58 & 150 & 218 & 226 & 236 & 176 & 36 & 36 & 102 & 152 & 6 & 110 & 76 & 98 \\
216 & 218 & 206 & 90 & 254 & 140 & 230 & 146 & 62 & 190 & 170 & 102 & 250 & 86 & 44 & 34 \\
114 & 42 & 204 & 136 & 10 & 48 & 140 & 164 & 150 & 168 & 144 & 118 & 246 & 206 & 246 & 180 \\
16 & 52 & 152 & 86 & 76 & 242 & 200 & 58 & 120 & 88 & 140 & 36 & 140 & 58 & 198 & 114 \\
96 & 194 & 192 & 108 & 240 & 16 & 126 & 108 & 84 & 208 & 220 & 218 & 224 & 236 & 182 & 206 \\
60 & 156 & 148 & 246 & 8 & 72 & 172 & 64 & 60 & 222 & 44 & 222 & 210 & 230 & 118 & 216 \\
228 & 246 & 254 & 166 & 216 & 146 & 234 & 146 & 126 & 150 & 44 & 248 & 206 & 222 & 40 & 104
\end{bmatrix} \quad (4.10)$$

On comparing (4.8) and (4.10) in their binary form we notice that they differ by 504 bits out of 1024 bits. This also shows that the strength of the cipher is quite considerable one.

3. Cryptanalysis

In the study of cryptography, the different cryptanalytic attacks seen in the literature [10] are

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

In all these cryptanalytic attacks, the ciphertext and the encryption are available to the attacker. Generally an encryption algorithm must be designed to withstand the first two attacks [10].

Firstly, let us consider the brute force attack. In this cipher as the key is consisting of sixteen decimal numbers and as each element of the key can be represented as 8 bit EBCIDIC code format, the size of the key is 112 bits. Hence the size of the key space is

$$2^{112} = (2^{10})^{11.2} \approx (10^3)^{11.2} = 10^{33.6}$$

If the time taken to get the plaintext by using one value of the key in the key space is10^{-7} seconds, then the time taken to execute the cipher with all possible keys in key space is

$$\frac{10^{33.6} \times 10^{-7}}{365 \times 24 \times 60 \times 60} = 3.171 \times 10^{18.6} \text{ years.}$$

As this number is very large, it is impractical to break the cipher by using this attack.

Now let us examine the known plaintext attack, to carry out this attack, assume that the attacker knows the plaintext P ciphertext C pairs as many as required.

If we confine our attention to only one round of the iteration process i.e r = 1. Then the relations governing the encryption process for one round can be written as

$$P_1 = (K\ Q_0\ K) \bmod N \quad (5.1)$$
$$Q_1 = P_0 \oplus P_1 \quad (5.2)$$
$$P_1 = M (P_1) \quad (5.3)$$
$$P_1 = S (P_1) \quad (5.4)$$
$$Q_1 = M (Q_1) \quad (5.6)$$
$$Q_1 = S (Q_1) \quad (5.7)$$
$$C = P_1 \,\|\, Q_1 \quad (5.8)$$

In known plaintext attack, we know P_0, Q_0 and C and the encryption algorithm. As the Shift ( ) and Mix ( ) fucntions are known, using (5.8) to (5.3), we get P_1 and Q_1 occurring on the left hand side of (5.1) and (5.2) As P_0 and the Q_0 occurring on the right hand side of (5.1) and (5.2) are known to the attacker, the key K can not be obtained due to the modulo of N used in (5.1). Thus the cipher cannot be broken by the known plaintext attack even if we confine our attention only to the first round of the iteration process. This shows that it is impossible to break the final cipher obtained after sixteen rounds of the iteration process by using the known plaintext attack.

Intuitively choosing a plaintext or ciphertext and determining the key or a function of the key is a formidable task in the case of this cipher.

Hence from the above discussion we conclude that this cipher is not breakable by all the possible attacks that are available in cryptography.

4. Computations and Conclusions

In this paper, we have investigated the modified Fesitel cipher. In order to introduce confusion and diffusion, we have used a pair of functions namely Shift ( ) and Mix( ), and the modulo operation in every round of the iteration. As these features thoroughly mix the plaintext at every stage of the iteration process, the strength of the cipher is good which is proved by the avalanche effect discussed in section 4.

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

In order to encrypt the entire plaintext given in (4.1), the entire plaintext is divided into 5 blocks, with each block containing 128 characters. As the last block contains only 107 characters, we have included 21 blank characters in order to make it a complete block of size 128 characters. On adopting the encryption algorithm given in section 3 with necessary inputs, we get the ciphertext corresponding to the entire plaintext as

28 188 242 196 4 152 196 240 246 144 220 218 18 68 132 224
58 110 90 220 182 222 22 146 102 108 30 8 44 84 154 100
200 58 122 230 136 10 48 140 172 236 106 176 182 172 138 74
140 230 146 62 190 54 42 54 210 230 44 254 210 88 170 52
36 176 86 104 144 118 246 206 16 22 98 18 194 218 70 114
110 94 150 150 48 236 164 108 6 206 70 114 96 194 166 120
182 216 250 66 4 56 202 118 184 38 248 110 168 182 100 138
122 206 126 250 150 62 124 54 110 102 100 124 152 128 122 186

248 128 178 74 26 114 158 190 106 90 244 52 198 114 74 22
200 218 70 128 96 220 182 222 22 108 136 40 42 52 200 58
122 98 22 244 108 246 54 242 142 166 210 176 104 26 144 86
108 64 88 140 108 36 188 76 130 182 190 118 180 170 192 122
186 102 158 64 218 194 204 96 44 96 222 58 146 14 22 2
102 136 138 242 98 174 142 230 222 14 108 190 120 128 164 194
194 218 38 178 222 244 102 60 144 0 246 124 182 122 74 152
222 102 150 222 250 154 102 78 174 72 254 246 118 86 216 32
184 206 222 52 172 250 26 228 214 90 254 140 224 0 100 150
50 98 184 36 138 228 236 162 192 176 238 90 110 138 182 196
148 148 154 100 28 188 48 138 250 182 122 26 120 198 210 232
88 184 128 178 24 218 72 122 152 6 110 124 238 106 86 128
244 116 206 60 130 182 132 152 194 88 194 188 118 38 28 44
6 206 16 22 230 196 92 28 204 190 30 218 124 240 2 72
134 134 182 144 184 20 102 188 144 0 246 124 182 122 74 152
222 102 150 222 250 154 102 78 174 72 254 246 118 86 216 32
242 220 236 46 124 12 242 106 44 254 70 112 128 178 74 24
48 220 146 196 242 118 208 224 88 118 44 182 196 218 98 74
74 204 50 14 222 152 196 124 62 148 138 136 10 48 140 164
150 168 144 118 246 206 246 180 120 14 86 76 242 200 58 120
88 140 36 140 58 198 114 96 194 192 108 240 16 126 108 84
208 220 218 224 236 182 206 16 52 152 120 122 104 26 144 86
106 200 16 14 118 118 216 228 246 254 172 158 198 222 118 244
54 204 158 92 146 252 236 236 33 221 220 45 244 16 65 200
218 206 254 166 216 146 76 100 190 232 206 16 22 72 98 38
58 146 88 190 78 218 60 10 14 228 182 248 186 108 72 72
88 166 64 218 210 24 142 166 210 176 112 2 228 186 218 92
250 26 234 48 140 164 150 168 144 118 246 206 246 180 120 14
86 76 242 200 58 120 88 140 36 140 58 198 114 96 194 192
108 240 16 126 108 84 208 220 218 224 236 182 206 16 52 152
120 122 104 26 144 86 106 200 16 14 118 118 216 228 246 254
172 158 206 92 146 252 236 236 174 176 66 150 20 222 16 37

5. References

[1] V.U.K. Sastry and K. Anup Kumar, "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", International Journal of Computer Technology and Applications, ISSN 2229-6093,Vol 3 (1), pp, 23-31 , 2012.

[2] V.U.K. Sastry and K. Anup Kumar, "A Modified Feistel Cipher involving a key as a multiplicant on both the sides of the Plaintext matrix and supplemented with Mixing, Permutation and Modular Arithmetic Addition", International Journal of Computer Technology and Applications, ISSN 2229-6093,Vol 3 (1), pp, 32-39 , 2012.

[3] V.U.K. Sastry and K. Anup Kumar, "A Modified Feistel Cipher Involving a Key as a Multiplicant on Both the Sides of the Plaintext Matrix and Supplemented with Mixing, Permutation, and Modular Arithmetic Addition", International Journal of Computer Science and Information Technologies ISSN 0975 – 9646. Vol. 3 (1) , 2012, pp, 3133 – 3141.

[4] V.U.K. Sastry and K. Anup Kumar, "A Modified Feistel Cipher involving a pair of key matrices, Supplemented with Modular Arithmetic Addition and Shuffling of the plaintext in each round of the iteration process", International Journal of Computer Science and Information Technologies ISSN 0975 – 9646. Vol. 3 (1) , 2012, pp, 3119 – 3128.

[5] V.U.K Sastry and K. Anup Kumar, "A Modified Feistel Cipher Involving XOR Operation and Modular Arithmetic Inverse of a Key Matrix" International Journal of Advanced Computer Science and Applications ISSN : 2156-5570(Online), U.S.A ,Published in Vol.3. N0.7, 2012

[6] V.U.K Sastry and K. Anup Kumar, "A Modified Feistel Cipher Involving Modular Arithmetic Addition and Modular Arithmetic Inverse of a Key Matrix" International Journal of Advanced Computer Science and Applications ISSN : 2156-5570(Online), U.S.A Published in Vol . 3 No. 7, 2012

[7] V.U.K Sastry and K. Anup Kumar, "A Modified Feistel Cipher Involving Substitution, Shifting of rows, Mixing of columns, XOR operation with a Key and

Shuffling" International Journal of Advanced Computer Science and Applications ISSN : 2156-5570(Online), U.S.A. Accepted for Publication Vol.3. No. 8 , 2012

[8] V.U.K Sastry and K. Anup Kumar, "A Modified Feistel Cipher Involving Key Based Substitution, Shifting of rows, Key Based Mixing of columns, Modular Arithmetic Addition and Shuffling (IJERA) Accepted for Publication , Aug, 2012.

[9] Bhibhudendra Acharya, Saroj Kumar Pahigrahy, Sarat Kumar Parta, and Ganapati Panda, "Image encryption using Advanced Hill cipher Algorithm", International Journal of Recent Trends in Engineering, Vol . 1(1), May 2009.

[10] William Stallings, Cryptography and Network Security, Principles and Practice, Third Edition, Pearson, 2003.

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 done 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.