V.U.K. Sastry, K. Anup Kumar / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 5, September- October 2012, pp.237-245

A Modified Feistel Cipher Involving Key Based Substitution, Shifting Of Rows, Key Based Mixing Of Columns, Modular Arithmetic Addition And Shuffling

[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 paper, we have developed a block cipher by modifying the Feistel cipher. Here we have taken the plaintext in the form of a pair of matrices and we have introduced a set of operations called key based substitution, Shifting of rows, Key based mixing of columns, modular arithmetic addition and shuffling. In this analysis, the key based substitution and the key based mixing of columns play a vital role in strengthening the cipher. The cryptanalysis carried out in this investigation clearly indicates that the cipher is a strong one and it cannot be broken by any conventional attack in cryptography.

Key words: encryption, decryption, cryptanalysis, avalanche effect, modular arithmetic addition.

1. Introduction
The basic ideas under lying in the development of the Feistel cipher laid the foundation for the development of several block ciphers. Some of them are, DES [1] and AES [2].

In a recent investigation [3], we have developed a block cipher called modified Feistel cipher which has included the functions namely, substitute, shifting of rows, mixing of columns, XORing with key and shuffling of elements in the plaintext. From the cryptanalysis carried out in the investigation, it has been found that the cipher is a strong one on account of all the aforementioned functions applied on the plaintext in the encryption process.

In the present analysis, our objective is to arrive at a novel block cipher, by modifying the Feistel cipher. In this we have introduced key based substitution; key based mixing and modular arithmetic addition of the plaintext and the key. In addition to these, features, we have included here, the process of shifting of rows and shuffling the elements of the plaintext. Our interest here is to develop a very strong block cipher which cannot be broken by any cryptanalytic attack.

In what follows we present the plan of the paper. In section 2 we introduce the development of the cipher, and present the flowcharts and the algorithms required in this analysis. We illustrate the cipher by giving a suitable example in section 3. In addition to this, here we discuss avalanche effect. Section 4 is devoted to the study of cryptanalysis. In section 5, we deal with computations and draw conclusions.

2. Development of the cipher
Consider a plaintext P having $2m^2$ characters. On using EBCIDIC code, P can be written in the form of a matrix given by

$P = [P_{ij}]$ i= 1 to m and j = 1 to 2m

Now the matrix P can be written in the form of a pair of square matrices, given by
$P_0 = [P_{ij}]$ i = 1 to m and j= 1 to m

and

$Q_0 = [Q_{ij}]$ i = 1 to m and j= 1 to m.

Let us take the key matrix, K in the form

$K= [K_{ij}]$ i = 1 to m and j = 1 to m.

In the process of encryption, we have used some functions, namely, key based substitution, shifting of rows, key based mixing, modular arithmetic addition and shuffling. In this analysis, the functions key based substitution, shifting of rows, key based mixing, and shuffling are denoted by KSub ( ), Shift ( ), KMix ( ), and Shuffle ( ) respectively.

In the development of the cipher, we have used the functions KSub ( ), Shift ( ), and KMix ( ), together with modular arithmetic addition on both P_{i-1} and Q_{i-1}, and the resulting plaintexts are shuffled in a specific manner. The details of these functions are given a little later. It may be noted here that the aforementioned functions, but for the shuffle, can be used in any order.

The flow charts describing encryption and decryption, in the present modified Feistel cipher, can be depicted as shown below.

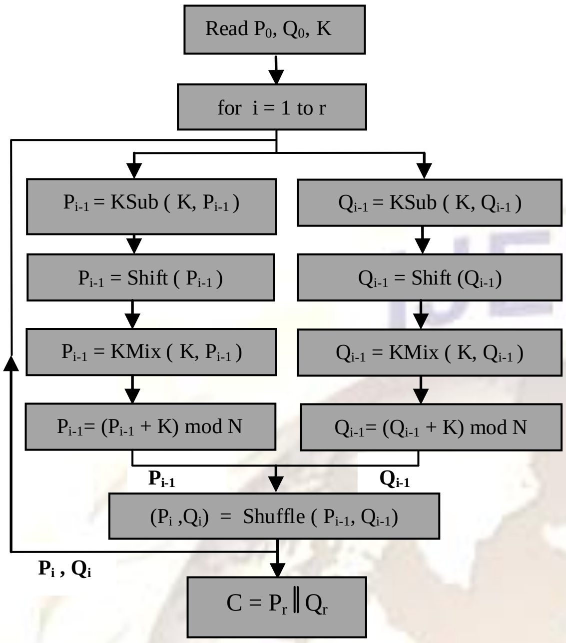

Fig 1. The process of encryption

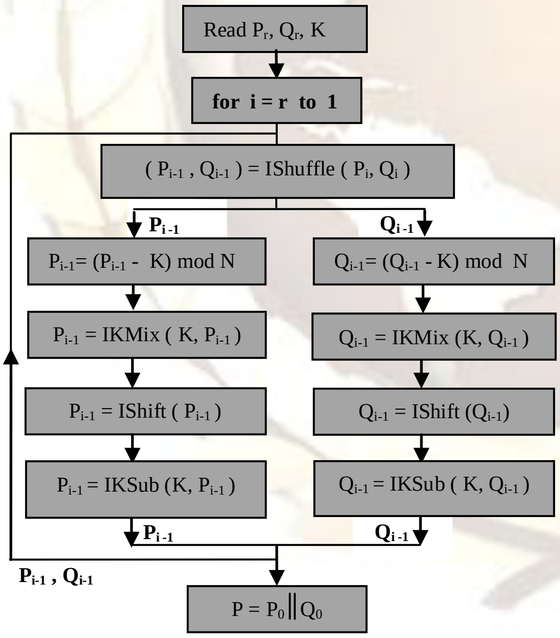

Fig 2. The process of Decryption

Now we write the algorithms for the process of encryption and for the process of decryption as given below.

Algorithm for Encryption

1. Read P, K
2. P_0 = Left half of P.
3. Q_0 = Right half of P.
4. for i = 1 to r

   begin

   P_{i-1} = KSub (K, P_{i-1})

   P_{i-1} = Shift (P_{i-1})

   P_{i-1} = KMix (K, P_{i-1})

   P_{i-1} = (P_{i-1} + K) mod N

   Q_{i-1} = KSub (K, Q_{i-1})

   Q_{i-1} = Shift (Q_{i-1})

   Q_{i-1} = KMix (K, Q_{i-1})

   Q_{i-1} = (Q_{i-1} + K) mod N

   (P_i , Q_i) = Shuffle (P_{i-1}, Q_{i-1})

   end
5. C = $P_r \| Q_r$ /* $\|$ represents concatenation */
6. Write(C)

Algorithm for Decryption

1. Read C, K
2. P_r = Left half of C.
3. Q_r = Right half of C.
4. for i = r to 1

   begin

   (P_{i-1} , Q_{i-1}) = IShuffle (P_i, Q_i)

   P_{i-1} = (P_{i-1} - K) mod N

   P_{i-1} = IKMix (K, P_{i-1})

   P_{i-1} = Shift (P_{i-1})

   P_{i-1} = IKSub (K, P_{i-1})

   Q_{i-1} = (Q_{i-1} - K) mod N

Q_{i-1} = IKMix (K, Q_{i-1})

Q_{i-1} = Shift (Q_{i-1})

Q_{i-1} = IKSub (K, Q_{i-1})

In order to have a clear insight into the basic ideas underlying in the different functions involved in the development of the cipher, for simplicity, let us take the key matrix whose size is 4.

Thus we have

$$K = \begin{bmatrix} 33 & 115 & 220 & 18 \\ 93 & 62 & 13 & 190 \\ 142 & 255 & 10 & 82 \\ 96 & 15 & 43 & 73 \end{bmatrix} \qquad (2.1)$$

end

5. P = $P_0 \| Q_0$ /* $\|$ represents concatenation */

6. Write (P)

We now see the formation of the function KSub ( ), which is based upon the elements of K. Consider a square matrix of size 16. Let us fill up the first row of this matrix with the elements of the key taken in the row wise order.

Excluding these numbers, which are occurring in the key, let us fill up the rest of the positions of the matrix with the remaining integers occurring in 0 to 255, maintaining the order of the integers. Thus we get the key based substitution table in the form, wherein hexadecimal notation is used in the representation of rows, columns and the numbers occurring in the Table.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	21	73	DC	12	5D	3E	0D	BE	8E	FF	0A	52	60	0F	2B	49
1	00	01	02	03	04	05	06	07	08	09	0B	0C	0E	10	11	13
2	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	22	23	24
3	25	26	27	28	29	2A	2C	2D	2E	2F	30	31	32	33	34	35
4	36	37	38	39	3A	3B	3C	3D	3F	40	41	42	43	44	45	46
5	47	48	4A	4B	4C	4D	4E	4F	50	51	53	54	55	56	57	58
6	59	5A	5B	5C	5E	5F	61	62	63	64	65	66	67	68	69	6A
7	6B	6C	6D	6E	6F	70	71	72	74	75	76	77	78	79	7A	7B
8	7C	7D	7E	7F	80	81	82	83	84	85	86	87	88	89	8A	8B
9	8C	8D	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C
A	9D	9E	9F	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC
B	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC
C	BD	BF	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD
D	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DD	DE
E	DF	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE
F	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE

Table 1. Key Based Substitution Box

The inverse substitution table, corresponding to the above substitution table, can be obtained in the form

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	10	11	12	13	14	15	16	17	18	19	0A	1A	1B	06	1C	0D
1	1D	1E	03	1F	20	21	22	23	24	25	26	27	28	29	2A	2B
2	2C	00	2D	2E	2F	30	31	32	33	34	35	0E	36	37	38	39
3	3A	3B	3C	3D	3E	3F	40	41	42	43	44	45	46	47	05	4B
4	49	4A	4B	4C	4D	4E	4F	50	51	0F	52	53	54	55	56	57
5	58	59	0B	5A	5B	5C	5D	5E	5F	60	61	62	63	04	64	65
6	0C	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74
7	75	76	77	01	78	79	7A	7B	7C	7D	7E	7F	80	81	82	83
8	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	08	92
9	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	A0	A1	A2
A	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2
B	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	C0	07	C1
C	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1
D	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	02	DE	DF	E0
E	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0
F	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF	09

Table 2. Key Based Inverse Substitution Box

As these tables are having sixteen rows and sixteen columns, we have made use of the hexadecimal notation of the numbers 0 to 15 in row wise manner as well as column wise manner.

The usage of the substitution table can be done as follows.

Let us suppose that, a character occurring in the plaintext is represented by a number N_u. Let this be written in the form of 8 binary bits. The most significant four binary bits will specify the row in the table, and the least significant four binary bits will specify the column in the table. Thus we get a number corresponding to N_u. So, substitution can be carried out by replacing N_u with the number occurring in the specified row and the specified column. The inverse substitution process can be carried out in a similar manner by using the inverse substitution table.

The details of the process involved in the function shift can readily be found in [3].

Let us now consider the development of the function KMix ( ) which includes the mixing process that depends upon the key.

Consder the set of numbers, occurring in the key, by taking them in the row wise order of the key matrix (2.1).
Let us label each one of the elements occurring in the key, by a number lying in [0, 15] (see third row of the Table), assuming that the key numbers are

arranged in ascending order. Thus we get the Table given below.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
y	33	115	220	18	93	62	13	190	142	225	10	82	96	15	43	73
z	4	11	14	3	9	6	1	13	12	15	0	8	10	2	5	7

Table 3. Key Based Mixing

Here x denotes the serial number, y denotes the numbers in the key, and z indicates the order of the key numbers (corresponding to the ascending order numbers in the key).

This table will be used for the purpose of mixing. Let us now see how the mixing is carried out.

Consider the plaintext obtained at some stage of the iteration process. Let it be denoted by

P[ij], i= 1 to 4 and j= 1 to 4.

On writing each element in its binary form, we get the plaintext matrix in the form

$$\begin{bmatrix} P_{111}P_{112}\ldots\ldots P_{118} & P_{121}P_{122}\ldots\ldots P_{128} & P_{131}P_{132}\ldots\ldots P_{138} & P_{141}P_{142}\ldots\ldots P_{148} \\ P_{211}P_{212}\ldots\ldots P_{218} & P_{221}P_{222}\ldots\ldots P_{228} & P_{231}P_{232}\ldots\ldots P_{238} & P_{241}P_{242}\ldots\ldots P_{248} \\ P_{311}P_{312}\ldots\ldots P_{318} & P_{321}P_{322}\ldots\ldots P_{328} & P_{331}P_{332}\ldots\ldots P_{338} & P_{341}P_{342}\ldots\ldots P_{348} \\ P_{411}P_{412}\ldots\ldots P_{418} & P_{421}P_{422}\ldots\ldots P_{428} & P_{431}P_{432}\ldots\ldots P_{438} & P_{441}P_{442}\ldots\ldots P_{448} \end{bmatrix}$$

This contains four rows and thirty two columns.

In the process of mixing, we interchange the columns indicated by the pair x and z (see Table 3), till we exhaust all the first sixteen columns in the matrix. Then we adopt the same procedure for the remaining sixteen columns by following the numbers corresponding to x and z in Table 3. Thus, we get a new matrix of the plain text of the same size 4x32. Then by taking the binary bits in two adjacent columns in to consideration, we can form a decimal number. Thus, we get sixteen decimal numbers. These numbers are arranged in row wise manner and hence we get a 4x4 matrix. This gives us the resulting matrix after mixing.

The process involved in the function shuffle is given in [4].

Illustration of the cipher

Consider the plaintext given below

My dear young man, you are very much correct. But you did not realize the gap. Our families are having the same status financially, politically and socially. I agree we both belong to the same cast. I tell you when my sister was to marry a Brahmin some years back, the cast became a problem for them. Of course they could win over the difficulty by taking a firm decision. They got married and they are happy. Today our problem is a different one. We both are well qualified, we can go anywhere, the trouble is with my father and your father. You know my father belongs to congress and your father belongs to BJP. They are not accepting our marriage. I tell you firmly we have to take our own decision and act in an appropriate manner. Yours Y. (3.1)

Let us focus our attention on the first 32 characters of the plaintext. This is given by

My dear young man, you are very (3.2)

On using EBCIDIC code, we get

$$P = \begin{bmatrix} 77 & 121 & 32 & 100 & 101 & 97 & 114 & 32 \\ 121 & 111 & 117 & 110 & 103 & 32 & 109 & 97 \\ 110 & 44 & 32 & 121 & 111 & 117 & 32 & 97 \\ 114 & 101 & 32 & 118 & 101 & 114 & 121 & 32 \end{bmatrix} \quad (3.3)$$

This can be written in the form

$$P_0 = \begin{bmatrix} 77 & 121 & 32 & 100 \\ 121 & 111 & 117 & 110 \\ 110 & 44 & 32 & 121 \\ 114 & 101 & 32 & 118 \end{bmatrix} \quad (3.4)$$

and

$$Q_0 = \begin{bmatrix} 101 & 97 & 114 & 32 \\ 103 & 32 & 109 & 97 \\ 111 & 117 & 32 & 97 \\ 101 & 114 & 121 & 32 \end{bmatrix} \quad (3.5)$$

Let us take the key matrix K, in the form

$$K = \begin{bmatrix} 33 & 115 & 220 & 18 \\ 93 & 62 & 13 & 190 \\ 142 & 255 & 10 & 82 \\ 96 & 15 & 43 & 73 \end{bmatrix} \quad (3.6)$$

On using the encryption process, mentioned in section 2, in which we use the plaintext portions P_0 and Q_0, the key K and the functions whose details are spelt out in section 2, we get the cipher text C in the form

$$C = \begin{bmatrix} 102 & 129 & 191 & 61 & 81 & 38 & 253 & 244 \\ 201 & 16 & 126 & 233 & 182 & 100 & 254 & 134 \\ 37 & 157 & 190 & 117 & 41 & 110 & 76 & 146 \\ 115 & 203 & 219 & 147 & 36 & 153 & 150 & 119 \end{bmatrix} \quad (3.7)$$

On adopting the decryption process, given in section 2, we get back the original plaintext (3.2). This enabled us to checkup the correctness of the encryption process.

Let us now study the avalanche effect which throws some light on the efficacy of the cipher. In order to carry out this one, firstly, let us consider a one bit change in the plaintext. To achieve this one, we change the second row, second column element of P, from 111 to 110. We notice that these two numbers differ by one binary bit. On applying the encryption algorithm on the modified plaintext, keeping the key as it is, we get the cipher text in the form

$$C = \begin{bmatrix} 190 & 153 & 62 & 72 & 91 & 187 & 124 & 220 \\ 153 & 212 & 165 & 136 & 29 & 188 & 147 & 160 \\ 121 & 207 & 47 & 217 & 150 & 119 & 68 & 30 \\ 233 & 182 & 100 & 177 & 47 & 110 & 168 & 147 \end{bmatrix} \quad (3.8)$$

On comparing (3.7) and (3.8), after writing them in their binary form, we find that, these two cipher texts differ by 121 bits (out of 256 bits).

Let us now explore the effect of one bit change in the key. In order to have this one, let us change the second row, second column element of K, given by (3.6), from 62 to 63. On using the original plaintext and the modified key, we now make use of the encryption algorithm and obtain the cipher text given by

$$C = \begin{bmatrix} 64 & 237 & 228 & 157 & 62 & 129 & 231 & 60 \\ 164 & 148 & 253 & 153 & 103 & 116 & 65 & 238 \\ 155 & 102 & 75 & 18 & 246 & 234 & 137 & 58 \\ 217 & 189 & 244 & 201 & 121 & 51 & 180 & 220 \end{bmatrix} \quad (3.9)$$

Now on comparing the ciphertexts (3.7) and (3.9), on converting them into their binary form, we find that they differ by 133 bits out of 256 bits.

This also shows that this cipher is expected to be a potential one.

1. Cryptanalysis

The cryptanalytic attacks that are well known in the literature of cryptography are

1. Ciphertext only attack (Brute force attack),
2. Known plaintext attack,
3. Chosen plaintext attack,
4. Chosen ciphertext attack.

Generally, every algorithm is to be designed so that it withstands atleast the first two attacks, i.e. , cipher text only attack and the known plaintext attack [5].

Let us now consider the brute force attack.

When the key is taken in the form of a square matrix of size m, then the size of the key space

$2^{(8m^2)}$.

If the time required for the computation of the cipher with one value of the key in the key space is 10^{-7} seconds, then the time required for the execution of the cipher with all the possible keys in the key space is

$$\frac{2^{(8m^2)} \times 10^{-7}}{365 \times 24 \times 60 \times 60} \text{ years}$$

This is approximately equal to

$$\frac{10^{(2.4m^2) - 7}}{365 \times 24 \times 60 \times 60} \text{ years}$$

$$= 3.12 \times 10^{(2.4)m^2 - 15} \text{ years}$$

When m=4, the time required for the entire computation

$$= 10^{(23.4)} \text{ years}$$

In the light of the above discussion, we conclude that, this cipher cannot be broken by anyone of the attacks available in the literature.

As this is a very large quantity, this cipher cannot be broken by the brute force attack.

Now let us examine the known plaintext attack.

In this case, we know as many pairs of plaintext and cipher text as we require for attempting to break the cipher. Thus, in this analysis, we know the corresponding pairs P_0 , Q_0 and P_r , Q_r , as many as we require, for breaking the cipher.

If we confine our attention only to one round of the iteration process, that is, if we take r=1, we have

P_0=KSub(K,P_0) , (4.1)

P_0 = Shift (P_0) , (4.2)

P_0=KMix(K,P_0) , (4.3)'

P_0=(P_0+K) mod N. (4.4)

Q_0=KSub(K,Q_0), (4.5)

Q_0 = Shift (Q_0) , (4.6)

Q_0=KMix(K,Q_0), (4.7)

Q_0=(Q_0+K) mod N , (4.8)

(P_1 , Q_1) = Shuffle (P_0, Q_0) (4.9)

C = $P_1 \| Q_1$ (4.10)

From (4.10), we can readily obtain P_1 and Q_1 as C is known to us. Now on using IShuffle ( ), the reverse process of Shuffle ( ), we get P_0 and Q_0, occurring in the left hand side of (4.4) and (4.8) respectively.

We know the P_0 and Q_0 occurring on the right hand side of (4.1) and (4.5), as the plaintext at the beginning of the iteration is known to us. But we cannot determine P_0 and Q_0, occurring on the left hand side of (4.1) and (4.5), as the key K is unknown to us, and hence the reverse process of the key dependent substitution, IKSub ( ), cannot carried out.

In the light of the aforementioned discussion, we conclude that the key K cannot be found even in the first round of the iteration process. Thus, this cipher cannot be broken by the known plaintext attack.

Intuitively, choosing either the plaintext or the ciphertext and proceeding for breaking the cipher, either by the third attack or by the fourth attack is effectively ruled out as we are having several functions, such as KSub ( ), Shift ( ), KMix ( ) , modular arithmetic addition and shuffle ( ) for modifying the plaintext in each round of the iteration process.

5. Computations and Conclusions

In this paper, we have developed a block cipher, by modifying the Feistel cipher, in which we have included several functions, namely, KSub ( ), Shift ( ), KMix ( ), modular arithmetic addition and shuffle ( ), for creating confusion and diffusion in an effective manner. The cryptanalysis that we have carried out in this investigation clearly shows that the cipher is a very strong one, even if we confine our attention only to one round of the iteration process.

The programs required for carrying out the computations in encryption and decryption are written in C language.

The entire plaintext given in (3.1) is divided into 24 blocks, wherein each one is having 32 characters. In the last block, as we have only seven characters, we have appended 25 blank characters to make it a complete block of 32 characters. On carrying out the encryption process on these blocks, the ciphertext corresponding to this plaintext (excluding the cipher text of the first block which is already given in (3.7)) is obtained as follows.

130 215 12 162 180 115 200 71 240
05 120 63 209 76 10 199
116 220 234 201 11 168 110 36 12
34 115 20 56 121 212 86
111 15 162 130 122 12 131 62 218
191 123 56 111 126 12 17
19 33 140 118 176 58 120 113 230
118 06 37 50 200 17 255
216 14 33 158 171 88 122 215 213
77 16 109 133 140 184 01
12 32 111 18 161 170 114 118 33
48 54 190 22 19 158 39
63 116 219 55 84 73 191 125 50
68 51 64 223 148 254 134
37 157 190 117 41 110 76 146 115
203 219 147 36 153 150 119
115 247 77 179 36 71 238 155 102
51 64 223 148 254 134 37
157 190 117 41 110 76 146 115 203
219 147 36 168 147 126 250
211 108 198 104 115 247 77 178 55
229 63 161 137 103 111 157
74 91 147 36 156 242 246 228 201
42 36 223 190 154 102 89
28 250 109 152 205 119 77 233 182
99 52 200 223 77 179 25
174 83 250 24 150 118 249 212 165
185 50 73 207 47 110 76
51 92 167 244 49 44 237 243 158
155 102 50 106 82 220 153
36 231 151 183 38 73 81 38 253
244 211 50 206 242 68 126
161 154 229 63 173 157 190 115 140
77 179 44 237 243 158 140
154 148 183 38 73 57 229 237 201
146 84 73 191 125 52 204
244 219 50 206 223 57 232 201 169
75 114 100 147 158 94 220
153 37 68 155 247 211 76 203 59
201 17 250 57 244 219 49
50 106 82 220 153 36 231 151 183
38 73 81 38 253 244 211
50 206 242 68 126 142 125 54 204
102 187 167 166 70 250 109
91 59 124 231 25 67 53 202 126
155 102 89 219 231 61 25
203 59 201 17 250 57 244 219 49
154 238 158 153 27 233 181
108 237 243 156 101 12 215 41 250
109 153 103 111 156 244 100
212 164 201 169 75 114 100 147 158
83 126 250 105 147 36 156
242 244 155 247 211 76 203 59 201
17 250 57 244 219 49 154
238 158 153 27 233 181 108 237 243
156 101 12 215 41 250 109
153 103 111 156 244 100 212 164 201
169 75 113 185 50 74 143
59 201 17 250 57 244 219 49 154
166 253 244 119 79 76 141
244 218 182 118 249 206 50 134 107
148 253 54 204 179 183 206
122 50 106 82 100 212 165 185 50
73 207 41 191 125 52 201
146 78 121 122 77 251 233 166 101
157 228 136 253 28 250 109
152 205 119 79 76 141 244 218 182
118 249 206 50 134 107 148
253 54 204 179 183 206 122 50 106
82 100 212 165 184 220 153
37 71 183 38 73 81 101 157 228
136 253 28 250 109 152 205
119 79 76 141 244 218 182 118 249
206 50 134 107 148 253 54
204 179 183 206 122 50 106 82 100
212 165 185 50 73 207 105
147 36 156 242 244 155 247 211 76
203 59 201 17 250 57 244
219 49 154 238 158 153 27 233 181
108 237 243 156 101 12 215
41 250 109 153 103 111 156 244 100
212 164 201 169 75 113 185
114 100 149 30 221 166 217 140 213
55 239 164 201 42 44 179
186 122 100 111 166 213 179 183 206
113 148 51 92 167 233 182
101 157 190 115 209 147 82 147 38
165 45 201 146 78 121 77
251 233 166 76 146 115 203 210 111
223 77 51 44 239 36 71
232 231 211 108 198 107 186 122 100
111 166 213 179 183 76 146

The avalanche effect and the cryptanalysis discussed in this investigation are supporting very thoroughly the strength of the cipher, and this suggests that this cipher can be utilized in any context for the security of information.

References

[1] National Bureau of Standards NBS FIPS PUB 46-1, "Data Encryption Standard

(DES) ", National Bureau of Standards, US Department of Commerce, Jan 1988.

[2] Daemen J, and Rijmen V, "Rijndael, the Advanced Encryption Standard (AES)", Dr. Dobbs Journal, Vol. 26(3), pp. 137 - 139, Mar 2001.

[3] A Modified Feistel Cipher Involving Substitution, Shifting of rows, Mixing of columns, XOR operation with a Key and Shuffling (Accepted for publication, IJACSA, U.S.A)

[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, No.1, pp. 3119-3128, 2012.

[5] 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 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.