A Block Cipher Involving a Key Bunch Matrix and Including Another Key Matrix Supplemented with Xor Operation

V.U.K. Sastry, PhD.
Professor of CSE, Director (SCSI),
Dean (Admin), Dean (R&D),
SreeNidhi Institute of Science & Technology,
Hyderabad, India,

K. Shirisha
Associate Professor,
Dept. of Computer Science & Engineering,
SreeNidhi Institute of Science & Technology,
Hyderabad, India,

ABSTRACT

In this paper, we have developed a block cipher which includes a pair of key matrices. The first key matrix E is a key bunch matrix which is containing several odd integers as keys lying in [1-255], and the second key matrix F is an additional matrix, linked with xor operation, containing the integers in [0-255]. The corresponding key bunch D, used in the decryption process, is obtained by using the concept of multiplicative inverse. From the cryptanalysis carried out in this investigation we have found that this cipher is a very strong one and it cannot be broken by any attack.

Keywords

Encryption key bunch matrix, Decryption key bunch matrix, avalanche effect, cryptanalysis.

1. INTRODUCTION

The study of block ciphers is a fascinating area of research in cryptography. Transmission of any secret information in a secured manner can be done by using a block cipher. There are several popular block ciphers [1-5] available in the literature.

In a recent investigation, we have developed a novel block cipher [6], which includes a key bunch matrix in the encryption process and a corresponding key bunch matrix in the decryption process. The keys of the decryption are obtained by choosing the keys for encryption in an appropriate manner and applying the concept of multiplicative inverse in modular arithmetic. The process involved in this cipher is a elegant one and an interesting one.

In the present paper, our objective is to extend the analysis of the cipher discussed in [6] by introducing one more key, and using xor operation. The additional features that we have introduced in this analysis are expected to enhance the strength of the cipher. The basic equations governing the encryption of the cipher are given by

$$C=[c_{ij}]=([e_{ij} \times p_{ij}] \bmod 256) \oplus F, \text{ i=1 to n, j=1 to n.} \qquad (1.1)$$

The equations describing the decryption process are of the form

$$P=[p_{ij}]=[d_{ij} \times (C \oplus F)_{ij}] \bmod 256, \text{ i=1 to n, j = 1 to n.} \qquad (1.2)$$

Here, P= $[p_{ij}]$ is the plain text, C=$[c_{ij}]$, the cipher text, F=$[f_{ij}]$ is an additional key matrix, whose elements are in [0-255], and $[e_{ij}]$ and $[d_{ij}]$ are the key bunch matrices for the encryption and the decryption, respectively. As it has already been pointed out in [6], d_{ij} is the multiplicative inverse of e_{ij} and they are connected by the relation

$$(e_{ij} \times d_{ij}) \bmod 256 = 1. \qquad (1.3)$$

Here it is to be noted that e_{ij} and d_{ij} are odd numbers and they lie in the interval [1-255].

In the present analysis, our interest is to see, how the additional key matrix, F would influence the strength of the cipher, which we have discussed in [6].

In what follows, we present the plan of the paper. In section 2, we discuss the development of the cipher and provide flowcharts and algorithms for this cipher. In section 3, we illustrate the cipher with a suitable example and examine the avalanche effect. We discuss the cryptanalysis in section 4. Finally in section 5, we mention the computations carried out in this investigation and draw conclusions.

2. DEVELOPMENT OF THE CIPHER

Let us consider a plain text P. On using the EBCDIC code this can be written in the form

$$P = [p_{ij}], \text{ i=1 to n, j = 1 to n.} \qquad (2.1)$$

Let, C = $[c_{ij}]$, i=1 to n, j = 1 to n, be the cipher text.

E = $[e_{ij}]$, i=1 to n, j = 1 to n, be the encryption key bunch matrix,

D = $[d_{ij}]$, i=1 to n, j = 1 to n, be the decryption key bunch matrix, and

F = $[f_{ij}]$, i=1 to n, j = 1 to n, be the additional key matrix.

The flowcharts describing the encryption and the decryption are given as shown below in figure 1 and figure 2, respectively.

The corresponding algorithms for encryption and decryption are given below.

Algorithm for Encryption

```
1. Read n,E,P,F,r
2. for k = 1 to r do
   {
3. For i=1 to n do
```

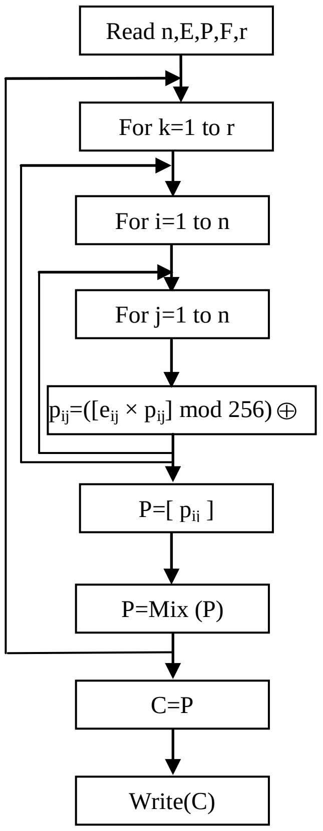

Figure 1. Flowchart for Encryption

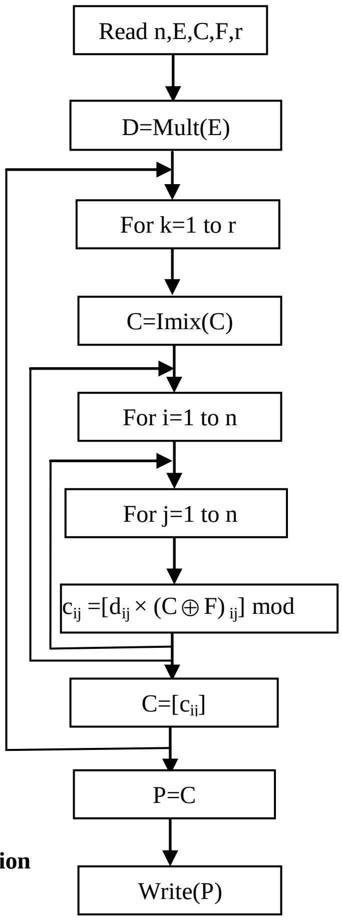

Figure 2. Flowchart for Decryption

```
      {
4.    For j=1 to n do
      {
5.    $p_{ij}$ = ([$e_{ij}$ × $p_{ij}$] mod 256) ⊕ $f_{ij}$
      }
      }
6.    P=[ $p_{ij}$]
7.    P=Mix(P)
      }
8.    C = P
9.    Write(C)
```

Algorithm for Decryption

```
1.  Read n,E,C,F,r
2.  D =Mult(E)
3.  for k = 1 to r do
    {
4.  C=Imix (C)
5.  For i=1 to n do
    {
6.  For j=1 to n do
    {
7.  $c_{ij}$ =[$d_{ij}$ × ($c_{ij}$ ⊕ $f_{ij}$)] mod 256
    }
    }
8.  C=[$c_{ij}$]
    }
9.  P=C
10. Write(P)
```

The number of iterations r =16.
Mult() is a function with which we obtain D for a given E, by using the concept of multiplicative inverse.

In the above flowcharts and the algorithms, we have used the function Mix() in each round of the iteration process. In every round of the iteration process, we convert the elements of the plaintext P into binary bits. We divide the matrix containing the binary bits into two halves (left half and right half). Firstly, the binary bits of first column are converted into decimal numbers by taking eight binary bits at a time, of course when n>8. Then the binary bits of the first column of the second half are converted into decimal numbers and carrying out the same process on all the columns of the first half and the second half one after the another, we get decimal numbers. However when n < 8, we take the binary bits of the first column of the first half first and then concatenate with binary bits of the first column of the second half. We carry out the same procedure till we exhaust all the binary bits in the columns of the first half and second half one after the other and form decimal numbers. Then we write all these numbers, in row-wise manner, in a matrix. Thus we get a matrix of size n x n. The process of mixing is carried out in this fashion. For a detailed discussion of the function Mix(), we may refer to [6]. The function Imix() used in the decryption process, is containing the reverse steps of Mix().

3. ILLUSTRATION OF THE CIPHER AND THE AVALANCHE EFFECT

Consider the plain text given below.

Dear Madam! Our separation has become a very small problem in the recent past. Everyday we are facing the challenges posed by the political parties. Everywhere a strike, a hartal, a dharna. To control the people and to control the politicians has become a hectic problem. I do not know when I will have the transfer. Yours. (3.1)

Let us focus our attention on the first sixteen characters of this plain text. This is given by the following:

Dear Madam! Our

On using the EBCDIC code, we get

$$P = \begin{bmatrix} 196 & 133 & 129 & 153 \\ 64 & 212 & 129 & 132 \\ 129 & 148 & 79 & 64 \\ 214 & 164 & 153 & 64 \end{bmatrix} \qquad (3.2)$$

Let us now choose the key bunch matrix E and the additional matrix F in the form

$$E = \begin{bmatrix} 99 & 147 & 71 & 95 \\ 189 & 153 & 15 & 97 \\ 7 & 191 & 35 & 153 \\ 237 & 143 & 65 & 235 \end{bmatrix} \qquad (3.3)$$

and

$$F = \begin{bmatrix} 222 & 21 & 23 & 58 \\ 99 & 100 & 41 & 167 \\ 77 & 23 & 55 & 78 \\ 91 & 223 & 182 & 135 \end{bmatrix} \qquad (3.4)$$

On using the concept of multiplicative inverse, given in (1.3), we get the decryption matrix D in the form

$$D = \begin{bmatrix} 75 & 155 & 119 & 159 \\ 149 & 169 & 239 & 161 \\ 183 & 63 & 139 & 169 \\ 229 & 111 & 193 & 195 \end{bmatrix} \qquad (3.5)$$

On using the P, the E and the F given by (3.2) – (3.4), and applying the encryption algorithm, given in section 2, we get the ciphertext C in the form

$$C = \begin{bmatrix} 45 & 253 & 66 & 3 \\ 231 & 4 & 236 & 140 \\ 88 & 27 & 190 & 23 \\ 82 & 201 & 191 & 20 \end{bmatrix} \qquad (3.6)$$

On applying the decryption algorithm, we get back the original plain text P.

Let us now examine the avalanche effect. On changing 4th row and 2nd column element of the plaintext P given by (3.2), from 164 to 165, we get a one bit change in the plain text. On using the modified plain text and the E and F, given by (3.3) and (3.4), and applying the encryption algorithm, we get the ciphertext C in the form

$$C = \begin{bmatrix} 130 & 163 & 89 & 10 \\ 116 & 89 & 71 & 167 \\ 178 & 221 & 211 & 40 \\ 85 & 87 & 55 & 230 \end{bmatrix} \qquad (3.7)$$

On comparing, (3.6) and (3.7), in their binary form, we find that these two ciphertexts differ by 70 bits out of 128 bits. This shows that the cipher is a potential one.

Now let us examine the effect of one bit change in the key. On changing the 4th row, 4th column element from 235 to 233, we get a one bit change. On using the modified key bunch matrix and the other required matrices P and F, and applying the encryption algorithm we get the ciphertext C in the form

$$C = \begin{bmatrix} 83 & 91 & 9 & 217 \\ 252 & 55 & 157 & 238 \\ 7 & 242 & 23 & 57 \\ 73 & 56 & 2 & 164 \end{bmatrix} \qquad (3.8)$$

On comparing (3.6) and (3.8), after converting them into their binary form, we find that they differ by 71 bits out of 128 bits. This also shows that the cipher is expected to be a strong one.

4. CRYPTANALYSIS

Let us now study the cryptanalysis, which confirms the characteristic features of the cipher in respect of its strength. The different types of cryptanalytic attacks available in the literature are

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

Generally, in the literature of cryptography [7], the first two attacks are examined very thoroughly and proofs are offered. But in the latter two cases, intuitive inspection is done in a meticulous manner.

Let us now consider the ciphertext only attack. In this cipher, we are having one key bunch matrix, called E, and an additional matrix F. These two matrices are to be sent by the sender to the receiver. Here E is containing all the odd integers lying in [1-255], while F is containing all the integers lying in [0-255], and both are square matrices of size n. Thus the size of the key space is

$$2^{15n^2} = \left(2^{10}\right)^{1.5n^2} \approx \left(10^3\right)^{1.5n^2} = 10^{4.5n^2}.$$

If we assume that the time required for the execution of the cipher with one E and one F in the key space is 10 $^{-7}$ seconds, then the time required for the computation of the cipher with all the keys in the key space is approximately equal to

$$\frac{10^{4.5n^2} \times 10^{-7}}{365 \times 24 \times 60 \times 60} = 3.12 \times 10^{4.5n^2-15} \ years.$$

In this analysis, as we have taken n=4, the time required can be written in the form

3.12 x 10^{57} years.

As this time is unimaginably large, it is impossible to break this cipher by the brute force attack.

Let us now examine the known plaintext attack. In this case, we know very clearly as many pairs of plaintext and ciphertext that we require for our analysis. Firstly, let us focus our attention on only round (r=1) of the iteration process. In this case, the equations governing the cipher (see the algorithm given in section 2) are given by

P=([e_{ij} x p_{ij}] mod256) $\oplus$ F, i = 1 to n, j=1 to n, (4.1)
P = Mix(P), (4.2)
and
C = P (4.3)

As C is known to us, we know P occurring in (4.3), and hence the P occurring on the left hand side of (4.2). On using Imix(), we get the P occurring in the right hand side of (4.2). This gives the P occurring on the left hand side of (4.1). Now, as (4.1) is containing two key matrices, namely, E (=e_{ij}) and F, and as it is involving mod operation, we cannot determine them in any way and break the cipher. Thus the cipher cannot be broken by the known plain text attack, even when r = 1. In this analysis, as we have taken r = 16, it is simply impossible to break the cipher by the known plaintext attack.

On inspecting the equations governing the encryption process (see the encryption algorithm given in section 2), we find that it is not at all possible to choose either the plaintext or the ciphertext in any manner and proceed to the attack. Thus the cipher cannot be broken by the last two attacks also.

In the light of the above discussion, we conclude that this cipher cannot be broken by any attack.

5. COMPUTATIONS AND CONCLUSIONS

In the development of this cipher, we have included two key matrices, namely, E and F. In view of this fact, the strength of the cipher has increased significantly.

The programs required for encryption and decryption are written in Java.

Now, we have divided the entire plain text given in (3.1) into 21 blocks, wherein each block is containing 16 characters. However, in the last block as we have only three characters we have included thirteen 0s as additional characters. On carrying out encryption for all these blocks (excluding the first block for which the cipher text is already obtained), we have obtained the cipher text in the form given below in (5.1).

From the above analysis, presented in this paper, we have seen that the strength of the cipher has increased tremendously−one, on account of the additional key matrix F linked with xor operation, and the other is, on account of the function Mix(). This cipher can be applied to transmit any secret information in a secured manner.

6. REFERENCES

[1] Lester Hill, (1929), "Cryptography in an algebraic alphabet", (V.36 (6), pp. 306-312.), American Mathematical Monthly.

[2] Arthur C. Clarke's Venus Prime, volume 2: Maelstrom. New York. Avon Books, 1988.

[3] Fiestal H., Cryptography and Computer Privacy, Scientific American, May 1973.

[4] National Bureau of Standards NBS FIPS PUB 46 "Data Encryption Standard (DES)", US Department of Commerce, January 1977.

[5] Daemen J., Rijman V., "Rijndael, The Advanced Encryption Standard (AES)", Dr. Dobb's Journal, vol. 26, No. 3, March 2001, pp. 137-139

[6] V.U.K. Sastry , K. Shirisha, "A Novel Block Cipher Involving a Key Bunch Matrix" sent for publication.

[7] William Stallings: Cryptography and Network Security: Principle and Practices", Third Edition 2003, Chapter 2, pp. 29.

146	85	176	127	44	8	222	197	58	196	110	214	30	171	207	198
120	187	132	106	68	199	27	14	126	9	74	249	106	0	2	19
182	241	19	132	43	14	169	232	120	35	206	114	63	13	50	202
110	154	217	93	161	134	178	154	130	97	191	84	149	226	215	25
239	87	246	216	26	61	54	28	236	203	37	53	157	10	209	0
141	171	73	145	50	232	76	97	150	28	80	136	148	9	168	61
12	19	129	104	129	181	14	116	219	69	228	214	132	80	96	6
143	169	36	59	47	82	201	194	223	2	196	5	76	153	111	81
87	147	202	134	209	104	59	222	92	221	246	197	172	2	95	22
65	58	232	27	232	148	88	26	136	88	113	28	37	84	113	185
63	166	101	245	252	238	93	173	158	24	106	61	240	195	86	248
211	167	235	57	66	25	48	237	49	198	93	127	102	218	74	204
184	228	68	145	94	49	44	69	47	10	55	191	10	247	164	83
247	153	215	192	146	13	184	220	154	76	45	179	122	156	189	11
224	162	184	138	181	64	95	114	163	80	149	40	20	4	74	104
211	120	161	80	198	87	0	45	36	125	70	135	184	85	246	182
76	239	171	32	144	11	161	4	175	78	254	187	227	233	40	60
152	155	187	85	134	26	238	13	88	137	54	2	192	105	98	9
135	123	114	116	185	124	152	102	129	209	116	115	137	220	164	42
196	184	110	196	235	140	95	113	74	151	159	231	209	35	220	156

(5.1)

7. 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 14 PhDs, and published more than 86 research papers in various international journals. He received the best Engineering College Faculty Award in Computer Science and Engineering for the year 2008 from the Indian Society for Technical Education (AP Chapter), Best Teacher Award by Lions Clubs International, Hyderabad Elite, in 2012, and Cognizant-Sreenidhi Best faculty award for the year 2012. His research interests are Network Security & Cryptography, Image Processing, Data Mining and Genetic Algorithms.

K. Shirisha is currently working as Associate Professor in the Department of Computer Science and Engineering (CSE), SreeNidhi Institute of Science & Technology (SNIST), Hyderabad, India, since February 2007. She is pursuing her Ph.D. Her research interests are Data Mining and Information Security.