A Block Cipher Involving a Key Bunch Matrix and Including another Key Matrix Supported With Modular Arithmetic Addition

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, by using a key bunch matrix and an additional key matrix. In order to carry out the decryption process, the decryption key bunch matrix is obtained, basing upon the encryption key bunch matrix, on using the concept of multiplicative inverse. In the cryptanalysis, we have found that, this cipher is a very strong one as it includes the additional key matrix and supported with modular arithmetic addition. From the viewpoint of efficiency and strength, this cipher is quite comparable with any other cipher available in the literature of cryptography.

Keywords
encryption key bunch matrix, decryption key bunch matrix, additional key matrix, cryptanalysis, avalanche effect.

1. INTRODUCTION
Transmission of information through internet has been a fascinating area of research, as every secret information is to be maintained in a secured manner. In a recent development [1-2], we have developed a novel block cipher by including a key bunch matrix for encryption, and extended the analysis by introducing another key matrix supplemented with xor operation. In this analysis the decryption matrix is obtained from the given encryption matrix by using the concept of multiplicative inverse.

In the present paper, our objective is to develop a block cipher by using a key bunch matrix and introducing another key matrix by associating it with modular arithmetic addition. This cipher is expected to be very strong as we have a pair of keys in this analysis and the operations are supported by modular arithmetic addition.

The basic equation governing the encryption of a cipher is given by

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

The corresponding equation describing the decryption process is given by

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

Here, $P = [p_{ij}]$ is the plaintext, $C = [c_{ij}]$, the ciphertext, $F=[f_{ij}]$ is an additional key matrix, whose elements are in [0-255]. The $[e_{ij}]$ are the elements of the encryption key bunch matrix E, and $[d_{ij}]$ are the elements of the decryption key bunch matrix D. It is to be noted here that the $[e_{ij}]$ and the $[d_{ij}]$ are odd integers lying in the interval [1,255] and they are governed by the relation

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

For every given e_{ij} we have a unique d_{ij}. Here our interest is to see, how the additional key F and modular arithmetic addition will add to the strength of the cipher.

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

2. DEVELOPMENT OF THE CIPHER
We consider a plaintext P. On using the EBCDIC code, this can be written in the form

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

The encryption key bunch matrix E is given by

$$E = [e_{ij}],\ i=1 \text{ to } n,\ j = 1 \text{ to } n. \quad (2.2)$$

Let us have the additional key matrix F in the form

$$F = [f_{ij}],\ i=1 \text{ to } n,\ j = 1 \text{ to } n. \quad (2.3)$$

We take the ciphertext C in the form

$$C = [c_{ij}],\ i=1 \text{ to } n,\ j = 1 \text{ to } n. \quad (2.4)$$

Let us take the decryption matrix D in the form

$$D = [d_{ij}],\ i=1 \text{ to } n,\ j = 1 \text{ to } n. \quad (2.5)$$

The flowcharts for the encryption and decryption can be drawn in the form shown in figure 1 and figure 2, respectively.

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

Algorithm for Encryption

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

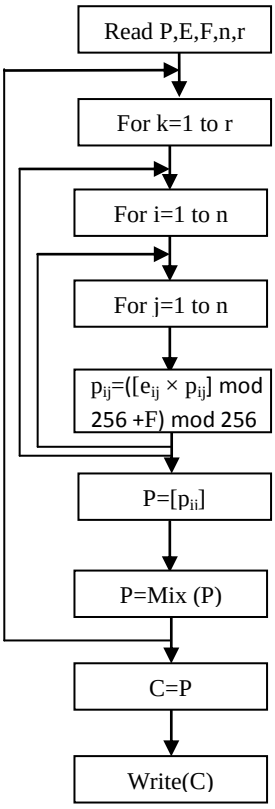

Figure 1. Flowchart for Encryption

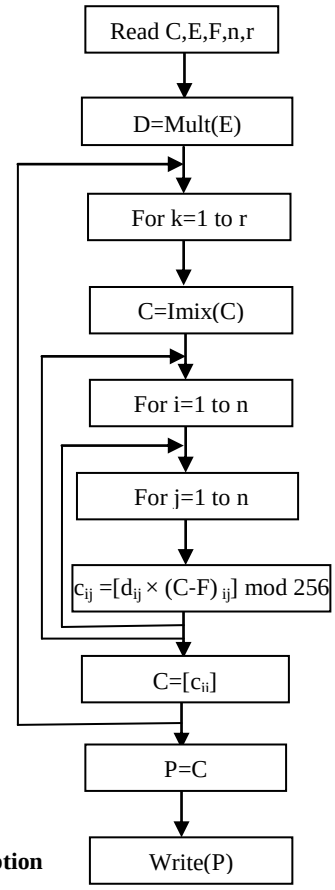

Figure 2. Flowchart for Decryption

5. $p_{ij} = ([e_{ij} \times p_{ij}] \bmod 256 + f_{ij}) \bmod 256$
   }
   }
6. P=[p_{ij}]
7. P=Mix (P)
   }
8. C = P
9. Write(C)

Algorithm for Decryption

1. Read C,E,F,n,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. $c_{ij} = [d_{ij} \times (c_{ij} - f_{ij})] \bmod 256$
   }
   }
8. C=[c_{ij}]
   }
9. P=C
10. Write(P)

The number of rounds in the iteration process of this analysis are taken as r = 16.

In the procedure for encryption, we have made use of the function Mix(). The basic ideas underlying in this function can be outlined as follows.

Let P = [p_{ij}], i=1 to n, j = 1 to n, be the plaintext in any round of the iteration process. Let us assume that n = 2m. Then this matrix can be viewed as two sub-matrices, of equal size, wherein the first one is containing n rows and m columns, and the second one is also containing n rows and m columns (from (m+1)th column to nth column). Now, we represent all the elements in both the matrices in their binary form. Thus each element is written in terms of eight binary bits. Now, taking the first eight bits in the first column, we write them in the form of a decimal number. We carry out the same process with the subsequent elements of the same column (if n > 8). Then we perform the same operations with the binary bits of the first column of the second sub-matrix. Similarly, the process is repeated with the binary bits of the first and second sub-matrices till we exhaust all the columns. In other words, summarizing all this process, we can say that we are concatenating the binary bits of the first sub-matrix with the binary bits of the first column of the second sub-matrix and then the binary bits of the second column of the first sub-matrix with the second column of the second sub-matrix, and obtaining a string of binary bits. Then taking eight binary bits at each instance from the beginning, we write them in terms of decimal numbers, and arrange them in a square matrix of size n in a row-wise manner. This function Mix() is utilized to create confusion and diffusion. The function Imix(), used in the decryption process, contains all the reverse operations of Mix().

The function Mult(), used in the decryption process, is intended to obtain the decryption key bunch matrix D for a given encryption key bunch matrix E.

3. ILLUSTRATION OF THE CIPHER AND THE AVALANCHE EFFECT

Consider the plaintext given below.

Dear Brother! At the time of partition our grandfather felt that our family migrating to Karachi is the best one as we are expected to be very happy among our own community. Your father felt that he must continue his stay in Hyderabad as our ancestors earned a lot of property there. I know, to-day the size of your family is very large consisting of fifty or sixty members, you cannot come to us. We have to bear this life. Be informing about your welfare as frequently as possible.

We focus out attention on the first 16 characters of the above plaintext. Thus we have

Dear Brother! At

On using EBCDIC code, we get

$$P = \begin{bmatrix} 196 & 133 & 129 & 153 \\ 64 & 194 & 153 & 150 \\ 163 & 136 & 133 & 153 \\ 79 & 64 & 193 & 163 \end{bmatrix} \quad (3.1)$$

Let us take the encryption key bunch matrix E in the form

$$E = \begin{bmatrix} 39 & 29 & 205 & 177 \\ 87 & 35 & 81 & 137 \\ 233 & 119 & 69 & 247 \\ 1 & 59 & 193 & 203 \end{bmatrix} \quad (3.2)$$

We have the additional key matrix F in the form given in (3.3).

$$F = \begin{bmatrix} 111 & 23 & 56 & 80 \\ 66 & 45 & 26 & 99 \\ 23 & 56 & 98 & 33 \\ 22 & 55 & 34 & 78 \end{bmatrix} \quad (3.3)$$

Let us obtain the decryption key bunch matrix D by using the relation (1.3). Thus we have

$$D = \begin{bmatrix} 151 & 53 & 5 & 81 \\ 103 & 139 & 177 & 185 \\ 89 & 247 & 141 & 199 \\ 1 & 243 & 65 & 227 \end{bmatrix} \quad (3.4)$$

On using P, E and F, given by (3.1) to (3.3), and the encryption algorithm given in section 2, we get

$$C = \begin{bmatrix} 77 & 95 & 27 & 148 \\ 246 & 164 & 248 & 153 \\ 122 & 82 & 250 & 38 \\ 95 & 198 & 69 & 147 \end{bmatrix} \quad (3.5)$$

On using the above C, given by (3.5), the decryption key bunch matrix D, given by (3.4), and the decryption algorithm given in section 2, we get back the original plaintext P, given by (3.1).

Now let us examine the avalanche effect. On changing 2nd row 3rd column element of the plaintext P, given by (3.1), from 153 to 152, we get a one bit binary change in the plaintext. On using, the modified plaintext, the key matrices, given by (3.2) and (3.3), and the encryption algorithm, given in section 2, we get the ciphertext C in the form

$$C = \begin{bmatrix} 178 & 114 & 17 & 165 \\ 210 & 238 & 7 & 99 \\ 132 & 207 & 17 & 216 \\ 229 & 178 & 127 & 27 \end{bmatrix} \quad (3.6)$$

On comparing (3.5) and (3.6), after representing them in their binary form, we notice that these two ciphertexts differ by 76 bits out of 128 bits.

Now, let us have a one bit change in the encryption key bunch matrix E. To this end, we change the 1st row 3rd column element of (3.2), from 205 to 201. On using this modified E, the plaintext P, given by (3.1), the additional key matrix F, given by (3.3), and using the encryption algorithm, we get the ciphertext C in the form

$$C = \begin{bmatrix} 26 & 232 & 4 & 12 \\ 72 & 45 & 68 & 161 \\ 16 & 8 & 20 & 202 \\ 3 & 94 & 183 & 187 \end{bmatrix} \quad (3.7)$$

On comparing, (3.7) and (3.5), in their binary form, we find that they differ by 69 bits out of 128 bits.

From the above analysis, we conclude that the avalanche effect is conspicuous.

4. CRYPTANALYSIS

In the development of every cipher, cryptanalysis play a fundamental role as it decides whether a cipher is having strength or not. The different types of attacks that are carried out in cryptanalysis are

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

Generally every cipher is designed [3] so that it sustains the first two attacks.

Let us firstly analyze the ciphertext only attack. In this analysis, the key bunch matrix E is containing only odd integers lying in [1-255], and the additional key matrix F is containing integers lying in [0-255]. In the light of this fact, the size of the key space is

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

If we assume that the time required for the computation of this cipher with one E and one F in the key space is 10^{-7} seconds, then the time required for the execution 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 3.12×10^{57} years.

As this time is very large, it is not at all possible to break this cipher by the ciphertext only attack.

Let us now examine the known plaintext attack. In order to carry out this attack, the attacker knows as many plaintexts as he wants, and the corresponding ciphertexts. Having this entire bunch, his job is to break this cipher if possible.

If we restrict our attention to the first round of iteration process, (that is r = 1) the equations governing the encryption process (see the algorithm in section 2) are given by

$$P=([e_{ij} \times p_{ij}] \bmod 256 + F) \bmod 256, \; i = 1 \text{ to } n, \; j = 1 \text{ to } n, \quad (4.1)$$

$$P = Mix(P), \quad (4.2)$$

and

$$C = P \quad (4.3)$$

In this system of equations, the p_{ij} occurring in (4.1), and the ciphertext in (4.3) are known to us. As C is known, the P occurring left hand side of (4.2) is available. On using Imix() operation, the P occurring on the right hand side of (4.2) and hence the P occurring on the left hand side of (4.1) can be determined. As the equation (4.1) contains the additional key matrix F, the e_{ij} cannot be determined by any means. Hence this cipher cannot be broken by this attack. This is the situation when r=1. Here in this analysis, as we have taken r=16, we can say very firmly that this cipher cannot be broken by the known plaintext attack.

Intuitively, we notice that, we cannot choose either a plaintext or a ciphertext in a convenient manner to attack this cipher.

5. COMPUTATIONS AND CONCLUSIONS

In this analysis, we have developed a block cipher by using two key matrices. The first matrix, E, contains a bunch of keys, where in each key is an odd integer which lies in [1-255], while the second one, F, includes integers which lie in [0-255]. The cryptanalysis carried out in this investigation has clearly indicated that this cipher is a strong one and it cannot be broken by any cryptanalytic attack.

The programs for encryption and decryption are written in Java.

The plaintext given in (3.1) is divided into 31 blocks, where in each block is having 16 characters. The last block which contains three characters is made a complete block by

appending thirteen 0s as characters. On using, E and F, and the encryption algorithm, given in section 2, we have obtained the ciphertext for all blocks. Here, we have presented the ciphertext corresponding to all these blocks (excluding the ciphertext of the first block) in (5.1).

The strength of the cipher has increased significantly on account of the introduction of additional key matrix, F, and the modular arithmetic addition operation. This cipher which we have developed in this analysis is quite comparable with any other cipher available in the literature of cryptography.

80	132	150	8	10	19	83	66	214	133	104	238	184	238	205	201
254	32	238	228	255	62	242	131	149	130	202	77	69	133	71	162
10	17	71	12	26	232	49	9	63	51	134	121	52	146	228	117
63	229	53	240	75	218	14	92	41	218	41	197	82	71	79	47
119	32	196	119	52	204	98	136	117	242	72	177	236	203	58	65
169	204	114	92	248	49	94	63	112	127	106	145	249	98	178	49
26	119	187	213	137	196	154	122	217	35	131	210	193	181	29	111
39	208	212	252	121	188	211	123	187	45	48	95	55	0	118	252
182	152	169	54	90	52	163	224	126	118	25	67	215	68	187	68
31	236	218	48	235	8	100	19	39	176	5	19	225	17	244	242
91	36	86	12	160	25	171	75	199	77	175	5	102	4	147	195
25	12	205	102	89	124	206	110	202	16	123	43	75	76	99	37
162	198	119	56	41	216	120	250	104	26	119	236	50	154	18	111
253	177	106	150	139	181	226	226	178	35	80	119	206	235	77	216
50	216	184	99	53	226	254	216	85	91	83	84	122	117	224	127
156	121	130	68	193	60	169	90	40	209	223	123	180	156	235	195
56	166	22	236	175	187	176	149	189	29	126	31	167	10	226	255
231	102	254	48	82	162	214	148	78	196	14	239	246	37	39	86
99	193	223	165	71	176	87	232	136	198	0	30	60	209	27	195
80	129	144	125	236	143	129	78	244	251	243	152	204	38	170	235
104	121	126	62	143	81	0	222	120	191	133	61	149	70	189	224
181	122	33	58	56	103	211	157	22	252	126	71	24	238	187	123
60	154	198	228	95	137	253	77	62	2	6	143	191	107	86	132
249	168	197	100	213	109	21	27	163	118	150	49	159	181	42	71
155	130	67	50	222	171	67	115	94	9	12	36	227	247	196	240
78	189	201	242	54	222	201	128	8	89	112	168	104	136	103	58
104	172	255	35	15	158	233	131	77	177	109	136	10	67	89	179
223	174	129	108	59	2	42	51	240	178	86	119	217	244	55	94
30	18	49	239	235	23	214	23	98	94	59	87	216	23	132	98
169	5	8	10	170	196	243	229	127	155	185	162	196	142	247	72

(5.1)

6. REFERENCES

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

[2] V.U.K. Sastry, K. Shirisha, "A Block Cipher Involving a Key Bunch Matrix and Including another Key Matrix Supplemented with Xor Operation" sent for publication.

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

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.