A Novel Block Cipher Involving a Key Bunch Matrix

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 novel block cipher, which involves a key bunch matrix in the process of encryption. In order to carry out the decryption process, we have obtained the multiplicative inverse of each key in the encryption key bunch matrix by using the concept of multiplicative inverse, and constructed the decryption matrix. In this analysis, the cryptanalysis clearly shows that the strength of the cipher is remarkable, and this cipher can be used for the transmission of information, like any other well-known cipher, through internet.

Keywords

Key bunch matrix, encryption, decryption, avalanche effect, cryptanalysis.

1. INTRODUCTION

Cryptography is a well-known branch of Computer Science. Transmission of information concerned to an organization or a person, in a secured manner, can be achieved in a successful way, by designing a cipher. There are several classical ciphers, such as Hill Cipher [1], PlayFair Cipher [2], Feistel Cipher [3], DES [4], AES [5], which are well established in the area of cryptography. The Hill Cipher depends upon the modular arithmetic inverse of a key matrix. The PlayFair Cipher is based upon the key and the arrangement of the characters occurring in the alphabet (excluding the characters that are in the key), and a typical set of rules applied for writing the ciphertext corresponding to each pair of characters. The Feistel Cipher forms a strong foundation for the development of a number of block ciphers. In this, the plaintext string is divided into two halves. In each round of the iteration process, the right half, operated by the key, is xored with the left half. Then the left and right halves are interchanged for achieving confusion in a thorough manner. DES and AES are the subsequent developments, came into existence in the literature, basing upon Feistel Cipher. In the last one decade, several modifications/ extensions [6-20] of the afore mentioned ciphers have appeared in the literature.

In the present investigation, our objective is to develop a block cipher, which involves several keys that can be represented for convenience in the form of a matrix, called a key bunch matrix. In this analysis each plaintext character is multiplied by a key. In order to carry out the decryption process, the multiplicative inverse of each key is separately obtained. In the development of this cipher, we have adopted an iterative procedure, and the multiplication of the plaintext components and the keys is carried out in each round of the iteration process. Here our interest is to see, how the bunch of keys would influence the cipher and strengthen the cipher.

In what follows, we present the plan of the paper. In section 2, we deal with the development of the cipher. Here we present flowcharts and algorithms required in the analysis. Section 3 contains an illustration of the cipher. In this, we discuss the avalanche effect, which gives an idea of the strength of the cipher. In section 4, we examine the cryptanalysis. Finally in section 5, we put forth the computations carried out in this analysis, and draw conclusions.

2. DEVELOPMENT OF THE CIPHER

Consider a plaintext P. On using EBCDIC code, this can be written in the form of a square matrix, given by

$$P = [p_{ij}],\ i=1 \text{ to } n,\ j=1 \text{ to } n, \tag{2.1}$$

where each p_{ij} lies in [0, 255].

Let E be the key bunch matrix for encryption. Let us suppose that, this can be written in the form

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

Let D be the decryption matrix that can be written in the form

$$D = [d_{ij}],\ i=1 \text{ to } n,\ j=1 \text{ to } n. \tag{2.3}$$

Here, for every given e_{ij} , we can obtain the corresponding d_{ij} by using the relation

$$(e_{ij} \times d_{ij}) \bmod 256 = 1. \tag{2.4}$$

In view of the relation (2.4), it is to be noted that each e_{ij} is to be selected as an odd number, which lies in the interval [1, 255]. Correspondingly, we get each d_{ij} as an odd number lying in the interval [1, 255].

When E = [16(i-1) +2j-1], i=1 to 16 and j= 1 to 8, (2.5)

we can readily find the decryption matrix D in the form

$$D = \begin{bmatrix} 1 & 171 & 205 & 183 & 57 & 163 & 197 & 239 \\ 241 & 27 & 61 & 167 & 41 & 19 & 53 & 223 \\ 225 & 139 & 173 & 151 & 25 & 131 & 165 & 207 \\ 209 & 251 & 29 & 135 & 9 & 243 & 21 & 191 \\ 193 & 107 & 141 & 119 & 249 & 99 & 133 & 175 \\ 177 & 219 & 253 & 103 & 233 & 211 & 245 & 159 \\ 161 & 75 & 109 & 87 & 217 & 67 & 101 & 143 \\ 145 & 187 & 221 & 71 & 201 & 179 & 213 & 127 \\ 129 & 43 & 77 & 55 & 185 & 35 & 69 & 111 \\ 113 & 155 & 189 & 39 & 169 & 147 & 181 & 95 \\ 97 & 11 & 45 & 23 & 153 & 3 & 37 & 79 \\ 81 & 123 & 157 & 7 & 137 & 115 & 149 & 63 \\ 65 & 235 & 13 & 247 & 121 & 227 & 5 & 47 \\ 49 & 91 & 125 & 231 & 105 & 83 & 117 & 31 \\ 33 & 203 & 237 & 215 & 89 & 195 & 229 & 15 \\ 17 & 59 & 93 & 199 & 73 & 51 & 85 & 255 \end{bmatrix} \tag{2.6}$$

On carrying out the encryption, we get the cipher text C in the form,

$$C = [c_{ij}] = [e_{ij} \times p_{ij}] \bmod 256, \tag{2.7}$$

where i = 1 to n, j = 1 to n.

Now applying the decryption process, we get

$$P = [p_{ij}] = [d_{ij} \times c_{ij}] \bmod 256. \quad (2.8)$$

The flowcharts for the encryption process and the decryption process can be drawn in the form given below

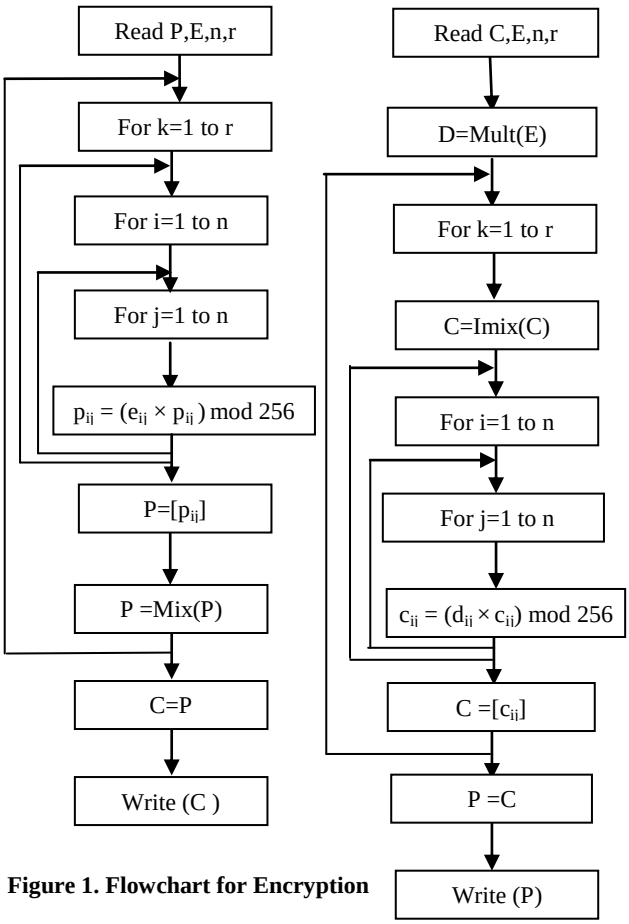

Figure 1. Flowchart for Encryption

Figure 2. Flowchart for Decryption

Here r denotes the number of rounds in the iteration process. Mult() is a function to obtain the decryption key bunch matrix D for the given E.

The algorithms corresponding to the above flowcharts can be written as follows.

Algorithm for Encryption

1. Read P,E,n,r
2. for k = 1 to r do
   {
3. For i=1 to n do
   {
4. For j=1 to n do
   {
5. $p_{ij} = (e_{ij} \times p_{ij}) \bmod 256$
   }
   }
6. P=[p_{ij}]
7. P= Mix (P)
   }
8. C = P
9. Write(C)

Algorithm for Decryption

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

In the afore mentioned flowcharts and algorithms, we have used the function Mix() for mixing the binary bits of the plaintext, in each round of the iteration process, so that thorough confusion and diffusion are created for strengthening the cipher. The process involved in the Mix() can be summarized as follows. Let

$$P = [p_{ij}],\ i = 1 \text{ to } n,\ j = 1 \text{ to } n$$

be the plaintext in a round of the iteration process. Let us suppose that n = 2m. Then the matrix P can be written in the form.

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots p_{1m} & p_{1(m+1)} & \cdots & p_{1(n-1)} & p_{1n} \\ p_{21} & p_{22} & \cdots p_{2m} & p_{2(m+1)} & \cdots & p_{2(n-1)} & p_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ p_{n1} & p_{n2} & \cdots p_{nm} & p_{n(m+1)} & \cdots & p_{n(n-1)} & p_{nn} \end{bmatrix}$$

Let us write each element of this matrix P in its binary form. Thus we have a matrix consisting of n rows and 8n columns. This is given by (2.7).

Now focusing our attention on the first column, we consider the first eight bits of this column and write it in the form of a decimal number. Then we write the next eight bits of the same column (if n > 8) as the second decimal number. We follow this procedure and write the subsequent elements of this column under consideration in terms of decimal numbers. After this, we consider the (m+1)th column of the above matrix and write the decimal numbers as we have done earlier in the case of the first column. Then we proceed to the second column of this matrix and do in the same manner. After this we take up the next column in the second half (i.e., (m+2)th column) of the matrix. We arrange all these numbers one after another in a row-wise manner in a matrix. However, if the plaintext is containing less than eight rows, we consider the elements in the first column of the first half of the matrix and the elements of the first column in the second half of the matrix and form a decimal number. Then we write these decimal numbers in a row-wise manner one after the other and obtain a matrix of size n x n. In this way, we have mixed the elements of the plaintext in a thorough manner. The function Imix(), used in the decryption process, denotes the reverse process of Mix().

$$\begin{bmatrix} p_{111}p_{112}..p_{118} & \cdots & p_{1m1}p_{1m2}..p_{1m8} & p_{1(m+1)1}p_{1(m+1)2}..p_{1(m+1)8} & \cdots & p_{1n1}p_{1n2}..p_{1n8} \\ p_{211}p_{212}..p_{218} & \cdots & p_{2m1}p_{2m2}..p_{2m8} & p_{2(m+1)1}p_{2(m+1)2}..p_{2(m+1)8} & \cdots & p_{2n1}p_{2n2}..p_{2n8} \\ . & \cdots & . & . & \cdots & . \\ . & \cdots & . & . & \cdots & . \\ p_{n11}p_{n12}..p_{n18} & \cdots & p_{nm1}p_{nm2}..p_{nm8} & p_{n(m+1)1}p_{n(m+1)2}..p_{n(m+1)8} & \cdots & p_{nn1}p_{nn2}..p_{nn8} \end{bmatrix} \quad (2.7)$$

3. ILLUSTRATION OF THE CIPHER AND THE AVALANCHE EFFECT

Consider the plaintext P given below.

Dear Husband! I thought that I would have an excellent life by doing IAS. I did! As you are an IPS officer you are getting transferred from one place to another, very frequently, and I was also having the same situation earlier. Now after becoming a secretary in the state, I do not know how I am to act. The bossism of this minister or that minister is causing my life a hell. Be writing letters as frequently as possible so that I feel the thrill of your letter. Yours. (3.1)

Now let us focus our attention on the first 16 characters. This is given by

Dear Husband! I (3.2)

On using the EBCDIC code, we get

$$P = \begin{bmatrix} 196 & 133 & 129 & 153 \\ 64 & 200 & 164 & 162 \\ 130 & 129 & 149 & 132 \\ 79 & 64 & 201 & 64 \end{bmatrix} \quad (3.3)$$

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

$$E = \begin{bmatrix} 13 & 157 & 101 & 125 \\ 89 & 135 & 41 & 39 \\ 107 & 119 & 73 & 131 \\ 237 & 241 & 225 & 253 \end{bmatrix} \quad (3.4)$$

On using the basic concept of multiplicative inverse, given by (2.4), we get the corresponding decryption key bunch matrix, D in the form

$$D = \begin{bmatrix} 197 & 181 & 109 & 213 \\ 233 & 55 & 25 & 151 \\ 67 & 71 & 249 & 43 \\ 229 & 17 & 33 & 85 \end{bmatrix} \quad (3.5)$$

On using (3.3) and (3.4), and applying the process of encryption, given in section 2, we get the cipher text C in the form

$$C = \begin{bmatrix} 152 & 5 & 222 & 237 \\ 191 & 40 & 41 & 17 \\ 158 & 5 & 5 & 74 \\ 14 & 66 & 1 & 8 \end{bmatrix} \quad (3.6)$$

On using (3.6) and (3.5), and applying the process of decryption, given in section 2, we get back the original plaintext P, given by (3.3).

Now, let us examine the avalanche effect. We change the 4th row, 3rd column element of (3.3) from 201 to 193, Thus, we have one binary bit change in the plaintext P. On using the modified plaintext and the encryption key bunch E, given by (3.4), and applying the encryption algorithm, we get the cipher text C in the form

$$C = \begin{bmatrix} 66 & 20 & 45 & 23 \\ 36 & 82 & 159 & 135 \\ 100 & 111 & 223 & 82 \\ 180 & 150 & 235 & 170 \end{bmatrix} \quad (3.7)$$

On comparing (3.6) and (3.7), in their binary form, we find that these two ciphertexts differ by 72 bits (out of 128). This shows that the cipher is expected to be very good.

Now, on changing 4th row, 2nd column element of the key bunch E given by (3.4) from 241 to 240, we have one binary bit change. On using the modified encryption key bunch, the plaintext P, given by (3.3), and using the encryption algorithm, we get the ciphertext C in the form

$$C = \begin{bmatrix} 67 & 3 & 134 & 120 \\ 213 & 101 & 245 & 134 \\ 37 & 59 & 242 & 235 \\ 64 & 47 & 8 & 100 \end{bmatrix} \quad (3.8)$$

On comparing (3.6) and (3.8), after converting them in to their binary form, we find that the two ciphertexts under consideration differ by 69 bits out of 128 bits. This also shows that the cipher is a strong one.

4. CRYPTANALYSIS

In the development of every block cipher, cryptanalysis plays a vital role, as this decides the strength of the cipher and utility of the cipher. The different types of attacks that are available 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 cipher is designed so that it sustains the first two attacks. Theoretical proofs are offered regarding these two attacks [21]. However intuitive indications are given and decisions are taken in the case of the last two attacks.

Let us now consider the ciphertext only attack. In this, the ciphertext and the algorithm are known to us. In this analysis the size of the key bunch matrix, E is n x n. As each element of E is an odd number lying in [1, 255], it can be selected in 128 ways. Thus the size of the key space is

$$128^{n^2} = 2^{7n^2} = \left(2^{10}\right)^{0.7n^2} \approx 10^{2.1n^2}$$

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

$$\frac{10^{2.1n^2} \times 10^{-7}}{365 \times 24 \times 60 \times 60} = 3.12 \times 10^{2.1n^2 - 15} \ \textit{years}.$$

However, in the present analysis, we have taken n = 4. Thus the time required

$= 3.12 \times 10^{18.6}$ *years.*

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

Let us now examine the known plaintext attack. In order to carry out this attack, we know as many pairs of plaintexts and ciphertexts that we require for this purpose. If we confine our attention to only one round of the iteration process, that is if r = 1, then the equations that we obtain from the encryption algorithm presented in section 2, are

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

$$P = \text{Mix}(P), \quad (4.2)$$

and

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

In the known plaintext attack, we know $[p_{ij}]$, i= 1to n, j=1 to n, occurring in the right hand-side of (4.1), and we also know all the components of the ciphertext C, occurring in the equation (4.3). As C is known to us, we know P which is occurring on the left hand side of (4.2). On operating with function Imix on both the sides of (4.2) we can determine P on the left hand side of (4.1). As P and p_{ij} occurring in (4.1) are known to us, we can determine all e_{ij} by using multiplicative inverse, of course, by taking p_{ij} in appropriate manner. Thus the cipher can be broken when r=1.

Let us know focus our attention on the second round of the iteration process. When r = 2, the equations governing the encryption process are given by

$$P = [e_{ij} \text{ x } p_{ij}] \bmod 256, \; i = 1 \text{ to } n, \; j=1 \text{ to } n, \quad (4.4)$$

$$P = \text{Mix}(P), \quad (4.5)$$

$$P = [e_{ij} \text{ x } p_{ij}] \bmod 256, \; i = 1 \text{ to } n, \; j=1 \text{ to } n, \quad (4.6)$$

$$P = \text{Mix}(P), \quad (4.7)$$

$$C = P \quad (4.8)$$

Here, as C is known to us, we can determine P occurring on the right hand side of (4.8). On using this P, we can find out the P occurring on the right hand side of (4.7) by using Imix() on both the sides. Thus, the P on the left hand side of (4.6) is known to us. We also know the p_{ij} occurring on the right hand side of (4.4) as the plaintext is known. Though this is known to us, we cannot determine P, occurring on the left hand side of (4.4). Hence we cannot proceed further to find the e_{ij} (the elements of the key bunch matrix E). Thus, we cannot break this cipher in the case of the known plaintext attack, when r = 2. Here, in our analysis as we have taken r = 16, it is simply impossible to break the cipher by the known plaintext attack.

Even on using fully our intuition and making a thorough effort, we do not find any scope to choose a plaintext / ciphertext which will enable us to break the cipher.

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

5. COMPUTATIONS AND CONCLUSIONS

In this investigation, we have developed a block cipher by using a bunch of keys and their multiplicative inverses. From the cryptanalysis, we have found that the cipher is a strong one, as the keys are affecting the plaintext in each round of the iteration process.

The programs for encryption and decryption are written in Java.

The plaintext given by (3.1) is divided into thirty blocks. As the last block is containing 7 characters, we have appended 9 0s as additional characters to make it a complete block. On using the key bunch E, given by (3.4), and the encryption algorithm given in section 2, we have obtained the ciphertext corresponding to each one of the blocks. Thus we have the ciphertext for the entire plaintext (excluding the first block for which the ciphertext is already given in (3.7)) in the form shown in (5.1).

In this analysis, as each character is multiplied by a key in each round of the iteration process, the plaintext has undergone several transformations, and has resulted in a ciphertext that cannot be deciphered in any way. This is a simple interesting cipher that has some analogy with the classical Hill Cipher [1]. And this cipher cannot be broken by any cryptanalytic attack.

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, S. Udaya Kumar, and A. Vinay Babu, "A large Block Cipher using Modular Arithmetic Inverse of a Key Matrix and mixing of the Key Matrix and the Plaintext", Journal of Computer Science, 2(9), 2006, New York, pp. 690-697.

[7] S. Udaya Kumar, V. U. K. Sastry and A. Vinay Babu, "An iterative Process Involving Interlacing and Decomposition in the Development of a block Cipher", International journal of Computer Science and Network Security, vol. 6, No. 10, October 2006, Seoul, South Korea, pp. 236-245.

[8] V. U. K. Sastry, V. Janaki, "On the modular arithmetic Inverse in the cryptology of Hill Cipher", Proceedings of North American Technology and Business Conference, September 2005, Canada.

[9] V. U. K. Sastry, V. Janaki, "A block Cipher using linear Congruences", accepted for publication in Journal of Computer Science, Science publications, Newcity, New York.

[10] V. U. K. Sastry, N. Ravi Shankar, "Modified Hill Cipher with Interlacing and Iteration ", Journal of Computer Science, Science Publications, 3(11):854-859, 2007.

[11] V. U. K. Sastry, N. Ravi Shankar, "Modified Hill Cipher for a large block of plaintext with Interlacing and Iteration", Journal of Computer Science, Science Publications, 4(1):15-20, 2008.

[12] V. U. K. Sastry, Prof. D.S.R. Murthy, Dr. S. Durga Bhavani, "A Block Cipher Invloving a Key Applied on both sides of the plaintext", International Journal of

Computer and Network Security (IJCNS), Vol. 1, No.1, pp. 27-30, Oct 2009.

[13] V. U. K. Sastry, Prof. D.S.R. Murthy, Dr. S. Durga Bhavani, "A modified Fiestal Cipher involving Modular Arithmetic and a Key on both sides of the Plaintext Matrix", International Journal of Computational Intelligence and Information Security (IJCIIS), Special Issue, Vol. 1, No. 4, pp. 10-16, Jun 2010.

[14] V.U.K.Sastry, Aruna Varanasi, " A Modified Hill Cipher Involving Permutation, Iteration and the Key in a Specified Position"(IJCNS) International Journal of Computer and Network Security, Vol. 2, No. 10, pp. 157-162, October 2010.

[15] V.U.K.Sastry, Aruna Varanasi, S.Udaya Kumar, "A Modern Hill Cipher Involving a Permuted Key and Modular Arithmetic Addition Operation", International Journal of Advanced Research in Computer Science Vol.2 No.1,pp.162-165, Jan-Feb 2011.

[16] 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, No.1, pp. 23-31, 2012.

[17] 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, No.1, pp. 32-39, 2012.

[18] VUK Sastry, Ch. Samson, A Generalized Hill Cipher Involving Different Powers of a Key, Mixing and Substitution, International Journal of Advanced Research in Computer Science, July 2012.

[19] VUK Sastry, Ch. Samson, Generalized Hill Cipher Involving Multiple Keys, Mixing and Key Dependent Substitution, International Journal of Computational Intelligence and Information Security, July 2012.

[20] V.U.K Sastry and K. Anup Kumar, "A Modified Feistel Cipher Involving a Pair of Key Matrices, Supplemented with XOR Operation, and Blending 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. 31333141, 2012.

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

220	140	121	94	123	60	62	130	152	54	27	180	27	207	1	110
149	88	67	94	248	131	26	103	15	183	46	132	101	57	132	7
198	214	186	0	36	21	105	93	160	129	203	141	248	181	54	140
143	241	18	21	28	55	136	134	204	116	99	255	104	186	82	107
207	39	167	8	124	70	90	194	89	4	64	238	130	110	176	33
18	117	35	161	184	10	132	128	153	250	130	109	99	198	120	161
224	5	13	197	255	229	189	110	33	28	77	108	142	140	20	140
191	24	155	141	192	25	197	14	26	166	31	240	112	30	98	148
88	65	64	233	246	107	148	152	140	76	146	143	208	171	154	71
208	83	162	27	154	225	88	82	107	207	190	25	223	109	180	33
191	207	170	87	5	177	209	184	117	202	188	20	64	66	7	160
221	245	5	198	183	237	38	95	184	156	207	219	14	85	184	106
16	71	8	124	252	77	1	9	177	78	195	252	12	150	86	217
106	53	32	145	40	246	65	130	112	216	70	72	217	233	250	158
207	86	245	19	23	227	114	57	106	109	190	40	50	173	179	193
20	172	23	244	244	87	231	238	249	66	72	76	116	170	1	155
102	56	175	194	170	190	187	25	24	130	68	102	122	126	49	238
49	48	119	192	176	29	44	213	43	9	165	11	234	178	142	197
242	128	217	151	77	114	97	146	200	170	198	74	227	46	164	192
132	204	3	69	197	133	164	76	96	124	177	248	94	246	61	41
49	219	214	72	58	59	237	198	149	91	121	186	227	123	182	245
61	126	202	79	116	122	130	19	148	178	172	111	42	97	93	4
115	157	168	25	89	167	86	137	68	84	243	8	95	139	195	189
251	82	223	68	95	173	180	49	209	197	196	182	2	69	87	42
43	225	232	242	105	195	70	123	118	6	185	153	238	64	34	242
250	77	160	93	166	201	76	10	90	131	230	118	187	213	152	89
116	34	65	61	245	148	184	197	204	32	223	154	249	240	235	21
223	6	208	82	6	93	160	57	192	80	25	78	29	231	182	187
195	30	114	24	202	191	87	108	45	97	129	76	217	0	162	96

(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