ISSN No. 0976-5697

Volume 3, No. 4, July- August 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Generalized Hill Cipher Involving Different Powers of a Key, Mixing and Substitution

V.U.K.Sastry*
Department of computer Science and Engineering, SNIST
Hyderabad, India,
vuksastry@rediffmail.com

Ch. Samson
Department of Information Technology, SNIST
Hyderabad, India,
samchepuri@gmail.com

Abstract: In this paper we have generalized the classical Hill cipher by including certain additional features. In this the plaintext block is divided into several matrices. Here we have found several keys by finding different powers of a single key and using modular arithmetic. Then each plaintext matrix is converted into its corresponding ciphertext matrix. On arranging all these ciphertext matrices into a single matrix, we have got the ciphertext. In this analysis we have made use of mixing and substitution for strengthening the cipher. The cryptanalysis carried out in this investigation clearly indicates that the cipher is a strong one.

Keywords: Plaintext, ciphertext, encryption, generalized Hill cipher, decryption, cryptanalysis, avalanche effect.

I. INTRODUCTION

The study of the Hill cipher [1], which had its origin several decades back, has brought in a revolution, in the recent years, in the development of block ciphers in cryptography. Several authors [2-16] have studied different aspects of this cipher by using a single key, by modifying the key in different ways and by applying more than one key on the plaintext (on both the sides of the plaintext). In addition to multiplication with a key or with a pair of keys, they have introduced several other features such as permutation, mixing and substitution in each round of the iteration process. All these features which are introduced into these investigations create confusion and diffusion and strengthen the cipher significantly.

The basic relations governing the Hill cipher are

$$C = KP \bmod 26 \quad (1.1)$$

and

$$P = K^{-1}C \bmod 26, \quad (1.2)$$

Where P is the plaintext column vector, K the key matrix, C the ciphertext, and K^{-1} is the modular arithmetic inverse of K.

In the present paper, our objective is to develop a block cipher of the form

$$C_i = K_i P_i \bmod N, \quad i=1, 2 \dots s, \quad (1.3)$$

and

$$P_i = [\, K_i \,]^{-1} C_i \bmod N, \quad i=1,2 \dots s, \quad (1.4)$$

where P_i is the ith portion of the plaintext,
K_i the i^{th} power of the key matrix,
C_i the i^{th} portion of the ciphertext, corresponding to P_i, and
s denotes the number of sub matrices of the plaintext.

N is a positive integer chosen appropriately. We take N=256. Here $[\, K_i \,]^{-1}$ is the modular arithmetic inverse of the i^{th} power of K.

In the development of the cipher, we use an iteration process. Here we make use of a function called Compose ( ) for combining portions of the plaintext into a single matrix. Further, we use a pair of functions called Mix( ), and Substitute ( ) for transforming the plaintext (in a thorough manner) before it becomes ciphertext. In the light of these facts, the equations governing the encryption can be written in the form

$$P_i = K_i P_i \bmod N, \quad i=1, 2 \dots s, \quad (1.5)$$

$$P = \text{Compose}(P_i), \quad (1.6)$$

$$P = \text{Mix}(P), \quad (1.7)$$

$$P = \text{Substitute}(P). \quad (1.8)$$

At the end of the iteration process, we get the ciphertext C. The equations governing the decryption can be written in the form

$$C = \text{Isubstitute}(C) \quad (1.9)$$

$$C = \text{Imix}(C) \quad (1.10)$$

$$C_i = \text{Decompose}(C), \quad (1.11)$$

$$C_i = [\, K_i \,]^{-1} C_i \bmod N, \quad i=1, 2 \dots n, \quad (1.12)$$

After carrying out the iteration process, finally we get back P_i, and hence we obtain P. The processes Compose( ), Mix( ) and Substitute( ) are explained later. The functions Isubstitute( ), Imix( ) and Decompose( ) denote the reverse processes of the functions Substitute(), Mix( ) and Compose( ) respectively. Here our interest is to develop a block cipher wherein the size of the plaintext and the size of the ciphertext are quite up to the mark.

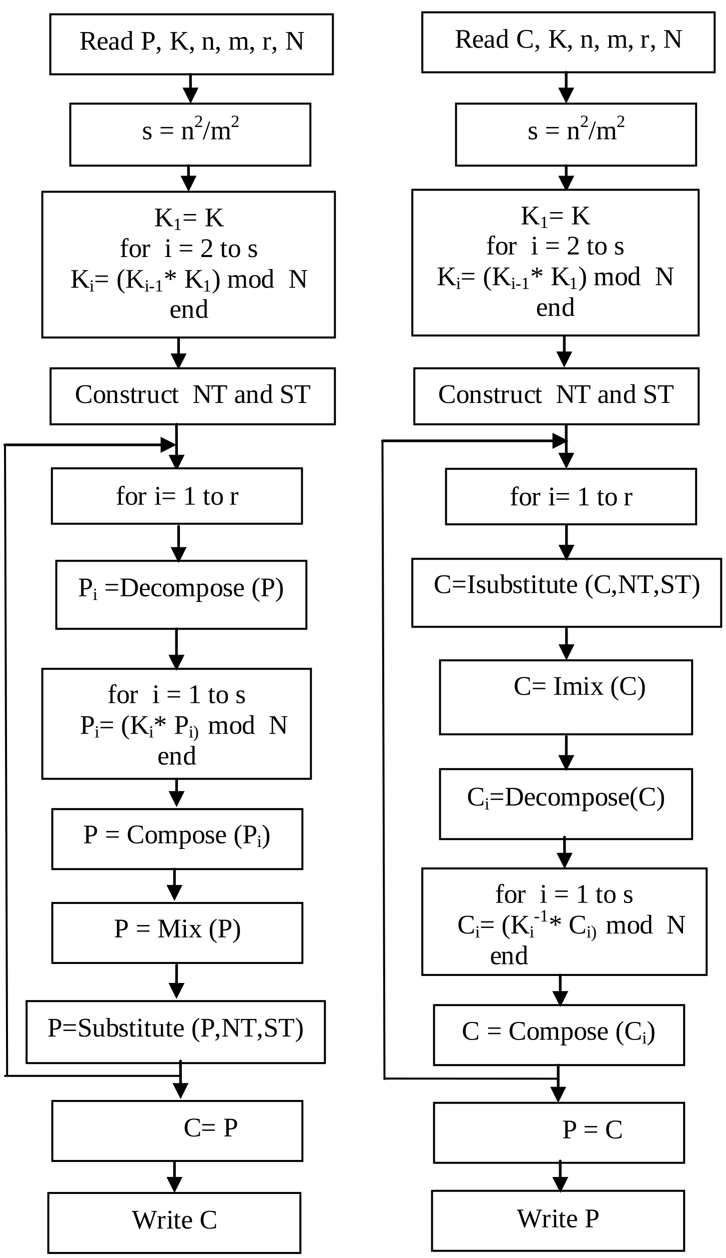
In what follows we mention the plan of the paper. In section 2 we discuss the development of the cipher, and present the flowcharts and algorithms governing encryption and decryption. In section 3, we illustrate the cipher with a suitable example. Here we also study the avalanche effect. Then we deal with the cryptanalysis in section 4. Finally, we discuss the computations carried out in this analysis and draw conclusions from the results.

II. DEVELOPMENT OF THE CIPHER

Consider a plaintext matrix P whose size is nxn. Let us divide this into s sub matrices wherein each sub matrix is a square matrix of size m. This is possible when n is divisible by m. Here we can write $s = n^2/m^2$.

Let us consider a key matrix K whose size is mxm. On applying the encryption process governed by (1.3), we get s ciphertext portions. On placing all these portions in an appropriate manner by using the function Compose ( ), we get a single matrix. Then we apply Mix ( ) and Substitute ( ), in each round of the iteration. Thus we get the final form of the cipertext. On adopting the decryption process, we get back the original plaintext. The details of the functions, Compose ( ), Mix ( ) and Substitute ( ) will be explained in section 3 in which the illustration is given.

The flow charts and algorithms for the encryption and the decryption are given below.



(a) Process of Encryption (b) Process of Decryption

Figure.1 Schematic diagram of the Cipher

NT and ST are a pair of tables which are explained later. Here r denotes the number of rounds in the iteration process and it is taken as 16. Further, we have N=256 as we have used EBCDIC code in the development of the cipher. In this analysis, the number of rounds, denoted by r, is taken as 16.

Algorithm for Encryption

a. Read P,K, n, m, r, N
b. $s= n^2/m^2$
c. $K_1= K$
d. for i = 2 to s
   $K_i= (K_{i-1}* K_1) \bmod N$
   end
e. Construct NT and ST
f. for i = 1 to r
   P_i =Decompose (P)
      for i = 1 to s
         $P_i= (K_i* P_i) \bmod N$
      end
   P = Compose (P_i)
   P= Mix(P)
   P=Substitute (P,NT,ST)
   end
g. C = P
h. Write C

Algorithm for Decryption

a. Read P,K, n, m, r, N
b. $s= n^2/m^2$
a. $K_1= K$
b. for i = 2 to s
   $K_i= (K_{i-1}* K_1) \bmod N$
   end
c. Construct NT and ST
d. for r = 1 to r
   C=Isubstitute (C, NT, ST)
   C= Imix (C)
   C_i=Decompose(C)
   for i = 1 to s
   $C_i= (K_i^{-1}* C_i) \bmod N$
   end
   C = Compose (C_i)
   end
e. P = C
   f. Write P

III. ILLUSTRATION OF THE CIPHER

Consider the plaintext given below.

Daddy! I married the brother-in-law as you insisted. It is very unfortunate! Though I told you several times that I would not go to India, you forced me and made me to come here on account of this marriage relationship. My brother-in-law is beautiful and having a fine personality as you said. But my life has become a hell. He goes to his organization right in the morning by 9 o' clock and comes back home by 12 o' clock late in the night. He does not allow me to go to any job. Though I am well qualified and having my MS degree of America. I am not allowed to try for any employment. He says very firmly that his father and mother, who are quite old must be taken care through all the day. He proclaims that he is an Indian and no Indian can allow his father and mother to stay in the old-age home. Now I want to take divorce and come abroad. Daddy! Do remember, comfort in life must be for both the parties in marriage. I have already contacted a well known lawyer to come out of this problem. Yours daughter... (3.1)

Let us focus our attention on the first 256 characters of the above plaintext. It is given by "Daddy! I married the brother-in-law as you insisted. It is very unfortunate! Though I told you several times that I would not go to India, you forced me and made me to come here on account of this marriage relationship. My brother-in-law is beautiful and h".

On using EBCDIC code, we get the plaintext P in the form

$$P=\begin{pmatrix} 196 & 129 & 132 & 132 & 168 & 90 & 64 & 201 & 64 & 148 & 129 & 153 & 153 & 137 & 133 & 132 \\ 64 & 163 & 136 & 133 & 64 & 130 & 153 & 150 & 163 & 136 & 133 & 153 & 96 & 137 & 149 & 96 \\ 147 & 129 & 166 & 64 & 129 & 162 & 64 & 168 & 150 & 164 & 64 & 137 & 149 & 162 & 137 & 162 \\ 163 & 133 & 132 & 75 & 64 & 201 & 163 & 64 & 137 & 162 & 64 & 165 & 133 & 153 & 168 & 64 \\ 164 & 149 & 134 & 150 & 153 & 163 & 164 & 149 & 129 & 163 & 133 & 90 & 64 & 227 & 136 & 150 \\ 164 & 135 & 136 & 64 & 201 & 64 & 163 & 150 & 147 & 132 & 64 & 168 & 150 & 164 & 64 & 162 \\ 133 & 165 & 133 & 153 & 129 & 147 & 64 & 163 & 137 & 148 & 133 & 162 & 64 & 163 & 136 & 129 \\ 163 & 64 & 201 & 64 & 166 & 150 & 164 & 147 & 132 & 64 & 149 & 150 & 163 & 64 & 135 & 150 \\ 64 & 163 & 150 & 64 & 201 & 149 & 132 & 137 & 129 & 107 & 64 & 168 & 150 & 164 & 64 & 134 \\ 150 & 153 & 131 & 133 & 132 & 64 & 148 & 133 & 64 & 129 & 149 & 132 & 64 & 148 & 129 & 132 \\ 133 & 64 & 148 & 133 & 64 & 163 & 150 & 64 & 131 & 150 & 148 & 133 & 64 & 136 & 133 & 153 \\ 133 & 64 & 150 & 149 & 64 & 129 & 131 & 131 & 150 & 164 & 149 & 163 & 64 & 150 & 134 & 64 \\ 163 & 136 & 137 & 162 & 64 & 148 & 129 & 153 & 153 & 137 & 129 & 135 & 133 & 64 & 153 & 133 \\ 147 & 129 & 163 & 137 & 150 & 149 & 162 & 136 & 137 & 151 & 75 & 64 & 212 & 168 & 64 & 130 \\ 153 & 150 & 163 & 136 & 133 & 153 & 96 & 137 & 149 & 96 & 147 & 129 & 166 & 64 & 137 & 162 \\ 64 & 130 & 133 & 129 & 164 & 163 & 137 & 134 & 164 & 147 & 64 & 129 & 149 & 132 & 64 & 136 \end{pmatrix} \quad (3.2)$$

Let the key matrix K be taken in the form

$$K=\begin{bmatrix} 196 & 224 & 77 & 140 \\ 38 & 25 & 105 & 152 \\ 204 & 5 & 47 & 87 \\ 45 & 69 & 184 & 153 \end{bmatrix} \quad (3.3)$$

On using K_1= K and the relation

$K_i= (K_{i-1}* K_1) \bmod N$ for $i = 2$ to 16, (3.4)

We get K_2 to $K_{16.}$ Let us now explain the procedures involved in the different functions, namely, Compose ( ), Mix ( ) and Substitute ( ) occurring in the encryption process. When

$P_1 = [P_{ij}]$, i=1 to 4 and j= 1 to 4,
$P_2 = [P_{ij}]$, i=1 to 4 and j= 5 to 8,
P3 = $[P_{ij}]$, i=1 to 4 and j= 9 to 12,
P4 = $[P_{ij}]$, i=1 to 4 and j= 13 to 16,
.
.
P13 = $[P_{ij}]$, i=13 to 16and j= 1 to 4,
P14 = $[P_{ij}]$, i=13 to 16and j= 5 to 8,
P15 = $[P_{ij}]$, i=13 to 16and j= 9 to 12,
and P16 = $[P_{ij}]$, i=13 to 16and j= 13to 16.

On arranging these 16 matrices, in a particular way, we get

$$P=\begin{bmatrix} P_1 & P_2 & P_3 & P_4 \\ P_5 & P_6 & P_7 & P_8 \\ P_9 & P_{10} & P_{11} & P_{12} \\ P_{13} & P_{14} & P_{15} & P_{16} \end{bmatrix} \quad (3.5)$$

The process involved here is called Compose ( ).

Thus we get P = $[P_{ij}]$, i=1 to 16 and j= 1 to 16.

In this analysis, the function Mix ( ) is carried out as follows. Here the plaintext P is a square matrix of size 16, and it is of the form

P = $[P_{ij}]$, i=1 to 16 and j= 1 to 16.

This can be written in the form of a matrix having 8 rows and 32 columns. Thus we get the plaintext matrix in the new form given by

P = $[P_{ij}]$, i=1 to 8 and j= 1 to 32. (3.6)

On writing each element of (3.6) in its binary form, we get a matrix having 8 rows and 256 columns . Thus we have

$$P=\begin{bmatrix} P_{111}\,P_{112}\ \ldots\ P_{118}\ \ldots.\ P_{1321}\ P_{1322}\ldots\ P_{1328} \\ P_{211}\,P_{212}\ \ldots\ P_{218}\ \ldots.\ P_{2321}\ P_{1322}\ldots\ P_{2328} \\ . \\ . \\ P_{811}\,P_{812}\ \ldots\ P_{818}\ \ldots.\ P_{8321}\,P_{8322}\ldots\ P_{8328} \end{bmatrix} \quad (3.7)$$

Here P_{111} P_{112} ... P_{118} are the binary bits of $P_{11.}$ In a similar manner, we have the binary bits of the other elements. On taking the 8 binary bits of the first column, we get a decimal number. We call this as the new $P_{11.}$ By considering the second column and performing in the same manner, we get the new $P_{12.}$ On proceeding in a similar way, finally we get the 256^{th} element which will be placed as the new $P_{1616.}$ Thus we have the matrix P, which can be written in the form

P = $[P_{ij}]$, i=1 to 16 and j= 1 to 16. (3.8)

Here it is to be noted that all the elements of (3.8) are obtained by performing Mixing ( ).

Let us now discuss the process of substitution. Consider the numbers 0 to 255. We write them in the form of a matrix containing 16 rows and 16 columns. Let us denote this as NT. This can be written in the form

NT (u, v) =16(u-1) + (v-1), u=1 to 16 and v=1 to 16. (3.9)

On taking all the 16 key matrices, obtained from (3.4), let us form a matrix of size 16x16 corresponding to these keys. In this formation, we take the elements of the first key matrix (of size 4x4) and place them in the first row of the 16x16 matrix, which we are forming, ignoring the numbers which are getting repeated. Similarly we place the elements of the second key matrix in the succeeding positions (taken in the row wise order) of the 16x16 matrix, of course, ignoring the repeated numbers, if any, in the second key matrix. We follow the same procedure with the rest of the key matrices and fill up the 16x16 matrix, partially or fully depending upon repetitions are there or not. However if it is partially filled up, we fill up the rest of the positions with the elements which are not occurring in the 16x16 matrix that we are forming. It may be noted that the rest of the elements with which we are filling up lie in the interval [0, 255].Thus we get a key matrix, say ST, of size 16x16 wherein no repetitions are there.

Let us now consider the plaintext matrix P, which is obtained in a particular round of the iteration process of the encryption, after using Mix ( ). Now let us form the matrix corresponding to the substitution process denoted by Substitute ( ). This can be done by adopting the rule which is given below:

If P (i,j) = NT(u,v)
Then P (i, j) =ST (u, v).

In other words, the above relation can be mentioned as follows. If the i^{th} row j^{th} column element of P is equal to the u^{th} row v^{th} column element of the matrix NT, then the i^{th} row j^{th} column element of the plaintext ,that is P (i,j), is replaced by the u^{th} row v^{th} column element of the matrix ST. Thus we are able to carry out the substitution as we complete the process for i= 1 to 16 and j=1 to 16. As the formation of the substitution table is a simple one, we have avoided the details of this formation for brevity. Now on using the encryption algorithm given in section 2, we get the ciphertext C in the form

C =

97	193	133	234	49	197	9	208	130	240	241	1	43	28	73	228
9	48	0	211	46	18	72	52	170	232	142	139	55	84	173	91
166	66	129	246	157	194	136	243	92	105	82	139	80	39	61	157
152	159	174	12	243	21	151	216	113	188	98	177	25	59	83	40
169	168	68	187	137	209	50	9	55	176	122	77	79	34	98	223
161	113	198	165	81	136	38	85	224	104	244	157	138	223	254	169
64	106	242	49	182	137	161	51	109	216	248	250	189	93	239	12
160	115	144	206	223	23	23	5	93	57	237	228	111	130	41	196
143	100	247	90	184	88	67	22	205	93	205	57	228	32	48	17
195	166	72	103	152	9	163	190	209	111	66	217	253	255	207	180
183	185	247	24	242	161	107	207	156	217	127	117	24	194	78	24
205	187	64	192	131	10	109	171	60	202	16	56	26	192	254	59
172	123	46	63	6	238	43	177	231	201	7	209	106	66	38	225
98	62	23	150	83	235	129	59	54	95	57	92	183	188	33	59
9	238	110	140	5	138	178	225	21	32	212	23	100	107	12	166
230	233	141	126	12	99	249	166	167	194	103	169	159	214	27	179

(3.10)

On carrying out the decryption process, by adopting the decryption algorithm, we get back the original plaintext.

Let us now examine the avalanche effect. On changing the first row tenth column element of (3.2) from 148 to 149, we get a one bit change in the plaintext. On using the modified plaintext, the keys given by (3.3) and (3.4) and applying the encryption algorithm, given in section 2, we get the corresponding ciphertext given by

C =

47	54	157	84	96	54	244	223	86	59	216	253	11	209	39	93
66	221	42	252	68	15	158	19	123	55	103	124	149	206	148	201
239	95	168	23	10	129	65	122	166	70	239	178	119	21	124	147
36	242	58	37	186	35	232	129	26	33	70	135	44	120	38	174
201	159	34	236	140	3	23	90	95	13	215	242	24	101	202	223
242	40	3	76	137	158	173	139	107	120	200	229	146	116	27	252
211	234	13	207	163	120	138	56	236	158	75	244	154	247	189	177
72	139	153	89	214	242	109	89	250	36	99	61	54	66	160	255
186	94	24	177	146	242	161	5	227	16	76	241	43	251	209	248
87	64	1	88	98	142	104	61	95	35	102	118	50	137	73	33
51	218	46	49	237	158	164	202	109	117	81	233	234	57	198	183
41	9	90	139	233	13	252	109	13	230	188	131	84	120	95	230
169	16	246	181	102	27	124	165	169	139	57	128	14	28	77	69
157	73	239	73	142	167	167	190	115	204	14	159	251	192	85	197
81	40	4	251	74	239	107	223	53	170	2	60	195	244	98	221
173	102	219	48	144	31	114	233	138	223	243	116	64	12	83	116

(3.11)

On converting (3.10) and (3.11) into their binary form and comparing them, we notice that they differ by 1074 binary bits out of 2048 bits. This shows that the avalanche effect is quite good.

Let us now consider a one bit change in the key. This is achieved by replacing the first row third column element of the key matrix K, given by (3.3), from 77 to 76. On using the original plaintext, the modified key K (together with the corresponding values K_2 to K_{16}), we get the ciphertext in the form

C =

146	162	39	247	220	145	139	220	255	202	54	153	222	252	207	80
207	201	177	216	100	131	249	53	209	54	211	68	197	199	39	207
5	87	80	235	50	93	66	220	46	114	109	176	88	29	65	36
215	202	231	232	38	121	53	19	173	30	32	251	97	30	12	183
191	45	233	67	1	199	219	170	151	110	159	243	30	104	163	163
116	49	59	84	108	26	156	155	152	169	219	139	52	32	163	219
119	161	174	158	25	26	201	169	69	208	236	136	243	168	21	155
255	110	150	31	172	203	201	172	103	40	181	219	8	35	224	135
18	216	227	245	242	235	142	220	167	174	206	74	84	216	125	152
93	28	231	199	205	236	203	21	252	79	19	75	125	249	41	25
80	44	204	76	101	1	169	219	35	112	62	46	153	159	196	201
71	252	159	34	130	223	203	154	111	223	153	120	251	122	12	146
123	33	188	59	246	69	219	169	63	221	100	185	158	154	207	91
64	56	181	191	62	218	19	156	31	188	147	107	171	51	154	36
92	61	205	31	104	100	38	23	186	223	203	183	127	158	95	21
29	48	150	168	144	61	181	133	148	88	117	144	168	172	63	216

(3.12)

On comparing the ciphertexts (3.10) and (3.12) in their binary form, we find that they differ by 1032 bits out of 2048 bits. This also shows that the avalanche effect is quite significant. In the light of the above analysis, we conclude that the strength of the cipher is expected to be very good.

IV. CRYPTANALYSIS

In the literature of cryptography, it is well known that the strength of a cipher can be determined by carrying out cryptanalysis. The different types of cryptanalytic attacks are as follows.

a. Cipertext only attack (Brute force attack)
b. Known plaintext attack
c. Chosen plaintext attack and
d. Chosen ciphertext attack.

Generally every cipher is to be designed so that it withstands the first two attacks [1].

In this analysis, as the key contains 16 decimal numbers, the size of the key space is
$2^{128} = (2^{10})^{12.8} \approx (10^{3})^{12.8} = 10^{38.4}$.

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

$$\frac{10^{38.4} \times 10^{-7}}{365 \times 24 \times 60 \times 60} = 3.12 \times 10^{38.4} \times 10^{-15} = 3.12 \times 10^{23.4} \ \textit{years}$$

Thus as the time required for the entire computation is very large, we cannot break this cipher by the brute force attack.

Let us now consider the known plaintext attack. In this case, we know as many plaintext and ciphertext pairs as we want for the attack. Basing upon these pairs, we must be able to find a relation which determines the key or a function of the key for breaking the cipher.

To carry out the known plaintext attack, in this investigation, let us consider the example given in the illustration. In this the plaintext P is divided into 16 matrices wherein each one is a square matrix of size 4. Here we have 16 key matrices denoted by $K_1, K_2 \dots K_{16}$. All these matrices are obtained basing upon the key K.

Let us suppose that we carry out only one round in the iteration process. The equations governing this process are

$$P_i = K_i P_i \mod N, \quad i=1, 2 \dots 16, \qquad (4.1)$$
$$P = \text{Compose}(P_i) \qquad (4.2)$$
$$P = \text{Mix}(P) \qquad (4.3)$$
$$P = \text{Substitute}(P, NT, ST) \qquad (4.4)$$
$$C = P \qquad (4.5)$$

From the equation (4.1) and (4.2), we get

$$P = \begin{bmatrix} K_1 P_1 & K_2 P_2 & K_3 P_3 & K_4 P_4 \\ K_5 P_5 & K_6 P_6 & K_7 P_7 & K_8 P_8 \\ K_9 P_9 & K_{10} P_{10} & K_{11} P_{11} & K_{12} P_{12} \\ K_{13} P_{13} & K_{14} P_{14} & K_{15} P_{15} & K_{16} P_{16} \end{bmatrix} \mod N \qquad (4.6)$$

Thus from the equations (4.3) – (4.6), we get a relation of the form

$$C = S(M(F(K_1 P_1, K_2 P_2, \dots K_{15} P_{15}, K_{16} P_{16}, \mod N))) \qquad (4.7)$$

Where M and S are written for Mix( ) and Substitute( ) respectively for elegance. F is used to denote a function having all the entities in (4.6) as variables.

Here as the plaintext and the ciphertext are known to us, we have $P_1, P_2 \dots P_{16}$ and C. From the equation (4.7), we find that it is simply impossible to find K (=K_1) as mod N is there, the Mix( ) function is mixing all the elements in F by converting them into binary bits, and the Substitute( ) is mapping the resulting elements into some other elements. This is the conclusion that we are able to arrive at even by carrying out only one iteration. We cannot say what happens to K after performing all the 16 rounds involved in the iteration process. Thus it is totally impossible to break the cipher by the known plaintext attack.

On looking at the encryption algorithm, we do find that it is not possible to choose, intuitively, a plaintext or a ciphertext and break the cipher. Thus chosen plaintext attack or chosen ciphertext attack cannot be applied in any way.

V. COMPUTATIONS AND CONCLUSIONS

In this paper we have developed a block cipher by generalizing the classical Hill cipher. In this the plaintext is decomposed into a set of matrices. Several key matrices are developed taking a single key and using the modular arithmetic. The corresponding ciphertexts are obtained by applying the relations of the classical Hill cipher. Finally a single ciphertext is generated by arranging the portions of the ciphertext obtained earlier. This process is repeated in the iteration scheme. In each iteration, we have employed two functions namely Mix( ) and Substitute( ) for achieving diffusion and confusion.

Computer programs are developed for encryption and decryption by using MATLAB [17].

The plaintext (3.1) is divided into 4 blocks, wherein each block is having 256 characters, and the last block is appended with 7 blanks so that it becomes a complete block (consisting 256 characters). On using the encryption algorithm, given in section 2, we get the ciphertext corresponding to the entire plaintext (excluding the ciphertext of the first block) in the form

191	101	218	11	115	143	116	99	253	77	248	118	74	145	13	251
110	218	35	226	101	248	208	208	241	114	86	139	255	65	127	162
128	109	85	92	5	18	239	11	199	234	215	46	31	220	191	40
5	83	63	171	227	203	161	231	36	158	220	134	121	211	106	207
5	131	239	77	33	53	195	104	14	178	199	30	89	46	93	180
201	249	161	212	144	58	99	100	117	26	32	140	21	81	246	240
49	63	87	43	153	167	52	222	6	61	120	103	222	147	133	143
148	123	178	220	99	31	53	75	239	175	209	49	90	134	242	165
87	178	95	128	72	53	17	114	254	116	151	197	214	180	144	97
7	9	54	27	191	62	58	154	151	226	236	230	243	176	58	186
246	138	180	123	136	185	38	114	105	170	146	186	90	137	50	161
140	152	62	3	176	225	128	141	199	199	22	171	56	181	170	106
15	194	246	181	238	36	12	255	248	78	35	31	243	32	182	202
81	242	82	51	61	253	100	33	107	12	112	84	133	199	247	251
46	148	16	115	129	161	73	101	95	47	18	86	67	251	256	204
158	158	157	45	28	185	20	192	210	8	27	44	153	20	67	190
66	173	207	215	204	195	47	17	75	245	189	151	206	46	142	101
4	60	16	247	135	96	90	162	189	219	50	116	214	84	155	94
75	247	172	8	118	15	33	201	20	5	102	163	38	90	64	131
196	23	143	106	66	230	82	92	195	187	32	7	139	170	30	171
214	132	194	73	161	150	23	221	31	85	48	98	227	244	227	38
101	42	7	11	244	225	162	75	27	114	232	148	242	32	210	178
184	228	253	175	149	16	51	157	145	73	77	206	131	106	148	242
59	25	115	187	7	76	104	237	94	31	103	209	32	84	124	110
26	184	246	111	4	185	15	175	199	131	153	143	147	208	188	137
83	86	206	75	30	115	68	249	95	107	179	7	219	54	65	136
176	48	22	106	195	144	117	115	224	153	34	249	179	94	121	46
231	54	189	19	5	152	162	126	180	152	175	228	167	95	135	132
63	18	75	169	168	172	112	75	97	86	99	243	46	150	99	211
224	111	161	117	39	4	256	87	33	219	149	142	181	232	83	42
45	113	80	137	193	74	195	39	13	51	119	56	203	209	8	212
14	57	67	234	122	11	81	184	40	160	154	18	75	96	53	233
190	179	55	215	159	197	93	14	81	223	14	131	69	91	57	211
215	207	227	155	146	13	80	157	54	163	159	24	224	171	169	244
213	25	23	138	153	6	109	201	63	104	120	83	166	154	70	218
233	52	97	216	153	6	76	181	251	169	28	3	125	156	110	130

10	230	126	207	53	149	14	27	129	113	95	126	201	197	143	90
191	146	152	84	43	205	36	145	150	253	118	199	17	150	6	142
172	250	109	60	167	92	247	126	124	185	54	200	49	96	24	116
100	61	213	236	22	4	212	177	246	227	214	236	142	117	239	108
96	127	37	149	27	104	74	35	131	138	239	89	153	18	136	59
91	44	202	253	81	134	135	247	226	25	173	155	102	109	197	206
92	63	49	63	88	144	47	134	204	181	41	8	34	151	112	137
146	182	251	196	36	192	159	196	238	187	58	148	21	195	61	194
190	101	235	41	6	59	98	82	214	47	153	68	243	29	172	78
185	204	6	226	25	187	63	100	165	202	91	80	51	74	87	176
63	190	189	118	60	251	224	143	109	201	212	248	206	105	206	181
158	170	75	182	67	39	127	150	164	178	154	85	169	94	156	81

From the cryptanalysis, we have found that the strength of the cipher is remarkable. This has become possible as we have handled portions of the plaintext in arriving at the ciphertext. The inclusion of Mix( ) and Substitute( ) functions really enabled us to enhance the strength of the cipher.

VI. REFERENCES

[1] William Stallings, Cryptography and Network Security, Principles and Practice, Third edition, Pearson, 2003.

[2] S. Udaya Kumar, V.U.K. Sastry, and A. Vinaya babu, "A Block Cipher Basing Upon Permutation, Substitution, and Iteration", "Journal of Information Privacy and Security",3(1), 2007, Ivy League Publishing, P.O. Box 680392, Marietta, GA 30068 USA.

[3] S. Udaya Kumar, V.U.K. Sastry, and A. Vinaya babu, "A Block Cipher using an Iterative Method and the Modular Arithmetic Inverse of a Key Matrix", International Journal of Scientific Computing 1 (1) January – June 2007, pp. 69-78,. Serial Publications, New Delhi, India.

[4] V.U.K. Sastry and V. Janaki, "Modified Hill Cipher with key dependent permutation and circular Rotation", Journal of Computer Science, 3(9):736 – 739, 2007 ISSN 1549 – 3636.

[5] V.U.K. Sastry and V. Janaki, Modified Hill Cipher with Multiple Keys, International journal of Computational Science, 2008, 2(6), pp. 815-826.

[6] 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.

[7] 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

[8] V.U.K.Sastry, N.Ravi Shankar, S.Durga Bhavani, "A Modified Hill Cipher involving Interweaving and Iteration", International Journal of Network Security, 11(2): 51-56, September 2010.

[9] V. U. K. Sastry, D. S. R. Murthy, S. Durga Bhavani, "*A Block Cipher Involving a Key Applied on Both the Sides of the Plain Text*", International Journal of Computer and Network Security (IJCNS), Vol. 1, No.1, pp. 27 – 30, Oct 2009.

[10] 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.

[11] V.U.K.Sastry, Aruna Varanasi, S.Udaya Kumar, "A Modified Hill Cipher Involving a Pair of Keys and a Permutation",(IJCNS) International Journal of Computer and Network Security, Vol. 2, No. 9, pp. 105-108, September 2010.

[12] Dr. V. U. K. Sastry, Prof. D. S. R. Murthy, Dr. S. Durga Bhavani, "*A Block Cipher Having a Key on One Side of the Plain Text Matrix and its Inverse on the Other Side*", International Journal of Computer Theory and Engineering (IJCTE), Vol. 2, No.5, pp. , Oct 2010.

[13] 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.

[14] V.U.K.Sastry, Aruna Varanasi, S.Udaya Kumar, "A Modern Hill Cipher Involving XOR operation and a Permuted Key", International Journal of Advanced Research in Computer Science, Vol.2 No.1, pp.153-155, Jan-Feb 2011.

[15] Aruna Varanasi, V.U.K.Sastry, S.Udaya Kumar, "A Modern Hill Cipher Involving a Pair of Keys, Modular Arithmetic Addition and Substitution", International Journal of Advanced Research in Computer Science, Vol.2 No.3, pp. 460-464, May-June 2011.

[16] Aruna Varanasi, V.U.K.Sastry, S.Udaya Kumar, "A Modern Hill Cipher Involving a Pair of Keys, XOR operation and Substitution", International Journal of Advanced Research in Computer Science, Vol.2 No.3, pp. 496-500, May-June 2011.

[17] http://www.mathworks.com/products/matlab/

About authors

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 70 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), and Cognizant- Sreenidhi Best faculty award for the year 2012. His research interests are Network Security & Cryptography, Image Processing, Data Mining and Genetic Algorithms.

Mr. Ch. Samson obtained his Diploma from Govt Polytechnic, Hyderabad in 1994, B. E. from Osmania University in 1998 and M. E from SRTM University in 2000. Presently he is pursuing Ph.D. from JNTUH, Hyderabad since

2009. He published 10 research papers in various international journals and two papers in conferences. He is currently working as Associate Professor and Associate Head in the Dept. of Information Technology (IT), SNIST since June 2005. His research interests are Image Processing, Image Cryptography and Network Security.