A Block Cipher Involving a Key and a Key Bunch Matrix, Supplemented with Key-Based Permutation and Substitution

Dr. V.U.K. Sastry  
Professor (CSE Dept), Dean (R&D)  
SreeNidhi Institute of Science & Technology, SNIST  
Hyderabad, India

K. Shirisha  
Computer Science & Engineering  
SreeNidhi Institute of Science & Technology, SNIST  
Hyderabad, India

Abstract— In this paper, we have developed a block cipher involving a key and a key bunch matrix. In this cipher, we have made use of key-based permutation and key-based substitution. The cryptanalysis carried out in this investigation, shows very clearly, that this cipher is a very strong one. This is all on account of the confusion and the diffusion created by the permutation, the substitution, in each round of the iteration process.

Keywords— Key; key bunch matrix; encryption; decryption; permutation; substitution; avalanche effect; cryptanalysis

I. INTRODUCTION

The study of the block ciphers [1] is an interesting area of research in cryptography. In the very recent past, we have developed a pair of block ciphers [2-3], which include a key matrix, as it is in the case of the Hill cipher, and a key bunch matrix. In these investigations, we have made use of the concepts of the modular arithmetic inverse and the multiplicative inverse.

In [2], we have made use of function Mix(), which mixes the binary bits in each round of the iteration process, and in [3], we have introduced a function called Permute(), which carries out permutation of binary bits of the plaintext in each round of the iteration process. In these analyses, we have noticed that the key matrix and the key bunch matrix, and the additional function Mix()/ Permute() strengthen the cipher, in a conspicuous manner.

In the present paper, our objective is to develop a block cipher, wherein we use a key matrix together with a key bunch matrix. Here, we have introduced a key-based permutation and a substitution basing upon the key. In this, our interest is to see, how the permutation and the substitution would influence the cipher and enhance the strength of the cipher, due to the confusion and the diffusion arising in this process.

We now mention the plan of the paper. In section 2, we discuss the development of the cipher and introduce the flowcharts and the algorithms required in this analysis. We illustrate the cipher and discuss the avalanche effect in section 3. We study the cryptanalysis in section 4. Finally in section 5, we deal with the computation carried out in this investigation and draw conclusions.

II. DEVELOPMENT OF THE CIPHER

We consider a plain P having n(2) characters and represent it in the form of a square matrix of size n by using EBCDIC code. Thus we have

\[ P = \begin{bmatrix} p_{ij} \end{bmatrix}, \text{i=1 to n, j=1 to n}. \]  

Let the key matrix K be given by

\[ K = \begin{bmatrix} k_{ij} \end{bmatrix}, \text{i=1 to n, j=1 to n}, \]  

wherein each \( k_{ij} \) is an odd number lying in \([1-255]\).

On using the concept of the multiplicative inverse [4], the encryption key bunch matrix E is taken in the form

\[ E = \begin{bmatrix} e_{ij} \end{bmatrix}, \text{i=1 to n, j=1 to n}, \]  

wherein each \( e_{ij} \) is an odd number lying in \([1-255]\).

The basic equations governing the encryption can be written in the form

\[ P = (KP) \mod 256, \]  

\[ P = \begin{bmatrix} p_{ij} \end{bmatrix} \times \begin{bmatrix} k_{ij} \end{bmatrix} \mod 256, \text{i=1 to n, j=1 to n}, \]  

\[ P = \text{Permute}(P), \]  

\[ P = \text{Substitute}(P), \]  

and

\[ C = P. \]  

The corresponding equations of the decryption process are given by

\[ C = \text{ISubstitute}(C), \]  

\[ C = \text{IPermute}(C), \]  

\[ C = \begin{bmatrix} d_{ij} \end{bmatrix} \times \begin{bmatrix} c_{ij} \end{bmatrix} \mod 256, \text{i=1 to n, j=1 to n}, \]  

\[ C = (K(-1) C) \mod 256, \]  

\[ P = C. \]  

The details of the function Permute() and the function Substitute() are explained later. It is to be noted here, that the functions ISubstitute() and IPermute() denote the reverse process of the functions Substitute() and Permute().
The flowcharts depicting the process of the encryption and the decryption are given in Figs. 1 and 2.

The algorithms, for the encryption and the decryption are as follows.

**Algorithm for Encryption**
1. Read P,E,K,n,r
2. For k = 1 to r do
   1. \( P = (KP) \mod 256 \)
3. For i = 1 to n do
   1. For j = 1 to n do
      1. \( p_{ij} = (e_{ij} \times p_{ij}) \mod 256 \)
5. \( P = [p_{ij}] \)
6. \( P = \text{Permute}(P) \)
7. \( P = \text{Substitute}(P) \)
8. \( C = P \)
9. Write(C)

**Algorithm for Decryption**
1. Read C,E,K,n,r
2. \( K^{-1} = \text{Inv}(K) \)
3. \( D = \text{Mult}(E) \)
4. For k = 1 to r do
   1. \( C = \text{ISubstitute}(C) \)
2. \( C = \text{IPermute}(C) \)
3. For i = 1 to n do
   1. For j = 1 to n do
      1. \( c_{ij} = (d_{ij} \times c_{ij}) \mod 256 \)
6. \( C = [c_{ij}] \)
7. \( C = (K^{-1}C) \mod 256 \)
8. \( P = C \)
9. Write(P)
Let us now explain the basic ideas underlying in functions Permute() and Substitute(). Both are dependent on a key. Let us take the key \( K \) in the form

\[
K = \begin{pmatrix}
120 & 182 & 102 & 13 \\
25 & 14 & 16 & 200 \\
30 & 147 & 61 & 122 \\
40 & 127 & 206 & 91
\end{pmatrix}
\] (2.15)

The numbers in this key are listed in the 2\textsuperscript{nd} row of the following table, Table-1.

| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 120| 182| 102| 13 | 25 | 14 | 16 | 200| 30 | 147| 61 | 122| 40 | 127| 206| 91|

Table-1. Relation between serial numbers and the ascending order of the key numbers.

The 1\textsuperscript{st} row of this table contains the serial number, and the 3\textsuperscript{rd} row of this table indicates the ascending order of the numbers in the key, given in the 2\textsuperscript{nd} row.

Consider the plaintext \( P \) in any round of the iteration process. It is possible to see this plaintext as a set of square matrices of size 16 whenever \( n \) is divisible by 16. As the Table-1 is suggesting, we interchange the rows

\[
(1,10), (2,14), (3,9), (5,4), (8,15), (11,7) \text{ and } (13,6).
\] (2.16)

Similarly, it may be done in the case of the columns. It may be noted here, that once we have made an interchange involving a row or a column, we do not do anymore interchange involving that row or column subsequently so that plaintext remains in a systematic manner. This is the basic idea underlying in the function Permute(), when \( n \geq 16 \). On the other hand, when \( n \) takes a value less than 16, for example when \( n = 4 \), then let us see how the process of the permutation will be carried out.

Consider an example when \( n = 4 \). In this case, the plaintext is of the form

\[
P = \begin{pmatrix}
    p_{11} & p_{12} & p_{13} & p_{14} \\
    p_{21} & p_{22} & p_{23} & p_{24} \\
    p_{31} & p_{32} & p_{33} & p_{34} \\
    p_{41} & p_{42} & p_{43} & p_{44}
\end{pmatrix}
\] (2.17)

On representing each element of this matrix in terms of binary bits, in a row-wise manner, we have

This is a matrix having 4 rows and 32 columns. This can be written, for convenience, in the form of another matrix, given by (2.19).

\[
C = \begin{pmatrix}
P_{111}P_{112} \cdots P_{118} & P_{119}P_{120} \cdots P_{128} \\
P_{211}P_{212} \cdots P_{218} & P_{219}P_{220} \cdots P_{228} \\
P_{311}P_{312} \cdots P_{318} & P_{319}P_{320} \cdots P_{328} \\
P_{411}P_{412} \cdots P_{418} & P_{419}P_{420} \cdots P_{428}
\end{pmatrix}
\] (2.19)

10. \( C = [C_{ji}] \)
11. \( C = (K^{-1}C) \mod 256 \)

On noting the correspondence between the matrices, given by (2.20) and (2.21), we can perform the substitution process having the numbers 0 to 255, in some other order. This table includes the number 0 to 255, can be written in the form of another matrix, given by (2.21).

\[
EB(i, j) = [16(i - 1) + j - 1], i = 1 \text{ to } n, \ j = 1 \text{ to } n \quad (2.20)
\]

This has 16 rows and 16 columns. On interchanging the rows first and then the columns next, we get a new matrix, this matrix has 8 rows and 16 columns. In order to carry out permutation, we swap the rows (5,4) as indicated by (2.16). The rest of the rows are untouched, as we do not have the possibility of interchange. Then the columns are interchanged by following the content of (2.16).

Let us now consider the process of substitution, which depends upon the permutation. The EBCDIC code, which includes the number 0 to 255, can be written in the form of a matrix, given by

\[
EB(i, j) = [16(i - 1) + j - 1], i = 1 \text{ to } n, \ j = 1 \text{ to } n \quad (2.20)
\]

This has 16 rows and 16 columns. On interchanging the rows first and then the columns next, we get a new matrix, having the numbers 0 to 255, in some other order. This table can be written in the form, given in (2.21).

On noting the correspondence between the matrices, given by (2.20) and (2.21), we can perform the substitution process in any plaintext. Thus we have the function Substitute().

The function \( \text{Inv()} \) is used to obtain the modular arithmetic inverse of the key matrix \( K \). The function \( \text{Mult()} \) results in the decryption key bunch matrix \( D \) for the given encryption key bunch matrix \( E \). For a thorough understanding of these
functions, we may refer to [2].

In this cipher, \( r \) denotes the number of rounds carried out in the iteration process. Here, we have taken \( r=16 \).

III. ILLUSTRATION OF THE CIPHER AND THE AVALANCHE EFFECT

Consider the plaintext given below.

Dear Brother! With all the training that you had from NCC in your college, having a strong feel that India is our motherland, and it is our responsibility to protect this country from the invasion by other countries, you left us some years back. After that, there are several changes within the country. When you were leaving us, we had only few parties such as Congress, Communist and BJP. Today, parties have grown as mushrooms and the number of parties is that many. We do not know, in what way unity can be achieved in this country! Each party wants to destroy the other party, each party want to come to power, and each thinks that it must rule the whole country, crushing all the other parties. Ethical values have gone down! Each person want to earn crores and crores, so that he would be able to build up his own party, and to feed all the members entering into his party in a grand manner with additional facilities, such as liquor and all the other attractions satisfying the passion. This is the fate of the country! You may protect the country at the borders, but I do not know who can protect this country within this country from the tyranny of all the political parties and the people supporting them.

(3.1)

On focusing our attention on the first 16 characters, we have

Dear Brother! We

(3.2)

On using the EBCDIC code, we write the plaintext (3.2) in the form

\[
\begin{bmatrix}
153 & 157 & 152 & 148 & 147 & 156 & 154 & 158 & 146 & 144 & 150 & 155 & 149 & 145 & 151 & 159 \\
217 & 221 & 216 & 212 & 211 & 220 & 218 & 222 & 210 & 208 & 214 & 219 & 213 & 209 & 215 & 223 \\
137 & 141 & 136 & 132 & 131 & 140 & 138 & 142 & 130 & 128 & 134 & 139 & 133 & 129 & 135 & 143 \\
73 & 77 & 72 & 68 & 67 & 76 & 74 & 78 & 66 & 64 & 70 & 75 & 69 & 65 & 71 & 79 \\
57 & 61 & 56 & 52 & 51 & 60 & 58 & 62 & 50 & 48 & 54 & 59 & 53 & 49 & 55 & 63 \\
201 & 205 & 200 & 196 & 195 & 204 & 202 & 206 & 194 & 192 & 198 & 203 & 197 & 193 & 199 & 207 \\
169 & 173 & 168 & 164 & 163 & 172 & 170 & 174 & 162 & 160 & 166 & 171 & 165 & 161 & 167 & 175 \\
233 & 237 & 232 & 228 & 227 & 236 & 234 & 238 & 226 & 224 & 230 & 235 & 229 & 225 & 231 & 239 \\
41 & 45 & 40 & 36 & 35 & 44 & 42 & 46 & 34 & 32 & 38 & 43 & 37 & 33 & 39 & 47 \\
9 & 13 & 8 & 4 & 3 & 12 & 10 & 14 & 2 & 0 & 6 & 11 & 5 & 1 & 7 & 15 \\
105 & 109 & 104 & 100 & 99 & 108 & 106 & 110 & 98 & 96 & 102 & 107 & 101 & 97 & 103 & 111 \\
185 & 189 & 184 & 180 & 179 & 188 & 186 & 190 & 178 & 176 & 182 & 187 & 181 & 177 & 183 & 191 \\
89 & 93 & 88 & 84 & 83 & 92 & 90 & 94 & 82 & 80 & 86 & 91 & 85 & 81 & 87 & 95 \\
25 & 29 & 24 & 20 & 19 & 28 & 26 & 30 & 18 & 16 & 22 & 27 & 21 & 17 & 23 & 31 \\
121 & 125 & 120 & 116 & 115 & 124 & 122 & 126 & 114 & 112 & 118 & 123 & 117 & 113 & 119 & 127 \\
249 & 253 & 248 & 244 & 243 & 252 & 250 & 254 & 242 & 240 & 246 & 251 & 245 & 241 & 247 & 255 
\end{bmatrix}
\]

\[
P = \begin{bmatrix}
196 & 133 & 129 & 153 \\
64 & 194 & 153 & 150 \\
163 & 136 & 133 & 153 \\
79 & 64 & 230 & 137 
\end{bmatrix}
\]

Let us take the key matrix \( K \), in the form

\[
K = \begin{bmatrix}
120 & 182 & 102 & 13 \\
25 & 14 & 16 & 200 \\
30 & 147 & 61 & 122 \\
40 & 127 & 206 & 91 
\end{bmatrix}
\]

Here, it may be noted that we have taken this \( K \) as the same as (2.15), as this is having modular arithmetic inverse.

Let us take \( E \) in the form

\[
E = \begin{bmatrix}
121 & 157 & 11 & 239 \\
11 & 167 & 189 & 23 \\
167 & 105 & 17 & 19 \\
237 & 109 & 33 & 187 
\end{bmatrix}
\]

On using the plaintext \( P \), the key matrix \( K \), the encryption key bunch matrix \( E \), given by (3.3) – (3.5), and applying the encryption algorithm, we get the ciphertext \( C \) in the form

\[
C = \begin{bmatrix}
116 & 103 & 184 & 219 \\
174 & 112 & 253 & 194 \\
231 & 86 & 28 & 189 \\
239 & 198 & 119 & 132 
\end{bmatrix}
\]

On using the concept of multiplicative inverse, we get the decryption key bunch matrix \( D \) in the form

\[
D = \begin{bmatrix}
201 & 181 & 163 & 15 \\
163 & 23 & 149 & 167 \\
23 & 217 & 241 & 27 \\
229 & 101 & 225 & 115 
\end{bmatrix}
\]
On the C, the D, and the K, given by (3.6), (3.7) and (3.4), and applying the decryption algorithm, we get back the plaintext P.

Let us now examine the avalanche effect. On replacing the 4th row 4th column element, 137 by 153, we have a change of one binary bit in the plaintext P. On using this modified plaintext, the K, and the E, and employing the encryption algorithm, we get the ciphertext C in the form

\[
C = \begin{bmatrix}
147 & 104 & 57 & 131 \\
21 & 46 & 177 & 26 \\
8 & 46 & 235 & 121 \\
5 & 197 & 189 & 55 
\end{bmatrix}
\] (3.8)

On comparing (3.6) and (3.8), after putting them in their binary form, we find that these two ciphertexts differ by 70 bits out of 128 bits.

Let us now consider a one binary bit change in the key K. On replacing the 2nd row 3rd column element, 16 of the key K, given by (3.4), by 48, we have a one bit change. On using this modified key, the plaintext P, and the encryption key bunch matrix E, and the encryption algorithm, given in section 2, the ciphertext corresponding to the modified key is obtained in the form

\[
C = \begin{bmatrix}
141 & 180 & 117 & 2 \\
255 & 166 & 5 & 61 \\
158 & 130 & 243 & 140 \\
94 & 20 & 136 & 3 
\end{bmatrix}
\] (3.9)

On comparing (3.6) and (3.9), after putting them in their binary form, we notice that these two ciphertexts differ by 81 bits out of 128 bits.

From the above discussion, we conclude that, this cipher exhibits a strong avalanche effect, which stands as a benchmark in respect of the strength of the cipher.

IV. CRYPTOANALYSIS

This is the analysis which enables us to establish the strength of the cipher. The different types of attacks available in the literature of the cryptography are

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

 Generally, an analytical proof is offered in the first two cases, and a checkup is done with all possible intuitive ideas in the latter two cases. A cipher is said to be acceptable, if it withstands the first two attacks [1].

In this cipher, we are having a key matrix K and key bunch matrix E. Both are taken to be square matrices of size n. In view of this fact, the size of the key space is

\[
2^{8n^2} \times 2^{7n^2} = 2^{15n^2} = \left(2^{10}\right)^{\frac{5n^2}{2}} \approx \left(10^3\right)^{\frac{5n^2}{2}} = 10^{4.5n^2}. \quad (4.1)
\]

\[
\frac{10^{4.5n^2} \times 10^{-7}}{365 \times 24 \times 60 \times 60} = 3.12 \times 10^{4.5n^2 - 15} \text{ years.} \quad (4.2)
\]

On assuming that the time required for the computation with one value of the key and the one value of the E, in the key space as 10⁻⁷, the time required for the execution of the cipher with all possible keys (i.e., taking all possible pairs of K and E, into consideration) in the key space is

Specifically, in this analysis, as we have n=4, the time given by (4.2), takes the form 3.12 x 10⁻⁷ years. As this time is very large, we conclude that, this cipher cannot be broken by the brute force attack.

Let us examine the known plaintext attack. Here, we have as many pairs of plaintexts and ciphertexts that we like to have, can be had, at our disposal. Confining our attention to r=1, that is to only one round of the iteration process, the system of equations governing the encryption process, can be written in the form

\[
P = \text{Substitute}(\text{Permute}(P)), \quad (4.3)
\]

\[
P = \left[ e_{ij} \times P_{ij} \right] \mod 256, \quad i=1 \text{ to } n, \quad j=1 \text{ to } n, \quad (4.4)
\]

\[
P = \text{Permute}(P), \quad (4.5)
\]

\[
P = \text{Substitute}(P), \quad (4.6)
\]

and

\[
C = P. \quad (4.7)
\]

From (4.7), we can readily have P, as we know C. on using this P, we cannot proceed further, from bottom, as the function Substitute() and ISubstitute() depend upon the key K. Though P on the right hand of (4.3) is known to us, we cannot proceed further, as the P on the left hand side of (4.3) is unknown. In view of the above facts, we cannot the break this cipher by the known plaintext attack.

As the equations, governing the encryption process, are found to be very much involved, in view of the functions Permute() and Substitute(), which are based upon the key, and the modulo arithmetic operation, we cannot imagine to choose, intuitively, any plaintext or ciphertext, for breaking the cipher.

In the light of the above discussion, we conclude that this cipher cannot be broken by any attack, and it is a strong one by all means.

V. COMPUTATIONS AND CONCLUSIONS

In this investigation, we have developed a block cipher which involves the basic ideas of the Hill cipher [5] and the basic concepts of the key bunch matrix. Here, we have made use of the functions Permute() and Substitute(), for permuting the plaintext and for modifying the plaintext, by the substitution process.
On account of these functions and the iteration process, the plaintext has undergone several modifications, in the process of encryption.

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

The plaintext, given by (3.1), is divided into 77 blocks. As the last block is having only 2 characters, we have added 14 zeroes as additional characters to make it a complete block. On carrying out the encryption of each block separately, by using the K and the E, we get the ciphertext corresponding to the entire plaintext (3.1), in the form (5.1). The cryptanalysis carried out in this investigation, has clearly shown that this cipher is strong one and it cannot be broken by any attack. This investigation can be modified by including a large size key matrix and a corresponding encryption key bunch matrix. Then this can be applied to the encryption of images and security of images can be achieved very conveniently.

REFERENCES

[1] William Stallings: Cryptography and Network Security: Principle and Practices”, Third Edition 2003, Chapter 2, pp. 29.
[2] Dr. V.U.K. Sastry, K.Shirisha, “A Novel Block Cipher Involving a Key Bunch Matrix”, in International Journal of Computer Applications (IJCA) (0975 – 8887) Vol.55– No.16, Oct 2012, Foundation of Computer Science, NewYork, pp. 1-6.
[3] Dr. V.U.K. Sastry, K.Shirisha, “A Novel Block Cipher Involving a Key Bunch Matrix, Supplemented with Mix”, in The International Journal of Engineering And Science (IJES), ISSN: 2319 – 1813 ISBN: 2319 – 1805, Vol. – No.2, Dec 2012, pp. 40-47.
[4] Dr. V.U.K. Sastry, K.Shirisha, “A Novel Block Cipher Involving a Key Bunch Matrix”, in International Journal of Computer Applications (IJCA) (0975 – 8887) Vol.55– No.16, Oct 2012, Foundation of Computer Science, NewYork, pp. 1-6.
[5] Lester Hill, (1929), “Cryptography in an algebraic alphabet”, V.36 (6), pp. 306-312., American Mathematical Monthly.

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. She is pursuing her Ph.D. Her research interests are Information Security and Data Mining. She published 9 research papers in International Journals. She stood University topper in the M.Tech.(CSE).
| 81 | 32 | 59 | 178 | 183 | 217 | 133 | 179 | 231 | 203 | 207 | 128 | 246 | 119 | 81 | 62 |
|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|
| 55 | 61 | 56 | 229 | 42  | 33  | 145 | 112 | 82  | 243 | 229 | 126 | 185 | 149 | 154 | 38 |
| 92 | 126 | 1  | 136 | 223 | 37  | 76  | 248 | 61  | 38  | 85  | 234 | 124 | 163 | 133 | 106 |
| 76 | 229 | 178 | 120 | 38  | 189 | 141 | 139 | 184 | 128 | 48  | 30  | 49  | 107 | 154 | 26 |
| 194 | 159 | 88  | 12  | 45  | 6   | 212 | 105 | 9   | 218 | 86  | 107 | 178 | 254 | 72  | 52 |
| 16  | 252 | 20  | 174 | 80  | 61  | 3   | 121 | 72  | 185 | 57  | 193 | 97  | 80  | 174 | 113 |
| 47  | 105 | 22  | 82  | 5   | 140 | 4   | 99  | 112 | 201 | 149 | 172 | 141 | 95  | 127 | 65 |
| 111 | 226 | 137 | 109 | 93  | 208 | 77  | 110 | 223 | 240 | 103 | 187 | 25  | 0   | 66  | 54 |
| 87  | 70  | 124 | 16  | 161 | 250 | 155 | 78  | 172 | 166 | 184 | 203 | 237 | 155 | 138 | 162 |
| 48  | 173 | 149 | 227 | 17  | 171 | 17  | 252 | 241 | 71  | 114 | 211 | 234 | 26  | 109 | 233 |
| 34  | 203 | 112 | 156 | 80  | 156 | 30  | 222 | 29  | 211 | 154 | 233 | 121 | 142 | 244 | 226 |
| 109 | 103 | 255 | 214 | 126 | 112 | 82  | 54  | 206 | 44  | 164 | 111 | 38  | 50  | 170 | 181 |
| 169 | 152 | 20  | 34  | 52  | 205 | 196 | 16  | 249 | 125 | 127 | 173 | 148 | 140 | 182 | 100 |
| 29  | 137 | 247 | 206 | 198 | 170 | 147 | 143 | 97  | 145 | 182 | 180 | 19  | 152 | 76  | 140 |
| 11  | 215 | 73  | 3   | 12  | 6   | 227 | 237 | 113 | 28  | 80  | 107 | 152 | 108 | 226 | 49 |
| 72  | 81  | 236 | 101 | 99  | 81  | 121 | 222 | 114 | 252 | 41  | 131 | 60  | 145 | 42  | 164 |
| 136 | 151 | 255 | 122 | 178 | 149 | 94  | 18  | 9   | 49  | 52  | 83  | 179 | 176 | 137 |
| 88  | 201 | 184 | 97  | 101 | 202 | 46  | 47  | 40  | 200 | 12  | 208 | 197 | 205 | 110 | 51 |
| 28  | 134 | 28  | 94  | 118 | 6   | 165 | 100 | 14  | 98  | 125 | 124 | 130 | 168 | 228 | 214 |
| 139 | 249 | 61  | 52  | 60  | 199 | 54  | 210 | 225 | 238 | 68  | 8   | 105 | 151 | 143 | 138 |
| 119 | 183 | 186 | 132 | 219 | 30  | 232 | 244 | 171 | 153 | 187 | 93  | 104 | 37  | 22  | 33 |
| 241 | 210 | 23  | 176 | 205 | 20  | 107 | 66  | 126 | 17  | 208 | 219 | 16  | 73  | 232 | 224 |
| 180 | 15  | 226 | 39  | 119 | 139 | 167 | 146 | 51  | 89  | 53  | 187 | 67  | 30  | 199 | 99 |
| 234 | 6   | 59  | 199 | 6   | 195 | 55  | 195 | 162 | 132 | 151 | 63  | 168 | 62  | 14  | 84 |

(IJACSA) International Journal of Advanced Computer Science and Applications,
Vol. 4, No.1, 2013