A Novel Cyclic-Lower-Upper-Rectangular (CLUR) Cryptography Method

Suyash Kandele, Veena Anand
Department of Computer Science and Engineering
National Institute of Technology Raipur, India

***Abstract*— The proposed algorithm belongs to the category of symmetric algorithm and hence the decryption process is just the reverse of encryption process. This is a keyless technique of concealing the data, thus reducing the overhead of maintaining the key and its secured transmission. Unlike conventional algorithms which break the message into square matrix, the proposed algorithm partitions and rearranges the message in horizontal rectangular matrix. Here, in the first step we apply rotation pattern on the generated matrix. This step changes the position of elements in addition to changing their relative sequence. Our next step is to alter the number of repetitions and value of characters, which has been implemented by using a part of magic square matrix. We have performed distinct operations at different places which does not form any recognizable pattern for naïve guessing. The proposed algorithm has high randomness and is, therefore, dynamically changing with the varying length of string.**

Keywords—Rectangular matrix; Rotation pattern; Upper magic matrix; Lower magic matrix;

I. INTRODUCTION

Over a score of years, internet has found its applications in education, research, medical science, defense, commerce and many more besides mere communication. A significant amount of data is available on internet. In some fields, secrecy of data may not be important, but for some typical applications security is a crucial aspect. The encryption and decryption of data becomes too important while transmitting it over a shared medium from being stolen away or manipulated by any unauthorized user.

Let us assume a situation where a message has to be sent from one army troop to another through a vulnerable medium. Since the message is of high importance and should be delivered only to its desired destination, the security of this information transfer should be very high. It should not fall in the hands of “Witty and Vigilant” intruder who may temper this data or intercept it. The information is more vulnerable to the attacker who is not interested in data but is passionate about cryptanalysis. In this circumstance, we need to be meticulous about the security measures.

In today’s world, where we are approaching digitization in every possible sector, each user feels the need of a novel, unique and reliable cryptography system; for his/her personalized documents; that is unknown to others. So cryptography exists to be an interminable division, where the minutest and the mightiest algorithm; which is a remarkable exploration of human mind; has its prominent contribution.

In the present work the author has used basic but significantly important methodology to change the position of elements, break the sequence of consecutive elements and alter the number of repetitions & value of characters. Since the mathematical computations involved in the proposed algorithm are not sophisticated, thus it is also suitable for mobile devices and devices with low computational power.

II. BASIC TERMINOLOGY

A. *Horizontal Rectangular Matrix*

Horizontal rectangular matrix is a 2-Dimensional array in which the number of columns is twice the number of rows, i.e. the size will be (n x 2n).

B. *Rotation Pattern*

Rotation pattern comprises of a sequence of steps to modify the position of elements in the matrix.

C. *Magic Square Matrix*

Magic square matrix can be defined as a square matrix in which the elements are arranged in such a way that the sum of elements contained in a row, that in a column and that in the diagonals are all equal. Here we have used the magic square matrix of size 2n x 2n.

D. *Upper Magic Matrix*

The upper half of the magic square matrix is referred to, in this context, as an upper magic matrix. This matrix is formed by magic square matrix (2n x 2n) using its first half rows (1 to n^{th} row) along with all its columns (1 to $2n^{th}$ column).

E. *Lower Magic Matrix*

The lower half of the magic square matrix is referred to, in this context, as a lower magic matrix. This matrix is formed by magic square matrix (2n x 2n) using its second half rows ($(n+1)^{th}$ to $2n^{th}$ row) along with all its columns (1 to $2n^{th}$ column).

F. *XOR Operation*

Here we have performed bitwise XOR operation upon two numbers. When the two bits are identical, the result is evaluated to zero, otherwise to one.

III. PROPOSED ENCRYPTION ALGORITHM WITH EXAMPLE

Step-1
First and foremost, convert each element of the input string into its corresponding ASCII value and calculate its length.

Consider the entered input string is: "Presentation Layer is responsible for Encryption & Decryption."

Here, the ASCII equivalent of the string is: [80 114 101 115 101 110 116 97 116 105 111 110 32 76 97 121 101 114 32 105 115 32 114 101 115 112 111 110 115 105 98 108 101 32 102 111 114 32 69 110 99 114 121 112 116 105 111 110 32 38 32 68 101 99 114 121 112 116 105 111 110 46]

Length of string = 62

Step-2
We break the sequence of input string into rectangular matrices of size n x 2n; such that n is assigned maximum possible value; and place the remaining sequence into a variable REMAINDER_STRING. Note the value of n in a variable "matrix_size[ ]" that maintains the sequence of division. This step is repeated using remainder REMAINDER_STRING of this step as input string, till REMAINDER_STRING contains 8 or more elements.

In this example, the matrices generated are:

$$\begin{bmatrix} 80 & 114 & 101 & 115 & 101 & 110 & 116 & 97 & 116 & 105 \\ 111 & 110 & 32 & 76 & 97 & 121 & 101 & 114 & 32 & 105 \\ 115 & 32 & 114 & 101 & 115 & 112 & 111 & 110 & 115 & 105 \\ 98 & 108 & 101 & 32 & 102 & 111 & 114 & 32 & 69 & 110 \\ 99 & 114 & 121 & 112 & 116 & 105 & 111 & 110 & 32 & 38 \end{bmatrix}$$

$$\begin{bmatrix} 32 & 68 & 101 & 99 \\ 114 & 121 & 112 & 116 \end{bmatrix}$$

REMAINDER_STRING = [105 111 110 46]

matrix_size = [5 2]

Step-3
Then we apply rotation pattern on all the generated rectangular matrices. The sequence of steps in rotation pattern is:

(i) Apply single-up-shift to the even columns of the matrix.

(ii) Rotate the outer-most frame of elements in the matrix in anti-clock wise direction, its inner frame in clock-wise direction, and so on. If the generated matrix has odd number of rows, i.e. value of n is odd, then reverse the elements of middle row which did not participate in either of the rotations in this step.

The matrices after single-up-shift are:

$$\begin{bmatrix} 80 & 110 & 101 & 76 & 101 & 121 & 116 & 114 & 116 & 105 \\ 111 & 32 & 32 & 101 & 97 & 112 & 101 & 110 & 32 & 105 \\ 115 & 108 & 114 & 32 & 115 & 111 & 111 & 32 & 115 & 110 \\ 98 & 114 & 101 & 112 & 102 & 105 & 114 & 110 & 69 & 38 \\ 99 & 114 & 121 & 115 & 116 & 110 & 111 & 97 & 32 & 105 \end{bmatrix}$$

$$\begin{bmatrix} 32 & 121 & 101 & 116 \\ 114 & 68 & 112 & 99 \end{bmatrix}$$

The matrices after rotations are:

$$\begin{bmatrix} 110 & 101 & 76 & 101 & 121 & 116 & 114 & 116 & 105 & 105 \\ 80 & 108 & 32 & 32 & 101 & 97 & 112 & 101 & 110 & 110 \\ 111 & 114 & 114 & 32 & 115 & 111 & 111 & 32 & 32 & 38 \\ 115 & 101 & 112 & 102 & 105 & 114 & 110 & 69 & 115 & 105 \\ 98 & 99 & 114 & 121 & 115 & 116 & 110 & 111 & 97 & 32 \end{bmatrix}$$

$$\begin{bmatrix} 121 & 101 & 116 & 99 \\ 32 & 114 & 68 & 112 \end{bmatrix}$$

Since, the first matrix has odd number of rows, so reversing the un-changed elements. After this step, the first matrix becomes:

$$\begin{bmatrix} 110 & 101 & 76 & 101 & 121 & 116 & 114 & 116 & 105 & 105 \\ 80 & 108 & 32 & 32 & 101 & 97 & 112 & 101 & 110 & 110 \\ 111 & 114 & 32 & 111 & 111 & 115 & 32 & 114 & 32 & 38 \\ 115 & 101 & 112 & 102 & 105 & 114 & 110 & 69 & 115 & 105 \\ 98 & 99 & 114 & 121 & 115 & 116 & 110 & 111 & 97 & 32 \end{bmatrix}$$

Step-4
We take a magic square matrix of size 2n x 2n. If the number of rows in the rectangular matrix under consideration is odd (value of n is odd) then we proceed to step-5, otherwise if the number of rows in the rectangular matrix under consideration is even (value of n is even) then we continue to step-6.

$$\left[\begin{array}{cccccccccc} 92 & 99 & 1 & 8 & 15 & 67 & 74 & 51 & 58 & 40 \\ 98 & 80 & 7 & 14 & 16 & 73 & 55 & 57 & 64 & 41 \\ 4 & 81 & 88 & 20 & 22 & 54 & 56 & 63 & 70 & 47 \\ 85 & 87 & 19 & 21 & 3 & 60 & 62 & 69 & 71 & 28 \\ 86 & 93 & 25 & 2 & 9 & 61 & 68 & 75 & 52 & 34 \\ \hline 17 & 24 & 76 & 83 & 90 & 42 & 49 & 26 & 33 & 65 \\ 23 & 5 & 82 & 89 & 91 & 48 & 30 & 32 & 39 & 66 \\ 79 & 6 & 13 & 95 & 97 & 29 & 31 & 38 & 45 & 72 \\ 10 & 12 & 94 & 96 & 78 & 35 & 37 & 44 & 46 & 53 \\ 11 & 18 & 100 & 77 & 84 & 36 & 43 & 50 & 27 & 59 \end{array}\right] \begin{array}{c} \text{Upper Magic Matrix} \\ \\ \text{Lower Magic Matrix} \end{array}$$

$$\left[\begin{array}{cccc} 16 & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ \hline 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{array}\right] \begin{array}{c} \text{Upper Magic Matrix} \\ \\ \text{Lower Magic Matrix} \end{array}$$

Step-5
We take the upper magic matrix of size n x 2n and perform $\odot$ operation on the corresponding element of generated rectangular matrix. If the value of element of upper magic matrix is odd then $\odot$ means XOR, otherwise $\odot$ means addition.

Using the upper magic matrix:

$$\begin{bmatrix} 92 & 99 & 1 & 8 & 15 & 67 & 74 & 51 & 58 & 40 \\ 98 & 80 & 7 & 14 & 16 & 73 & 55 & 57 & 64 & 41 \\ 4 & 81 & 88 & 20 & 22 & 54 & 56 & 63 & 70 & 47 \\ 85 & 87 & 19 & 21 & 3 & 60 & 62 & 69 & 71 & 28 \\ 86 & 93 & 25 & 2 & 9 & 61 & 68 & 75 & 52 & 34 \end{bmatrix}$$

The 1st matrix after this step is:

$$\begin{bmatrix} 202 & 6 & 77 & 109 & 118 & 55 & 188 & 71 & 163 & 145 \\ 178 & 188 & 39 & 46 & 117 & 40 & 71 & 92 & 174 & 71 \\ 115 & 35 & 120 & 131 & 133 & 169 & 88 & 77 & 102 & 9 \\ 38 & 50 & 99 & 115 & 106 & 174 & 172 & 0 & 52 & 133 \\ 184 & 62 & 107 & 123 & 122 & 73 & 178 & 36 & 149 & 66 \end{bmatrix}$$

Step-6
We take the lower magic matrix of size n x 2n and perform ⦿ operation on the corresponding element of generated rectangular matrix. If the value of element of lower magic matrix is even then ⦿ means XOR, otherwise ⦿ means addition.

Using the lower magic matrix:

$$\begin{bmatrix} 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{bmatrix}$$

The 2nd matrix after this step is:

$$\begin{bmatrix} 130 & 108 & 114 & 111 \\ 36 & 124 & 83 & 113 \end{bmatrix}$$

Step-7
Calculate the sum of row of magic square matrix of size same as that of the value of each element in the variable “matrix_size[ ]” and store the sum in variable “sum_of_magic_matrix[ ]”.

Convert the magic square matrix of size 3 into a 1-dimensional array “magic_array[ ]” and then square each term in it.

Here, sum_of_magic_matrix = [65 5]

magic_array = [64 1 36 9 25 49 16 81 4]

Step-8
For all the elements in the variable REMAINDER_STRING, perform XOR operation with the corresponding value of element in “sum_of_magic_matrix[ ]” and then perform XOR operation with corresponding elements in “magic_array[ ]”.

After this step, the content of REMAINDER_STRING is:

[104 107 11 34]

Step-9
In this last step, we merge all the rectangular matrices and the variable REMAINDER_STRING, in the order they were divided, to form the cipher text.

[202 6 77 109 118 55 188 71 163 145 178 188 39 46 117 40 71 92 174 71 115 35 120 131 133 169 88 77 102 9 38 50 99 115 106 174 172 0 52 133 184 62 107 123 122 73 178 36 149 66 130 108 114 111 36 124 83 113 104 107 11 34]

Cipher text is:

ÊMmv7¼G£²¼'.u(G\®Gs#x©XMf &2csj®¬ 4¸>k{zI²$Blro$|Sqhk "

IV. Proposed Decryption Algorithm with Example

Step-1
First and foremost, convert each element of the input string into its corresponding ASCII value and calculate its length.

Consider the entered input string is:

“ÊMmv7¼G£²¼'.u(G\®Gs#x©XMf &2csj®¬ 4¸>k{zI²$Blro$|Sqhk "”

Here, the ASCII equivalent of the string is:

[202 6 77 109 118 55 188 71 163 145 178 188 39 46 117 40 71 92 174 71 115 35 120 131 133 169 88 77 102 9 38 50 99 115 106 174 172 0 52 133 184 62 107 123 122 73 178 36 149 66 130 108 114 111 36 124 83 113 104 107 11 34]

Length of string = 62

Step-2
We break the sequence of input string into rectangular matrices of size n x 2n; such that n is assigned maximum possible value; and place the remaining sequence into a variable REMAINDER_STRING. Note the value of n in a variable “matrix_size[ ]” that maintains the sequence of division. This step is repeated using remainder REMAINDER_STRING of this step as input string, till REMAINDER_STRING contains 8 or more elements.

In this example, the matrices generated are:

$$\begin{bmatrix} 202 & 6 & 77 & 109 & 118 & 55 & 188 & 71 & 163 & 145 \\ 178 & 188 & 39 & 46 & 117 & 40 & 71 & 92 & 174 & 71 \\ 115 & 35 & 120 & 131 & 133 & 169 & 88 & 77 & 102 & 9 \\ 38 & 50 & 99 & 115 & 106 & 174 & 172 & 0 & 52 & 133 \\ 184 & 62 & 107 & 123 & 122 & 73 & 178 & 36 & 149 & 66 \end{bmatrix}$$

$$\begin{bmatrix} 130 & 108 & 114 & 111 \\ 36 & 124 & 83 & 113 \end{bmatrix}$$

REMAINDER_STRING = [104 107 11 34]

matrix_size = [5 2]

Step-3
We take a magic square matrix of size 2n x 2n. If the number of rows in the rectangular matrix under consideration is odd (value of n is odd) then we proceed to step-4, otherwise if the number of rows in the rectangular matrix under consideration is even (value of n is even) then we continue to step-5.

$$\begin{array}{cl}
\left[\begin{array}{cccccccccc}
92 & 99 & 1 & 8 & 15 & 67 & 74 & 51 & 58 & 40 \\
98 & 80 & 7 & 14 & 16 & 73 & 55 & 57 & 64 & 41 \\
4 & 81 & 88 & 20 & 22 & 54 & 56 & 63 & 70 & 47 \\
85 & 87 & 19 & 21 & 3 & 60 & 62 & 69 & 71 & 28 \\
86 & 93 & 25 & 2 & 9 & 61 & 68 & 75 & 52 & 34 \\
\hline
17 & 24 & 76 & 83 & 90 & 42 & 49 & 26 & 33 & 65 \\
23 & 5 & 82 & 89 & 91 & 48 & 30 & 32 & 39 & 66 \\
79 & 6 & 13 & 95 & 97 & 29 & 31 & 38 & 45 & 72 \\
10 & 12 & 94 & 96 & 78 & 35 & 37 & 44 & 46 & 53 \\
11 & 18 & 100 & 77 & 84 & 36 & 43 & 50 & 27 & 59
\end{array}\right] &
\begin{array}{l}
\text{Upper Magic Matrix} \\ \\ \\ \\ \\
\text{Lower Magic Matrix}
\end{array}
\end{array}$$

$$\begin{array}{cl}
\left[\begin{array}{cccc}
16 & 2 & 3 & 13 \\
5 & 11 & 10 & 8 \\
\hline
9 & 7 & 6 & 12 \\
4 & 14 & 15 & 1
\end{array}\right] &
\begin{array}{l}
\text{Upper Magic Matrix} \\ \\
\text{Lower Magic Matrix}
\end{array}
\end{array}$$

Step-4

We take the upper magic matrix of size n x 2n and perform ◉ operation on the corresponding element of generated rectangular matrix. If the value of element of upper magic matrix is odd then ◉ means XOR, otherwise ◉ means subtraction.

Using the upper magic matrix:

$$\begin{bmatrix}
92 & 99 & 1 & 8 & 15 & 67 & 74 & 51 & 58 & 40 \\
98 & 80 & 7 & 14 & 16 & 73 & 55 & 57 & 64 & 41 \\
4 & 81 & 88 & 20 & 22 & 54 & 56 & 63 & 70 & 47 \\
85 & 87 & 19 & 21 & 3 & 60 & 62 & 69 & 71 & 28 \\
86 & 93 & 25 & 2 & 9 & 61 & 68 & 75 & 52 & 34
\end{bmatrix}$$

The 1st matrix after this step is:

$$\begin{bmatrix}
110 & 101 & 76 & 101 & 121 & 116 & 114 & 116 & 105 & 105 \\
80 & 108 & 32 & 32 & 101 & 97 & 112 & 101 & 110 & 110 \\
111 & 114 & 32 & 111 & 111 & 115 & 32 & 114 & 32 & 38 \\
115 & 101 & 112 & 102 & 105 & 114 & 110 & 69 & 115 & 105 \\
98 & 99 & 114 & 121 & 115 & 116 & 110 & 111 & 97 & 32
\end{bmatrix}$$

Now, continue to Step-6

Step-5

We take the lower magic matrix of size n x 2n and perform ◉ operation on the corresponding element of generated rectangular matrix. If the value of element of lower magic matrix is even then ◉ means XOR, otherwise ◉ means subtraction.

Using the lower magic matrix:

$$\begin{bmatrix}
9 & 7 & 6 & 12 \\
4 & 14 & 15 & 1
\end{bmatrix}$$

The 2nd matrix after this step is:

$$\begin{bmatrix}
121 & 101 & 116 & 99 \\
32 & 114 & 68 & 112
\end{bmatrix}$$

Step-6

Then we apply rotation pattern on all the generated rectangular matrices. The sequence of steps in rotation pattern is:

(i) Rotate the outer-most frame of elements in the matrix in clock-wise direction, its inner frame in anti-clock wise direction, and so on. If the generated matrix has odd number of rows, i.e. value of n is odd, then reverse the elements of middle row which did not participate in either of the rotations in this step.

(ii) Apply single-down-shift to the even columns of the matrix.

The matrices after rotations are:

$$\begin{bmatrix}
80 & 110 & 101 & 76 & 101 & 121 & 116 & 114 & 116 & 105 \\
111 & 32 & 32 & 101 & 97 & 112 & 101 & 110 & 32 & 105 \\
115 & 108 & 32 & 111 & 111 & 115 & 32 & 114 & 115 & 110 \\
98 & 114 & 101 & 112 & 102 & 105 & 114 & 110 & 69 & 38 \\
99 & 114 & 121 & 115 & 116 & 110 & 111 & 97 & 32 & 105
\end{bmatrix}$$

$$\begin{bmatrix}
32 & 121 & 101 & 116 \\
114 & 68 & 112 & 99
\end{bmatrix}$$

Since, the first matrix has odd number of rows, so reversing the un-changed elements. After this step, the first matrix becomes:

$$\begin{bmatrix}
80 & 110 & 101 & 76 & 101 & 121 & 116 & 114 & 116 & 105 \\
111 & 32 & 32 & 101 & 97 & 112 & 101 & 110 & 32 & 105 \\
115 & 108 & 114 & 32 & 115 & 111 & 111 & 32 & 115 & 110 \\
98 & 114 & 101 & 112 & 102 & 105 & 114 & 110 & 69 & 38 \\
99 & 114 & 121 & 115 & 116 & 110 & 111 & 97 & 32 & 105
\end{bmatrix}$$

The matrices after single-down-shift are:

$$\begin{bmatrix}
80 & 114 & 101 & 115 & 101 & 110 & 116 & 97 & 116 & 105 \\
111 & 110 & 32 & 76 & 97 & 121 & 101 & 114 & 32 & 105 \\
115 & 32 & 114 & 101 & 115 & 112 & 111 & 110 & 115 & 105 \\
98 & 108 & 101 & 32 & 102 & 111 & 114 & 32 & 69 & 110 \\
99 & 114 & 121 & 112 & 116 & 105 & 111 & 110 & 32 & 38
\end{bmatrix}$$

$$\begin{bmatrix}
32 & 68 & 101 & 99 \\
114 & 121 & 112 & 116
\end{bmatrix}$$

Step-7

Calculate the sum of row of magic square matrix of size same as that of the value of each element in the variable “matrix_size[ ]” and store the sum in variable “sum_of_magic_matrix[ ]”.

Convert the magic square matrix of size 3 into a 1-dimensional array “magic_array[ ]” and then square each term in it.

Here, sum_of_magic_matrix = [65 5]

magic_array = [64 1 36 9 25 49 16 81 4]

Step-8

For all the elements in the variable REMAINDER_STRING, perform XOR operation with the corresponding value of element in “magic_array[ ]” and then perform XOR operation with corresponding elements in “sum_of_magic_matrix[ ]”.

After this step, the content of REMAINDER_STRING is:

[105 111 110 46]

Step-9
In this last step, we merge all the rectangular matrices and the variable REMAINDER_STRING, in the order they were divided, to form the plain text.

[80 114 101 115 101 110 116 97 116 105 111 110 32 76 97 121 101 114 32 105 115 32 114 101 115 112 111 110 115 105 98 108 101 32 102 111 114 32 69 110 99 114 121 112 116 105 111 110 32 38 32 68 101 99 114 121 112 116 105 111 110 46]

Plain text is:

Presentation Layer is responsible for Encryption & Decryption.

V. FLOWCHART OF ENCRYPTION ALGORITHM

The flow chart of encryption algorithm is:

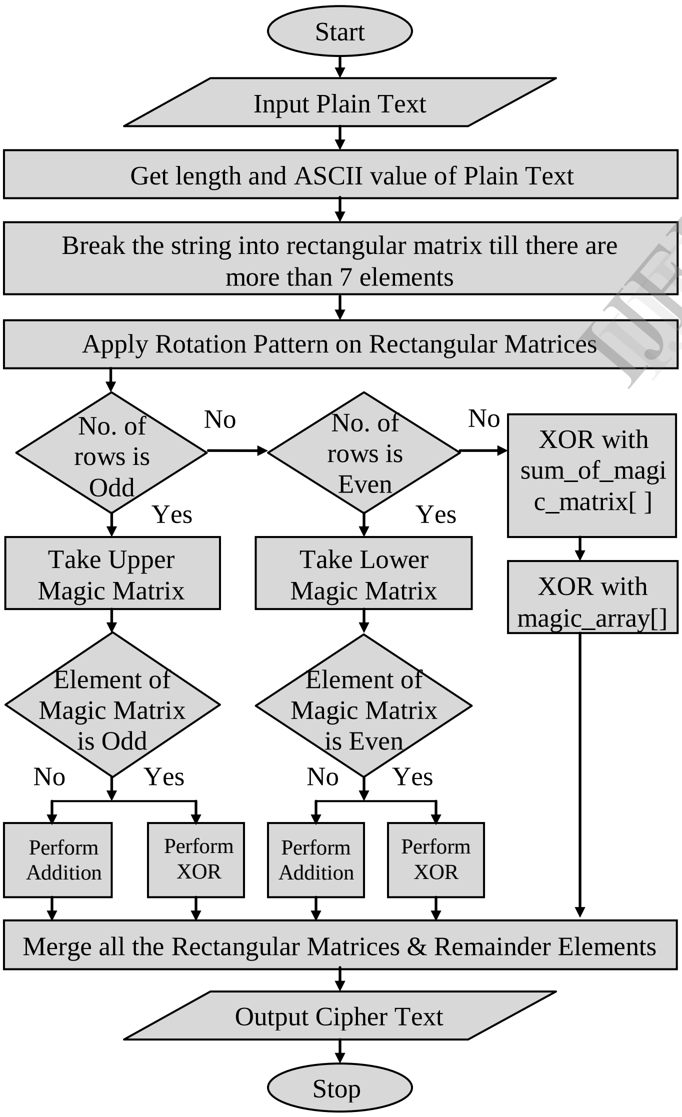

Fig. 1. Flow Chart of Encryption Algorithm.

The flow chart of rotation pattern for encryption is:

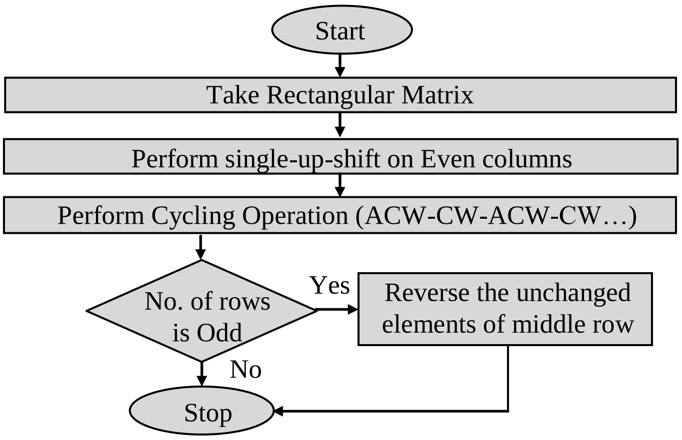

Fig. 2. Flow Chart of Rotation Pattern for Encryption.

VI. FLOWCHART OF DECRYPTION ALGORITHM

The flow chart of decryption algorithm is:

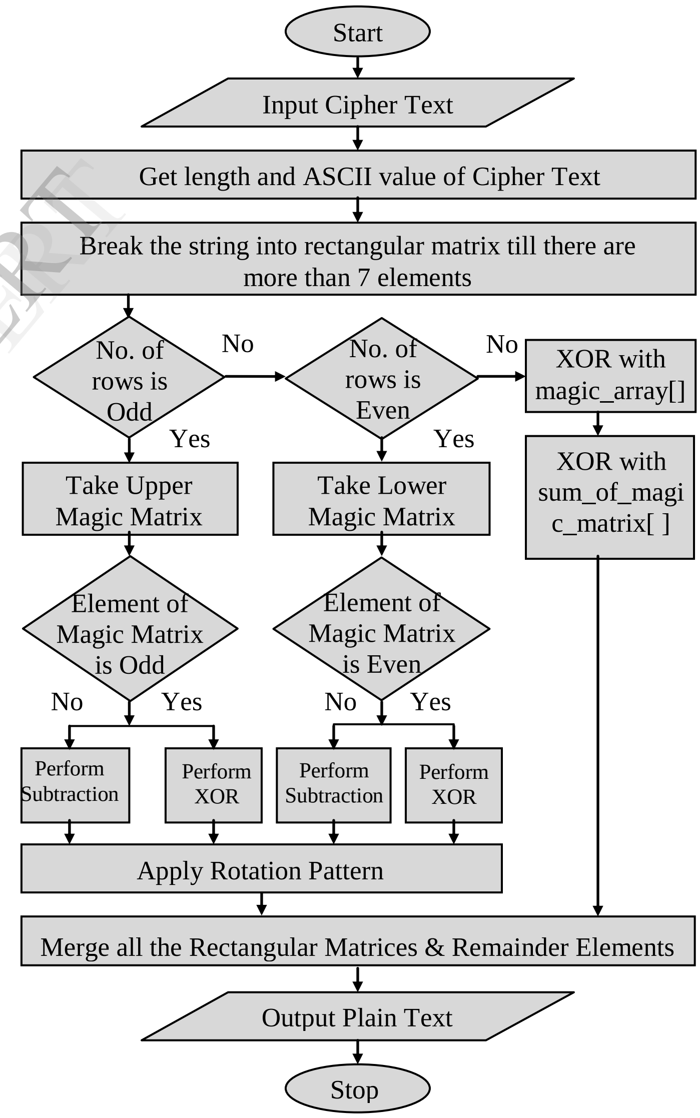

Fig. 3. Flow Chart of Decryption Algorithm.

The flow chart of rotation pattern for decryption is:

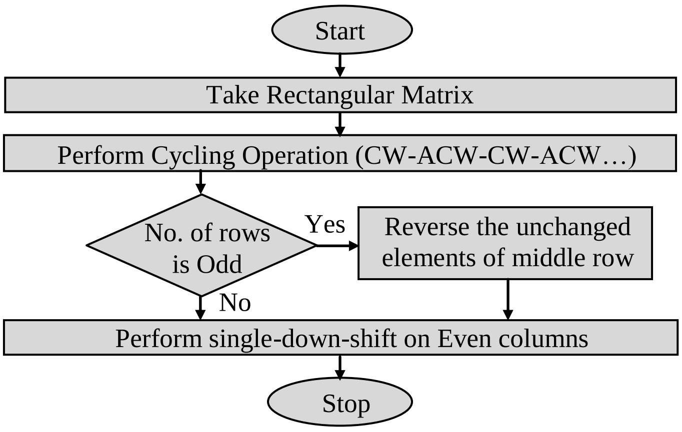

Fig. 4. Flow Chart of Rotation Pattern for Decryption.

VII. RESULT

We have applied the proposed algorithm on string of various lengths, and aroused with astonishing and remarkable results. The structure of plain text is found to be entirely changed. The following table enlists the sample plain text, their length, encryption time and the corresponding cipher text generated.

TABLE I. RESULTS OF CLUR ENCRYPTION ALGORITHM

Plain Text	Length	Time (in second)	Cipher Text
Password	8	0.093	xzb□Typs
aaaaaaaaaaaaa	13	0.095	jhgmeopb$e@m}
Money is 5000$ take it	22	0.107	Po;#NPs539v9q6~u*.Br
Network Security is essential	29	0.120	ukf•M~F~kt{gl4nlr□w }/o&eG
National Institute of Technology Raipur, C.G.	45	0.115	iVuG~w□sB2IQ/aR¤_~ T|YsmR-h~h*e$~asBG(K
Presentation Layer is responsible for Encryption & Decryption.	62	0.127	ÊMmv7¼G£²¼'.u(G\® Gs#x©XMf &2csj®¬ 4‚>k{zI²$Blro$|Sqhk "
Calculating efficiency of the algorithm to check the success. It's my success too.. ☺ feeling great…yahoooo…!! !!	112	0.124	ØªtxréæùCÑ¬"úºj~~rëøI Úl±§üþvQQÍ× éÈv`ÔÂÌYG•²!~ÀÂÃR •Zw&Qì¬(TÖLGpz Yñù£-[ÚÌÚWenáõñýË\ ximpfp•ª-
My name is Suyash Kandele, and I am not a terrorist, but I am a student. I live in my house and not in the entire city Bhilai. Studies are my hobbies and I have no time for hobbies. I like greenery, and wherever I find it, I remove it from there. Don't read the above passage, just enjoy it and bang your head here….with lots of jerks!!!!!!!	341	0.110	ˆ`y?NRÁ{¤ÙF>L© kÈ|¬=T«*Û⍰_³;ÀG á\-:ÉU õ)ÈEßüùlÓ¢°üf äåæûÄ+ŸQæçÿ&; Ý_óñúv4DÎÔìtq_Ì¿ ûó½osU- ³Äv1@\---ù

On enormously increasing the length of string, the time required to encrypt it, is still changed by a negligible amount. We have plotted a graph "Encryption Time Graph" to represent the encryption time taken by the strings of various lengths.

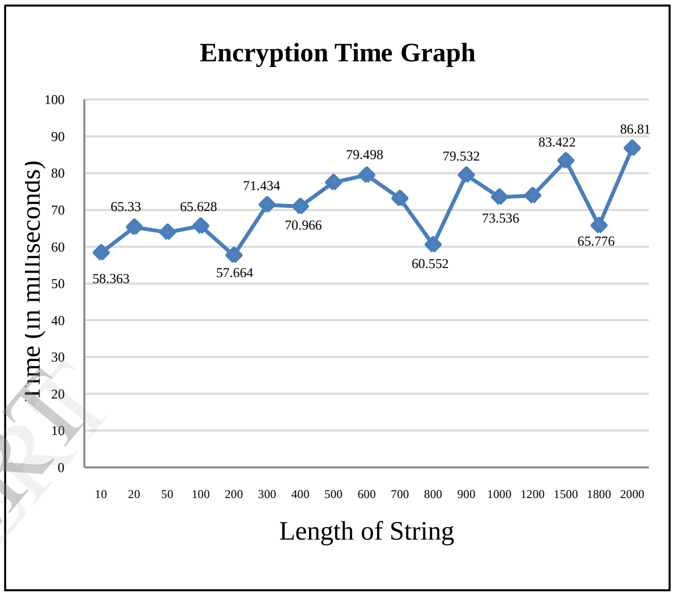

Fig. 5. Encryption Time Graph.

On the basis of above observations, we have calculated and plotted the value of time required to encrypt a single character, for each string, in Time per Character Graph, and found drastic minimization in time.

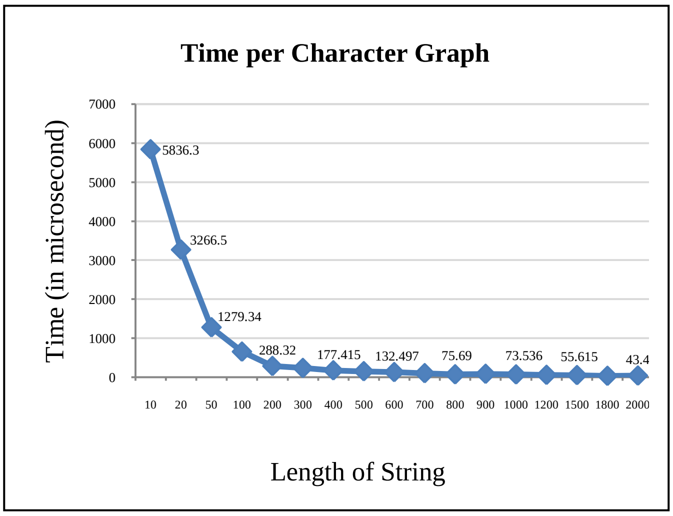

Fig. 6. Time per Character Graph.

VIII. Conclusion

In this proposed work, we have introduced a novel technique of encrypting text without using any key. The plain text is broken into rectangular matrices of varying sizes, with each matrix encrypted separately and distinctly. To change the number of times a character is repeated, we have employed the modified form of magic matrix. The values in the applied magic matrix directs the further encryption operation to be performed, and hence provides dynamism to this work. The basic and easy to implement mathematical and logical operations are used in this algorithm which makes it highly suitable for mobile devices and devices with low computation power.

IX. Future scope

This algorithm, though small, introduces a novel, less time consuming approach and is self sufficient for all kind of data encryption where processor utilization is a constraint. This algorithm provides a frame work and can be used for the innovation of much more unpredictable algorithms.

References

[1] Dripto Chatterjee, Suvadeep Dasgupta, Joyshree Nath, and Asoke Nath, "A new Symmetric key Cryptography Algorithm using extended MSA method: DJSA symmetric key algorithm", IEEE, 978-0-7695-4437-3/11, DOI 10.1109/CSNT.2011.25, 2011.

[2] Neeraj Khanna, Joyshree Nath, Joel James, Amlan Chakrabarti, Sayantan Chakraborty, and Asoke Nath, "New Symmetric Key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJJSAA symmetric key algorithm", IEEE Computer Society, 978-0-7695-4437-3/11, DOI 10.1109/ CSNT.2011.33, 2011.

[3] D. Rajavel, and S. P. Shantharajah, "Cubical Key Generation and Encryption Algorithm Based on Hybrid Cube's Rotation", IEEE, 978-1-4673-1039-0/12, March 21- 23, 2012.

[4] Gaurav Bhadra, Tanya Bala, Samik Banik, Asoke Nath, and Joyshree Nath,"Bit Level Encryption Standard (BLES): Version-II", IEEE, 978-1-4673-4805-8/12, 2012.

[5] Rishav Ray, Jeeyan Sanyal, Debanjan Das, and Asoke Nath, "A new Challenge of hiding any encrypted secret message inside any Text/ASCII file or in MS word file: RJDA Algorithm", IEEE, 978-0-7695-4692-6/12, DOI 10.1109/CSNT.2012.191, 2012.

[6] Somdip Dey, "SD-C1BBR: SD-Count-1-Byte-Bit Randomization: A New Advanced Cryptographic Randomization Technique", IEEE, 978-1-4673-4805-8/12, 2012.

[7] Akanksha Mathur, "A Research paper: An ASCII value based data encryption algorithm and its comparison with other symmetric data encryption algorithms", International Journal on Computer Science and Engineering, ISSN: 0975-3397, vol. 4, no. 09, 2012, pp. 1650-1657.

[8] Sayak Guha, Tamodeep Das, Saima Ghosh, Joyshree Nath, Sankar Das, and Asoke Nath, "A New Data Hiding Algorithm With Encrypted Secret Message Using TTJSA Symmetric Key Crypto System", Journal of Global Research in Computer Science,ISSN-2229-371X, vol. 3, no. 4, April 2012.

[9] Somdip Dey, "SD-AREE: An Advanced Modified Caesar Cipher Method to Exclude Repetition from a Message", International Journal of Information and Network Security, vol. 1, no. 2, ISSN: 2089-3299, June 2012.

[10] Somdip Dey, Joyshree Nath, and Asoke Nath, "An Integrated Symmetric Key Cryptographic Method-Amalgamation of TTJSA Algorithm, Advanced Caesar Cipher Algorithm, Bit Rotation and Reversal Method: SJA Algorithm", I. J. Modern Education and Computer Science, DOI: 10.5815/ijmecs.2012.05.01, 2012.

[11] Somdip Dey, Kalyan Mondal, Joyshree Nath, and Asoke Nath, "Advanced Steganography Algorithm Using Randomized Intermediate QR Host Embedded With Any Encrypts Secret Message: ASA_QR Algorithm", I. J. Modern Education and Computer Science, DOI:10.5815/ijmecs.2012.06.08, 2012.

[12] Mr. Rangaswamy D. A., and Mr. Punithkumar M. B., "New Symmetric Key Cryptographic Algorithm Using Combined Bit Manipulation and MSA Encryption Algorithm: NJJSAA Symmetric Key Algorithm", International Journal of Innovative Research and Development, vol. 2, Issue 6, ISSN: 2278-0211, June 2013.

[13] Georgiana Mateescu, and Marius Vladescu, "A Hybrid Approach of System Security for Small and Medium Enterprises: combining different Cryptography Techniques", IEEE, Proceedings of the 2013 Federated Conference on Computer Science and Information Systems pp. 659–662, 978-1-4673-4471-5, 2013

[14] Nehal Kandele, and Shrikant Tiwari, "New Cryptography Method Using Dynamic Base Trasformation: DBTC Symmetric Key Algorithm", International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 3, Issue 5, October 2013.

[15] Nehal Kandele, and Shrikant Tiwari, "New Cryptography Method Using Relative Displacement: RDC Symmetric Key Algorithm", International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 10, October – 2013.

[16] Nehal Kandele, and Shrikant Tiwari, "A New Combined Symmetric Key Cryptography CRDDBT Using – Relative Displacement (RDC) and Dynamic Base Transformation (DBTC)", International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 10, October – 2013.

[17] Mohammad A. AlAhmad, Imad Fakhri Alshaikhli, and Bashayer Moh. Jumaah, "Protection of the Digital Holy Quran Hash Digest by Using Cryptography Algorithms", IEEE, International Conference on Advanced Computer Science Applications and Technologies, 978-1-4799-2758-6/13 IEEE DOI 10.1109/ACSAT.2013.55, 2013.

[18] Rober Grimes, and Junhua Ding, "Development of a Novel Cryptography Tool for Personal Communication", IEEE, 978-1-4799-3106-4114.

[19] Ankur Chaudhary, Khaleel Ahmad, and M.A. Rizvi, "E-commerce Security Through Asymmetric Key Algorithm", IEEE, Fourth International Conference on Communication Systems and Network Technologies, 978-1-4799-3070-8/14, DOI 10.1109/CSNT.2014.163, 2014.

[20] Naitik Shah, Nisarg Desai, and Viral Vashi, "Efficient Cryptography for Data Security", IEEE, 978-93-80544-12-0/14, 2014.

[21] Md. Palash Uddin, Md. Abu Marjan, Nahid Binte Sadia, and Md. Rashedul Islam, "Developing a Cryptographic Algorithm Based on ASCII Conversions and a Cyclic Mathematical Function", IEEE, 3rd INTERNATIONAL CONFERENCE ON INFORMATICS, ELECTRONICS & VISION 2014, 978-1-4799-5180-2/14, 2014.