ENCRYPTION BASED ON MATRIX TRANSFORMATION TECHNIQUE

Sagar Suryavanshi[1], Jasmin Patil[2],Deepali Pawaskar[3], Prof. Asmita Deshmukh[4]
[1,2,3]B.E Student ,[4] Assistant Professor ,
Computer Engineering Department, KCCEMSR,
Thane, India
[1]suryavanshisagar19@gmail.com,[2]deepalipawaskar123@gmail.com[3]jasminishwarpatil@gmail.com, [4]
asmitadeshmukh7@gmail.com

Abstract - Cryptography is one way to transfer secure information over unsecure network such as Internet. In this paper we are proposing a safe method to transmit data over Internet. In our application, the data sent to a remote host is encrypted first using encryption key then the data is sent to the destination machine. We are performing matrix transformations which are based on circular queue techniques and random functions. At the sender side the first step is to convert the original message into an image, and then perform various transformations on image matrix randomly such as circular left shift, circular right shift and reverse.

Keywords
Encryption, Decryption, Matrix Scrambling, Circular Queue.

I. INTRODUCTION

Due to rapid growth of Internet, information tide brings about great economic and social benefit along with which the use of this technology has increased. However, Internet confronts many safety problems. The problems include network attack, hacking, interception of important information and tampering it which is a threat to Internet. Information security is a major concern of our society. The solution for secure transmission of data over network is cryptography. Cryptography is used to convert the plain text to encode or make unreadable form of text. The sensitive data is encrypted on the sender side using an encryption key then it is sent to the destination, At the receiver side, the data is decrypted using an algorithm and decryption key as shown in "Figure 1". After the decryption process the message is converted back to original format from encrypted format. There are some standard methods which are used with cryptography such as private-key encryption, public-key encryption, digital signature, and hash functions. This paper will put forward a safe method of data transmission to tackle the security problem. It proposes a new technique on matrix scrambling which will be based on random function, shifting and reversing techniques of circular queue. It will be able to resist all kinds of cryptanalytic, statistical attacks.

II. RELATED WORK

The queue transformation based digital image encryption algorithm proposed by F.Y. LI Min[1], which works efficiently with low time complexity compared to Yongwei et al[2]. However, it still has some shortcomings. Firstly, the algorithm signifies some certain regularity. In this algorithm one random element (i,j) is selected from the matrix and cyclic shifted leftwards and rightwards based on the position in the matrix. This makes encryption output periodic which makes the algorithm vulnerable to attacks.
Singh and Gilhorta[3] proposed encrypting a word of text to a floating point number that lies in the range of 0 and 1. The floating point number is then converted into a binary number and after that one time key is used to encrypt the binary number.

III. PROPOSED TECHNIQUE

In this paper, we propose an efficient digital encryption algorithm based on matrix scrambling technique which is based on random function, shifting and reversing techniques of circular queue, with efficient time complexity.

A. Encryption Algorithm

The plaintext chosen is arranged into a Bi-directional circular Queue data structure .In the matrix A of order m × n. Input an integer parameter, w, as the count of operations, say, the time of transformation we made to matrix. To some certain degree, this parameter represents the intensity of the encryption.

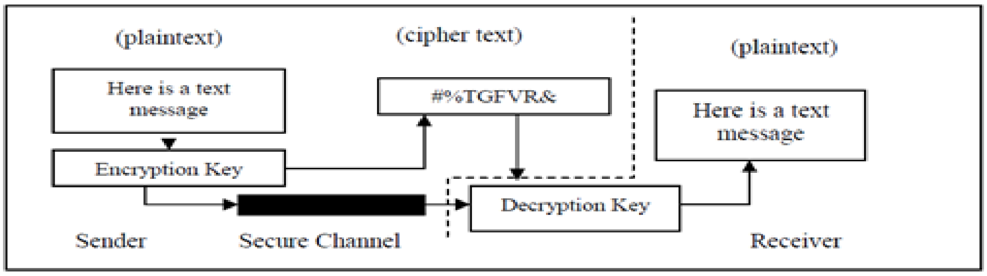


Figure 1: Encryption and Decryption methods with a secure channel for key exchange

However, this does not necessarily mean that the greater w is, the more intensive the encryption is. Random() function is used to generate random positive integer, this integer value has to be converted into a binary equivalent. Depending on the binary bit from least significant bit to most significant bit, the choice to select rows or columns is made.

In case of row, two rows r1 and r2 are selected randomly from the matrix similarly two random values of columns c1, c2 are chosen to determine the range of rows on which transformation has to be performed. To perform transformation operations, a value is calculated from Random( ) mod3, circular left shift, circular right shift and reverse operations on the selected rows is performed depending on the value which may be 0,1 or 2 In case of column, two columns c1 and c2 are selected randomly from the matrix, two random values of rows r1,r2 are chosen to determine the range of columns on which transformation has to be performed.

Then to perform transformation operations similar to rows, a value is calculated from Random( ) mod 3, based on it circular upward shift, circular downward shift and reverse operations are performed similar as rows.

The entire process is repeated w number of times; if w ≤ length (binary sequence) then the process is performed only w number of times from LSB to MSB else the binary sequence is repeated from LSB to MSB until the w operations.

In this manner, the entire matrix elements are transposed. For each operation performed, the operation should be recorded as a sub~key in a file, which becomes the key file. The key file should be maintained secret. The algorithms for row and column transformation are shown in Figure 3 and Figure 4 respectively.

The sub-key is 6-tuple and is given as follows, sub-key=(T,op,α1,α2,β1,β2) where,

T= Transformation applied either row or column i.e. 'R' or 'C'

Op= 0 or 1 or 2 (These values changes for row and column transformations)

α1,α2= Two random rows or columns selected depending on the transformation

β1,β2= min and max values of range for two selected α1,α2

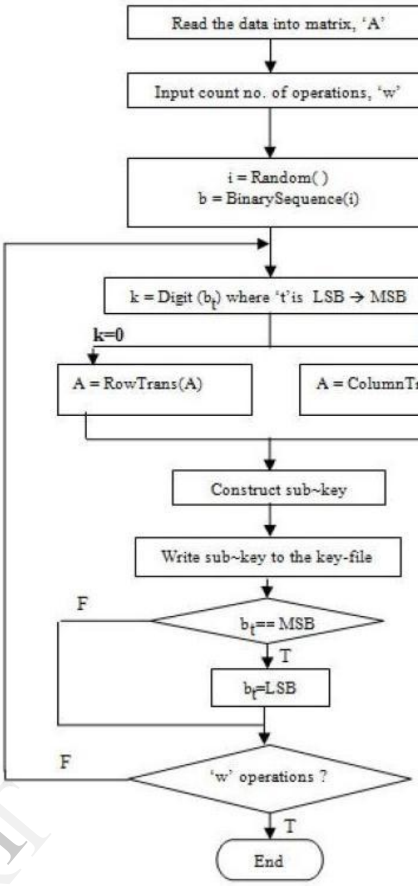

Read the data into matrix, 'A'
Input count no. of operations, 'w'
i = Random( )
b = BinarySequence(i)
k = Digit (b_t) where 't' is LSB → MSB
k=0
k=1
A = RowTrans(A)
A = ColumnTrans(A)
Construct sub~key
Write sub~key to the key-file
F
b_t== MSB
T
b_t=LSB
F
'w' operations ?
T
End

Figure 2: Encryption Algorithm

Row Transformation

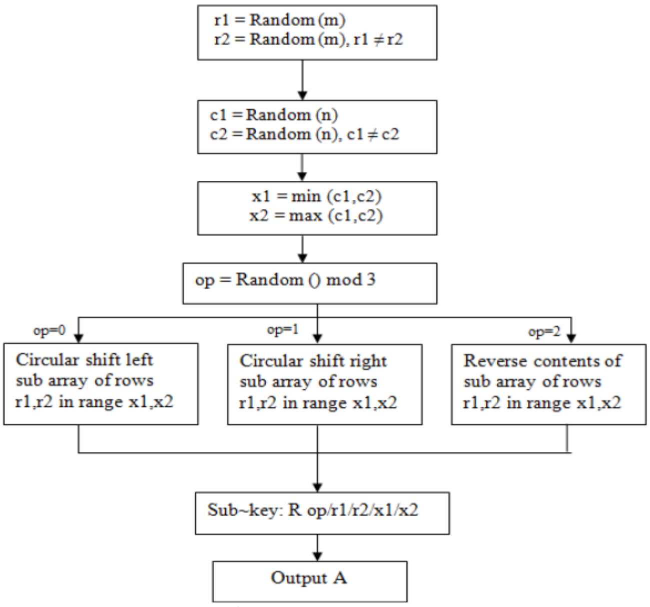

r1 = Random (m)
r2 = Random (m), r1 ≠ r2
c1 = Random (n)
c2 = Random (n), c1 ≠ c2
x1 = min (c1,c2)
x2 = max (c1,c2)
op = Random () mod 3
op=0
op=1
op=2
Circular shift left sub array of rows r1,r2 in range x1,x2
Circular shift right sub array of rows r1,r2 in range x1,x2
Reverse contents of sub array of rows r1,r2 in range x1,x2
Sub~key: R op/r1/r2/x1/x2
Output A

Figure 3: Row Transformation Process

Column Transformation

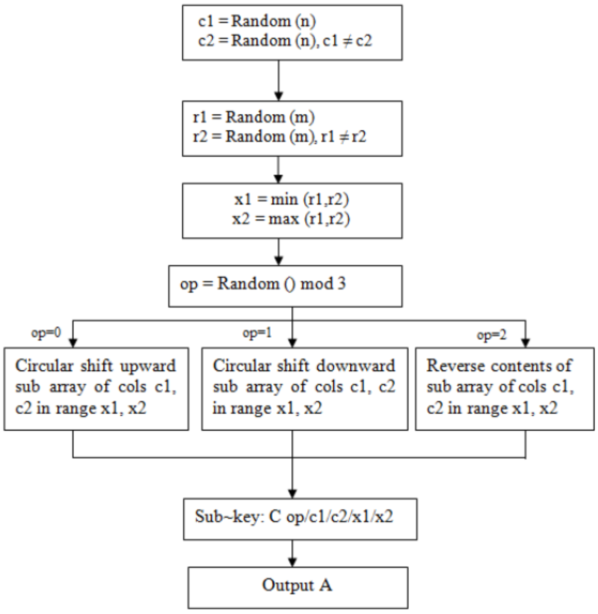

c1 = Random (n)
c2 = Random (n), c1 ≠ c2
r1 = Random (m)
r2 = Random (m), r1 ≠ r2
x1 = min (r1,r2)
x2 = max (r1,r2)
op = Random () mod 3
op=0
op=1
op=2
Circular shift upward sub array of cols c1, c2 in range x1, x2
Circular shift downward sub array of cols c1, c2 in range x1, x2
Reverse contents of sub array of cols c1, c2 in range x1, x2
Sub~key: C op/c1/c2/x1/x2
Output A

Figure 4: Column Transformation Process

B. Example of Encryption Process

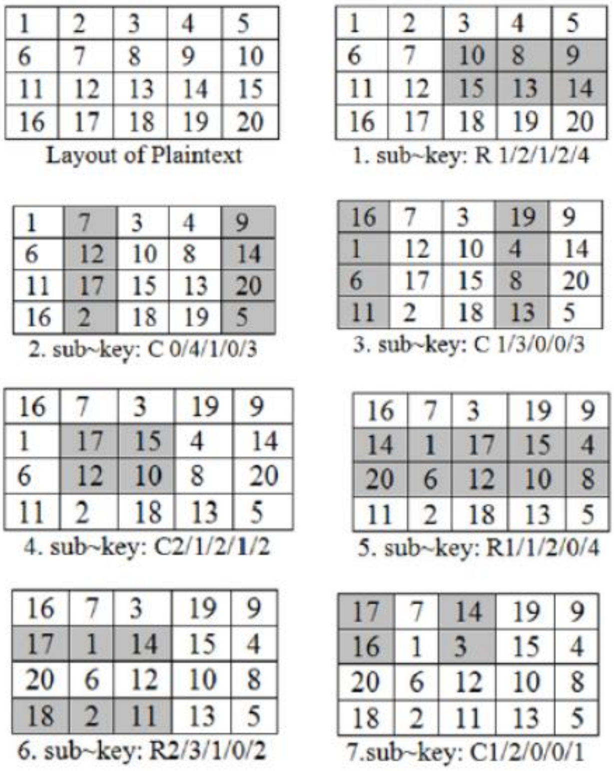
In this section, we show the detailed process of our encryption algorithm by an example. Let the data to be encrypted is taken as follows 1,2,3,4,5,6,7,8,9,10,11,12,13,14.15.16,17,18,19,20.

Let us set m = 4. n = 5. w = 7 i = 14, b = 01110. In vector b only 5 digits of the binary sequence is considered in the example. It depends on the user to consider how many bits to use without changing the actual bits of the value z. This provides irregularity and is used to increase the intensity of encryption. After the plain text is set into A, the layout of the matrix is shown in figure 4. After w operations and based on binary values in b, sub—keys recorded in the key—file is given in limit 5.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

R 1/2/1/2/4, C 0/4/1/0/3
C 1/3/0/0/3, C 2/1/2/1/2
R 1/1/2/0/4, R 2/3/1/0/2
C 1/2/0/0/1

The process of disordering or scrambling the matrix elements based on sub-keys is explained as follows.



1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

Layout of Plaintext

1	2	3	4	5
6	7	10	8	9
11	12	15	13	14
16	17	18	19	20

1. sub~key: R 1/2/1/2/4

1	7	3	4	9
6	12	10	8	14
11	17	15	13	20
16	2	18	19	5

2. sub~key: C 0/4/1/0/3

16	7	3	19	9
1	12	10	4	14
6	17	15	8	20
11	2	18	13	5

3. sub~key: C 1/3/0/0/3

16	7	3	19	9
1	17	15	4	14
6	12	10	8	20
11	2	18	13	5

4. sub~key: C2/1/2/1/2

16	7	3	19	9
14	1	17	15	4
20	6	12	10	8
11	2	18	13	5

5. sub~key: R1/1/2/0/4

16	7	3	19	9
17	1	14	15	4
20	6	12	10	8
18	2	11	13	5

6. sub~key: R2/3/1/0/2

17	7	14	19	9
16	1	3	15	4
20	6	12	10	8
18	2	11	13	5

7.sub~key: C1/2/0/0/1

The cipher text obtained after the entire process is given as 17,7,14,19,9,16,1,3,15,4,20,6,12,10,8,18,2,11,13,5

C. Decryption Process

The proposed algorithm is a kind of symmetric encryption algorithm, with decryption process is done by reversing the operations done in the encryption process. The cipher text is arranged into a matrix of the same order in the encryption as in and n.

The key file is partitioned by the R (row) operations & C (col) operations. A sub key starts from R or C operation, ends at the start of another R or C operation or the end of the key file.The sub keys are decrypted one by one from the last sub key to the first sub key.

For each sub key T. op, αl, α2, β1,β2 values are obtained whose terms are already explained in the encryption algorithm. Based on T value either R or C operation is decoded which are given as A = InverseRowTrans(A) and A = InverseColTrans(A).

In InverseRowTrans(A) depending on the value of 'op' transpositions are performed. For 0 right shifts, for I left shift and for 2 reverse operation is performed on the columns.

In InverseColTrans(A) depending, on the value of 'op' transpositions are performed. For 0 downward shift, for 1

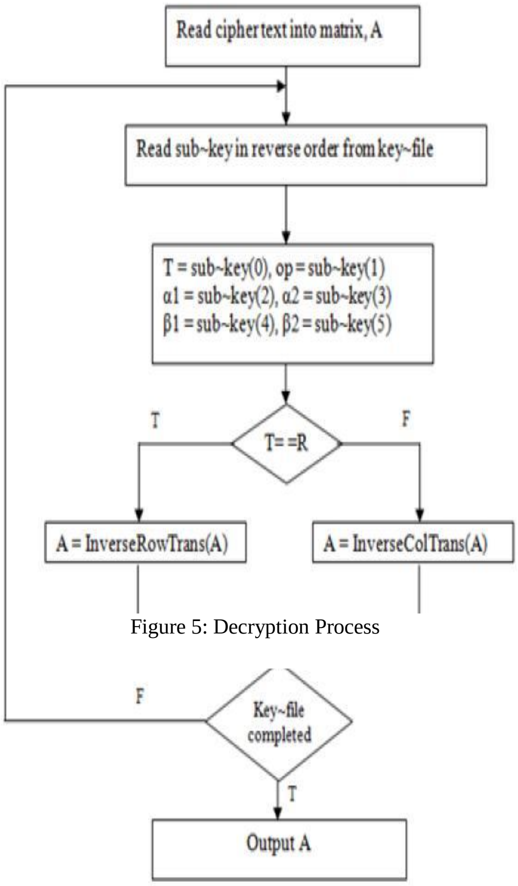

Read cipher text into matrix, A
Read sub-key in reverse order from key-file
T = sub-key(0), op = sub-key(1)
α1 = sub-key(2), α2 = sub-key(3)
β1 = sub-key(4), β2 = sub-key(5)
T
F
T==R
A = InverseRowTrans(A)
A = InverseColTrans(A)
F
Key-file completed
T
Output A

Figure 5: Decryption Process

upward shift and for 2 reverse operation is performed on the rows.

The process is done until the key file is completed and at the end of process matrix A contains the required original message i.e. plain text.

D. Example of Decryption Process

In this section, we show the detailed process of our decryption algorithm with an example. Consider the cipher text obtained after encryption as

17, 7, 14, 19, 9, 16, 1, 3, 15, 4, 20, 6, 12, 10, 8, 18, 2, 11, 13, 5.

17	7	14	19	9
16	1	3	15	4
20	6	12	10	8
18	2	11	13	5

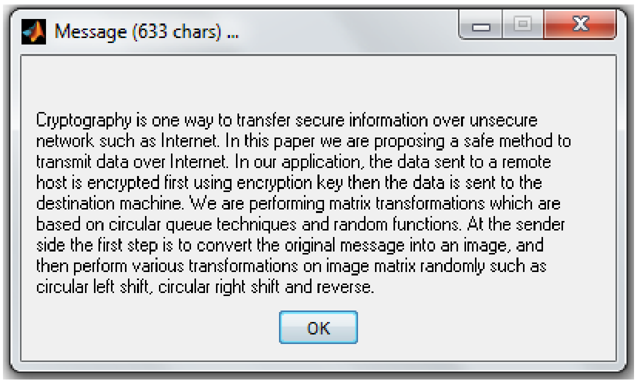
C 1/2/0/0/1, R 2/3/1/0/2
R 1/1/2/0/4, C 2/1/2/1/2
C 1/3/0/0/3, C 0/4/1/0/3
R 1/2/1/2/4

After performing transformations from sub-key files in reverse order we will get original matrix.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

IV. OUTPUT

We have partially completed the implementation of the project. We have managed to convert the message into an image. The message is taken as a input from the user.



Cryptography is one way to transfer secure information over unsecure network such as Internet. In this paper we are proposing a safe method to transmit data over Internet. In our application, the data sent to a remote host is encrypted first using encryption key then the data is sent to the destination machine. We are performing matrix transformations which are based on circular queue techniques and random functions. At the sender side the first step is to convert the original message into an image, and then perform various transformations on image matrix randomly such as circular left shift, circular right shift and reverse.

Figure 6: Original Message

Each character in the message is assigned a random RGB value which forms a single pixel in the image. The image size is determined from the length of the message i.e. no. of characters. The encrypted image is shown in Figure 7.

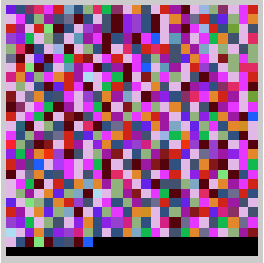


Figure 7: An Encrypted Image

The transformation algorithm will be applied on the image pixels matrix.

The reverse process is followed at the time of decrypting the message from an image.

V. CONCLUSION AND FUTURE WORK

In this approach the usage of random number selection firstly for generating binary sequence and, secondly for row (column) and column (row) selections, thirdly for selecting the operations for scrambling, avoids the regularity in the resultant cipher text which is transformed from plain text matrix; and hence improves the difficulty for decrypting. The algorithm can be applied to text encryption, image encryption, and multimedia encryption and so on.

REFERENCES

[1] F. Y. LI Min, (2005) "A new class of digital image scrambling algorithm based on the method of queue transformation", Computer Engineering. 01(31):148-149.

[2] W. Y. YE Yongwei and YANG Qinghua. (2003) "Magic cube encryption for digital image using chaotic sequence". Journal of Zhejiang University of Technology, 31(2):173-176.

[3] Singh, A., Gilhorta, R. (2011) Data security using private key encryption system based on arithmetic coding .International Journal of Network Security and its Applications (IJNSA), 3(3).

[4] Kui-HeYang and Shi-Jin Niu. -Data Safe Transmission Mechanism Based on Integrated Encryption Algorithm",Staffordshire University.

[5] G. J. HUANG Xiaosheng, (2007) "Image encryption algorithm based on compound chaotic sequence and wavelet transform". Computer Engineering, 14.