A Cryptographic Technique Involving Finite Fields and Logical Operators

Ayush Mittal* and Ravindra Kumar Gupta

S. R. K. University, Bhopal and Madhya Pradesh, India

Abstract

Objectives: The aim of this study is to propose a new encryption/decryption technique which involves logical operators and finite field. Methods/statistical analysis: The study proposes a new method for encrypting/decrypting messages which use logical operator XOR and finite field. The method involves sharing common keys between sender and receiver. Then converting plain text into ASCII values bit-form. XORing this with random matrix and then using finite field GF (2^5) for converting it into polynomial elements. These elements are then used for creating cipher text. Decrypting process is reverse of encrypting process. Findings: The technique uses 6×6 matrix form for encrypting messages which can be very difficult to crack with known plain text attacks. The use of XOR operations displaces bits in matrix which makes it difficult to break by brute force technique. Application/improvements: The technique is useful in areas where sensitive information is to be transferred like banking, military services. If the private key has got into third-party hands, the damage can be huge. To overcome this, the proposed algorithm can be used by randomizing the keys used for encrypting the messages which will provide different keys for encrypting different data blocks.

Keywords: Encryption, Decryption, Finite Fields, Logical Operator XOR.

1. Introduction

Rapid increase in wireless communications has led to increasing need for secure exchange of information between sender and receiver. Various places where securing transmitting information is necessary like banks, online currencies and such services needs extra precautions to be taken before transmitting information. Cryptography plays an important part in securing this information. For all aspects of information security, Cryptography is the study of mathematical techniques. Encryption is the process in cryptography which encodes some information in such way that hackers cannot read it. Only with a decryption code, it becomes readable [1–3].
The oldest art of communication is encryption/decryption process. Various secret or important messages are sent such that any third party cannot be able to read. Since many years military and government is using cryptography to establish secret communication. Due to the emerging use of communicating devices, nowadays cryptography is also being used in civilians systems.

Using an encryption algorithm the message or information is encrypted in an encryption scheme and convert the message into an unreadable form known as cipher text. The original message is not determine by anyone from encrypted message, is the main purpose of encryption. Using the corresponding decryption algorithm, only authorized party will be able to decode this cipher text. For unauthorized people, the stronger cipher text is harder to break.

1.1. Finite Fields

A finite field is a field with finite number of elements. If field ‘F’ is an abelian group under multiplication and addition then F is called as field. Finite field is also called as Galios field. The order of finite field i.e. number of elements of finite field is always equal to prime number or power of prime number. There exists exactly one finite field for each prime number. A finite field is set with certain rules with multiplication, addition, subtraction, and division operations. Galios field can of any degree \([4,5]\). For example, GF \((2^8)\). Its elements can be thought of as polynomials of degree 7 or less with binary coefficients (0 or 1). Addition of two field elements is addition of the two polynomials with coefficients being added modulo 2.

1.2. Logical Operators

Logical operators are used to combine two or more conditions or to complement the values under consideration. There are several logical operators like AND, OR, NOT, XOR, and XNOR which are used by various researchers [6]. Most logic gates have two inputs and one output and are based on Boolean algebra. Digital logic gates like XOR and XNOR are used in military cryptography [7] for generating linear feedback shift register. In our article, we will use properties of XNOR in combination of Galios filed GF \((2^n)\) for encryption/decryption.

2. Literature Survey

For encryption and decryption of message, in the field of cryptographic algorithm, there are huge amount of work done by the various researchers. In this section, some work are explained, which will be useful in the proposed work:

In Ref. [8], Sharma and Rehan proposed two-fold securities to the existing Hill cipher by using the elements of finite fields and logical operator. Author gives an algorithm which increases the security of Hill cipher. The proposed algorithm along with illustration
Involves the encryption and decryption of plaintext by making the use of elements of finite fields and logical XNOR operator.

In Ref. [9], Sharma and Sharma proposed to modify the Hill Cipher by multiplying $K$ and $K^{-1}$, one on the left side and another on right side of the plain text matrix in encryption as well as decryption and use the permutation on the binary bits. In addition, author use the elements of finite field for the purpose of more security.

In Ref. [10], Savas and Koc worked on to provide a concise perspective on designing architectures for efficient finite field arithmetic for usage in cryptography. Author present different architectures, methods and techniques for fast execution of cryptographic operations as well as high utilization of resources in the realization of cryptographic algorithms.

3. Methodology

Following [8–10] and others we will establish some new encryption and decryption algorithm of a message involving finite field and logical operations. Here we propose an algorithm which is based on the elements of finite fields, logical XOR operation, and bitwise complement operator. It will provide security in two levels. Therefore, there will be least possibilities of Brute force attack.

4. Algorithm

4.1. Encryption Algorithm

1. Sender and receiver shared a Secret key, where Secret key is the $(n - 1) \times (n - 1)$ non-singular matrix and $n$ is a positive integer.
2. Sender converts the Plaintext into pre-assigned numerical values (here we choose ASCII-7 code).
3. These numerical values are converted into 7-bit string using ASCII table, we get a matrix $M_1$ (say).
4. Consider a random matrix $A$ of order $(n - 1) \times (n - 1)$.
5. To perform XOR operation with each column of matrix $M_1$, randomly select columns/rows of $A_1$. Obtain a matrix $M_{XOR}$.
6. Applying bitwise complement operator on $M_{XOR}$, which gives the matrix $M'_{XOR}$.
7. Entries of $M'_{XOR}$ converted into the elements of GF ($2^n$). Multiply each entry of produced matrix by $\alpha^n$, which gives a matrix $K$. If $\alpha$ has the power less than $2^n - 1$ in matrix $M_3$ (say) then entries in $K$ are 0 otherwise 1 send this matrix $K$ to the receiver.
8. Applying mod ($2^n - 1$) on the entries of matrix $M_3$ for reducing the powers, we get another matrix $M_4$ (say).
9. Each entries of matrix $M_4$ are converted into 7-bit binary string. To obtain cipher text converts these binary strings into their corresponding text character using the ASCII code.
4.2. Decryption Algorithm

1. Convert the encrypted message into corresponding binary form (here we choose ASCII-7 code) of n-bits and arrange them in a matrix of order \((n - 1) \times (n - 1)\), then change into elements of GF \((2^n)\), we get a matrix \(D_1\) (say).
2. Those entries of \(D_1\) are multiplied by \(\alpha^{2n-1}\) which represents 1 in the corresponding key matrix \(K\) (say) and get a matrix \(D_2\) (say).
3. Each entries of \(D_2\) are multiplied by \(\alpha^{-n}\) and convert them into corresponding binary elements of n-bits using finite field GF \((2^n)\), we get a matrix \(D_4\) (say).
4. Applying bitwise complement operator, we get the matrix \(M_{XOR}\).
5. Recognizes the rows/columns of matrix \(A_1\) randomly preferred by the sender to perform XOR operation with each column of matrix \(M_{XOR}\) (say \(D_3\)).
6. Each entries of resulting matrix (find in step 5) are converted into their corresponding numerical values (here we choose ASCII-7 code). We get the matrix \(M\).
7. Each entries of \(M\) converted into their corresponding alphabet, we get Plaintext.

4.3. Example

4.3.1. Encryption Steps

1. Consider a non-singular matrix of order 6 × 6 randomly as follows:

\[
A = \begin{bmatrix}
5 & 3 & 13 & 2 & 14 & 37 \\
6 & 4 & 9 & 10 & 7 & 36 \\
8 & 11 & 12 & 15 & 16 & 35 \\
17 & 18 & 19 & 20 & 21 & 34 \\
22 & 23 & 24 & 25 & 26 & 33 \\
29 & 28 & 27 & 32 & 31 & 30 \\
\end{bmatrix}
\]

2. Consider a message

ENCRIPTION AND DECRYPTION ALGORITHM

3. Now, plain text message will be transformed into its ASCII code and writing each ASCII value in 6 × 6 matrix:

\[
M = \begin{bmatrix}
69 & 78 & 67 & 82 & 89 & 80 \\
84 & 73 & 79 & 78 & 32 & 65 \\
78 & 68 & 32 & 68 & 69 & 67 \\
82 & 89 & 80 & 84 & 73 & 79 \\
78 & 32 & 65 & 76 & 71 & 79 \\
82 & 73 & 84 & 72 & 77 & 32 \\
\end{bmatrix}
\]
4. Writing down ASCII number into its 7-bit string in form of matrix of order 6 × 6 say $M_1$:

\[
M_1 = \begin{bmatrix}
1000101 & 1001110 & 1000011 & 1010010 & 1011001 & 1010000 \\
1010100 & 1001001 & 1001111 & 1001110 & 0100000 & 1000001 \\
1001110 & 1000100 & 0100000 & 1000100 & 1000101 & 1000011 \\
1010010 & 1011001 & 1010000 & 1010100 & 1001001 & 1001111 \\
1001110 & 0100000 & 1000001 & 1001100 & 1000111 & 1001111 \\
1010010 & 1001001 & 1010100 & 1001000 & 1001101 & 0100000
\end{bmatrix}
\]

5. Writing down ASCII number into its 7-bit string in form of matrix of order 6 × 6 say $A_1$:

\[
A_1 = \begin{bmatrix}
0000101 & 0000011 & 0001101 & 0000010 & 0001110 & 0100101 \\
0000110 & 0000100 & 0001001 & 0001010 & 0000111 & 0100100 \\
0010001 & 0010010 & 0010111 & 0010000 & 0011011 & 0011000 \\
0010110 & 0010111 & 0011000 & 0011101 & 0011010 & 0100001 \\
0011101 & 0011100 & 0011011 & 0100000 & 0011111 & 0011110 \\
0010010 & 0010100 & 0010111 & 0100001 & 0011001 & 0011010
\end{bmatrix}
\]

6. Choose the rows/columns $C_1, R_2, C_3, R_4, C_5, R_6$ from the matrix $A_1$ at random to perform the logical XOR Operation with each column of matrix $M_1$.

(a) Select elements of $C_1$ of $A_1$ and performs logical operator XOR with first column of matrix $M_1$, we get

\[
00001010000110000100001000100110011110101
\]

XOR

\[
100010110100101000110100001110110001001111
\]

we get

\[
100000010100100110011011000110001111
\]

which is the first column of matrix $M_{\text{XOR}}$ in 7-bit form.

(b) Select elements of $R_2$ of $A_1$ and performs logical operator XOR with second column of matrix $M_1$, we get

\[
00001100000100000100100100001110100100
\]

XOR

\[
10011101001100010010110010100001001001
\]

we get

\[
1001000100110110101101100110101001111101101
\]

which is the second column of matrix $M_{\text{XOR}}$ in 7-bit form.
(c) Select elements of C_3 of A_1 and performs logical operator XOR with third column of matrix M_1, we get

\begin{align*}
00110100010001001100110000110111 \\
\text{XOR} \\
100011110011110100001010000111010100
\end{align*}
we get
\begin{align*}
10011101000110011010000111110011011111
\end{align*}
which is the third column of matrix M_{XOR} in 7-bit form.

(d) Select elements of R_4 of A_1 and performs logical operator XOR with fourth column of matrix M_1, we get

\begin{align*}
0010001001000100110010100001010100010 \\
\text{XOR} \\
101001010011101000100101010010011001000
\end{align*}
we get
\begin{align*}
100001110111001010111100000010110011101010
\end{align*}
which is the fourth column of matrix M_{XOR} in 7-bit form.

(e) Select elements of C_5 of A_1 and performs logical operator XOR with fifth column of matrix M_1, we get

\begin{align*}
0001111000011111001000001010101100001111111 \\
\text{XOR} \\
10110010100001110100110011110010011111001101
\end{align*}
we get
\begin{align*}
1010111010011110100110001111101000011111010
\end{align*}
which is the fifth column of matrix M_{XOR} in 7-bit form.

(f) Select elements of R_6 of A_1 and performs logical operator XOR with sixth column of matrix M_1, we get

\begin{align*}
0011101100001111001100000001111100111110 \\
\text{XOR} \\
10100001000001110100011111111011000000
\end{align*}
we get
\begin{align*}
10011011011011011000011111110100000111110
\end{align*}
which is the sixth column of matrix M_{XOR} in 7-bit form.
Therefore, M_{XOR} is:
A Cryptographic Technique Involving Finite Fields and Logical Operators

\[
M_{\text{XOR}} = \begin{bmatrix}
100000 & 100100 & 100110 & 100011 & 101011 & 100110 \\
101000 & 100110 & 100011 & 101110 & 010011 & 101110 \\
100011 & 101010 & 010110 & 100111 & 101011 & 101100 \\
101000 & 101010 & 100110 & 101100 & 110000 & 101110 \\
101100 & 010011 & 101100 & 101110 & 101001 & 100000 \\
100111 & 110110 & 101011 & 100000 & 110100 & 101000
\end{bmatrix}
\]

7. Applying bitwise complement operator on \(M_{\text{XOR}}\), which gives the matrix \(M'_{\text{XOR}}\), as follows:
Therefore, \(M'_{\text{XOR}}\) is:

\[
M'_{\text{XOR}} = \begin{bmatrix}
011111 & 011011 & 011001 & 011110 & 010100 & 011001 \\
010110 & 011001 & 011010 & 010011 & 101100 & 010010 \\
011100 & 010100 & 101001 & 010100 & 010101 & 010011 \\
010011 & 101100 & 010010 & 011010 & 011111 & 001000 \\
011000 & 001000 & 011000 & 010011 & 011110 & 100000 \\
011000 & 011000 & 011000 & 011111 & 010111 & 100001
\end{bmatrix}
\]

8. Converts the above entries to the elements of GF \( (2^7) \), we get

\[
M_2 = \begin{bmatrix}
\alpha^{119} & \alpha^7 & \alpha^6 & \alpha^{23} & \alpha^{17} & \alpha^{109} \\
\alpha^{77} & \alpha^{109} & \alpha^{104} & \alpha^{17} & \alpha^{34} & \alpha^{29} \\
\alpha^{104} & \alpha^{109} & \alpha^{104} & \alpha^{17} & \alpha^{113} & \alpha^{45} \\
\alpha^{23} & \alpha^{33} & \alpha^{23} & \alpha^{119} & \alpha^{19} & \alpha^{4} \\
\alpha^{45} & \alpha^{34} & \alpha^{68} & \alpha^{68} & \alpha^{29} & \alpha^{117} \\
\alpha^{11} & \alpha^{44} & \alpha^{11} & \alpha^{112} & \alpha^{77} & \alpha^{126}
\end{bmatrix}
\]

9. Multiply \(M_2\) by \(\alpha^7\), we get

\[
M_3 = \begin{bmatrix}
\alpha^{126} & \alpha^{66} & \alpha^{64} & \alpha^{60} & \alpha^{40} & \alpha^{116} \\
\alpha^{84} & \alpha^{116} & \alpha^{111} & \alpha^{26} & \alpha^{41} & \alpha^{126} \\
\alpha^{111} & \alpha^{116} & \alpha^{101} & \alpha^{124} & \alpha^{120} & \alpha^{52} \\
\alpha^{30} & \alpha^{40} & \alpha^{30} & \alpha^{126} & \alpha^{26} & \alpha^{11} \\
\alpha^{52} & \alpha^{41} & \alpha^{75} & \alpha^{75} & \alpha^{36} & \alpha^{124} \\
\alpha^{18} & \alpha^{71} & \alpha^{18} & \alpha^{119} & \alpha^{84} & \alpha^{133}
\end{bmatrix}
\]
10. Applying mod 127 to $M_3$, we get

$$M_4 = \begin{bmatrix}
\alpha^{126} & \alpha^{86} & \alpha^{94} & \alpha^{30} & \alpha^{24} & \alpha^{116} \\
\alpha^{84} & \alpha^{116} & \alpha^{111} & \alpha^{26} & \alpha^{41} & \alpha^{36} \\
\alpha^{111} & \alpha^{116} & \alpha^{101} & \alpha^{24} & \alpha^{120} & \alpha^{52} \\
\alpha^{30} & \alpha^{40} & \alpha^{30} & \alpha^{126} & \alpha^{26} & \alpha^{11} \\
\alpha^{52} & \alpha^{41} & \alpha^{75} & \alpha^{75} & \alpha^{36} & \alpha^{124} \\
\alpha^{18} & \alpha^{71} & \alpha^{18} & \alpha^{119} & \alpha^{84} & \alpha^{6}
\end{bmatrix}$$

and choose the key matrix $K$ such that if power of $\alpha$ in $M$ is greater than 127, then entry in the key matrix is taken 1 otherwise 0. Therefore, $K$ is

$$K = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

11. The cipher matrix is

$$M_4 = \begin{bmatrix}
\alpha^{126} & \alpha^{86} & \alpha^{94} & \alpha^{30} & \alpha^{24} & \alpha^{116} \\
\alpha^{84} & \alpha^{116} & \alpha^{111} & \alpha^{26} & \alpha^{41} & \alpha^{36} \\
\alpha^{111} & \alpha^{116} & \alpha^{101} & \alpha^{24} & \alpha^{120} & \alpha^{52} \\
\alpha^{30} & \alpha^{40} & \alpha^{30} & \alpha^{126} & \alpha^{26} & \alpha^{11} \\
\alpha^{52} & \alpha^{41} & \alpha^{75} & \alpha^{75} & \alpha^{36} & \alpha^{124} \\
\alpha^{18} & \alpha^{71} & \alpha^{18} & \alpha^{119} & \alpha^{84} & \alpha^{6}
\end{bmatrix}$$

12. Convert the element of matrix $M_4$ into 7-bit binary string, we get cipher matrix $C$ (say)

$$C = \begin{bmatrix}
1000001 & 1011001 & 1010011 & 1000100 & 1111000 & 1010110 \\
1110111 & 1010110 & 1001011 & 1100101 & 1101011 & 1100110 \\
1001011 & 1010110 & 1110110 & 1111000 & 1111110 & 1101001 \\
1000100 & 1110100 & 1000100 & 1000001 & 1100101 & 0110000 \\
1101001 & 1101011 & 1101010 & 1101010 & 1100110 & 1110001 \\
1010000 & 0110110 & 1010000 & 0111111 & 1110111 & 1000000
\end{bmatrix}$$
13. Using the ASCII code, all binary elements of the matrix C are coded to the corresponding text characters. Sequence of these characters (collect row-wise) gives the Cipher text as follows:

AYSDxVwVKekfKVvx~iDtDAe0ikjffqP6P?w@

Via public channel this cipher text is sent to the receiver.

4.3.2 Decryption Steps

1. Converts the cipher text AYSDxVwVKekfKVvx~iDtDAe0ikjffqP6P?w@ into 7-bit string using ASCII code. Put these 36 binary numbers in the form of a $6 \times 6$ matrix as follows:

$$D = \begin{bmatrix}
1000001 & 1011001 & 1010011 & 1000100 & 1111000 & 1010110 \\
1110111 & 1010110 & 1001011 & 1100101 & 1101011 & 1100110 \\
1001011 & 1010110 & 1110110 & 1111000 & 1111110 & 1111001 \\
1000100 & 1110100 & 1000100 & 1000001 & 1101001 & 1101010 \\
1101010 & 1100110 & 1110001 & 0111111 & 1110111 & 1111000 \\
1010000 & 0110110 & 1010000 & & & & 
\end{bmatrix}$$

2. Above 7-bit binary number converted into the elements of GF $(2^7)$ as follows:

$$D_1 = \begin{bmatrix}
\alpha^{126} & \alpha^{86} & \alpha^{94} & \alpha^{30} & \alpha^{24} & \alpha^{116} \\
\alpha^{84} & \alpha^{116} & \alpha^{111} & \alpha^{26} & \alpha^{41} & \alpha^{36} \\
\alpha^{111} & \alpha^{116} & \alpha^{101} & \alpha^{24} & \alpha^{120} & \alpha^{52} \\
\alpha^{30} & \alpha^{40} & \alpha^{30} & \alpha^{126} & \alpha^{26} & \alpha^{11} \\
\alpha^{52} & \alpha^{41} & \alpha^{75} & \alpha^{75} & \alpha^{36} & \alpha^{124} \\
\alpha^{18} & \alpha^{71} & \alpha^{18} & \alpha^{119} & \alpha^{84} & \alpha^6 
\end{bmatrix}$$

3. Multiply $\alpha^{127}$ into those entries of $D_1$ which represents 1 in the corresponding key matrix $K$, we get:

$$D_2 = \begin{bmatrix}
\alpha^{126} & \alpha^{86} & \alpha^{94} & \alpha^{30} & \alpha^{24} & \alpha^{116} \\
\alpha^{84} & \alpha^{116} & \alpha^{111} & \alpha^{26} & \alpha^{41} & \alpha^{36} \\
\alpha^{111} & \alpha^{116} & \alpha^{101} & \alpha^{24} & \alpha^{120} & \alpha^{52} \\
\alpha^{30} & \alpha^{40} & \alpha^{30} & \alpha^{126} & \alpha^{26} & \alpha^{11} \\
\alpha^{52} & \alpha^{41} & \alpha^{75} & \alpha^{75} & \alpha^{36} & \alpha^{124} \\
\alpha^{18} & \alpha^{71} & \alpha^{18} & \alpha^{119} & \alpha^{84} & \alpha^6 
\end{bmatrix}$$
4. Multiply $D_2$ by $\alpha^{-7}$, we get

$$D_3 = \begin{bmatrix}
\alpha^{119} & \alpha^{79} & \alpha^{87} & \alpha^{23} & \alpha^{17} & \alpha^{109} \\
\alpha^{77} & \alpha^{109} & \alpha^{104} & \alpha^{19} & \alpha^{34} & \alpha^{29} \\
\alpha^{104} & \alpha^{109} & \alpha^{94} & \alpha^{17} & \alpha^{113} & \alpha^{45} \\
\alpha^{23} & \alpha^{33} & \alpha^{23} & \alpha^{119} & \alpha^{19} & \alpha^4 \\
\alpha^{45} & \alpha^{34} & \alpha^{68} & \alpha^{68} & \alpha^{29} & \alpha^{117} \\
\alpha^{11} & \alpha^{64} & \alpha^{11} & \alpha^{112} & \alpha^{77} & \alpha^{126}
\end{bmatrix}$$

5. Convert the above elements into the 7-bit binary form by finite field GF($2^7$), we get

$$D_4 = \begin{bmatrix}
0111111 & 0110111 & 0110001 & 0111100 & 0101000 & 0110010 \\
0101101 & 0110010 & 0111001 & 0100011 & 1011000 & 0100010 \\
0111001 & 0101100 & 0111100 & 0111111 & 0100011 & 0010000 \\
0100111 & 1011000 & 0100110 & 0100110 & 0100010 & 0101111 \\
0111000 & 0010010 & 0110000 & 0000111 & 0101111 & 0000001
\end{bmatrix}$$

6. Now applying bitwise complement operator, which gives the matrix $M_{\text{XOR}}$ (say $D_5$) as follows:

$$D_5 = \begin{bmatrix}
1000000 & 1001000 & 1001100 & 1000011 & 1010111 & 1001101 \\
1010010 & 1001101 & 1000110 & 1011100 & 0100111 & 1011001 \\
1001110 & 1001010 & 1010111 & 1010101 & 1011000 & 1000011 \\
1000111 & 1001000 & 1000000 & 1010101 & 1010101 & 1011011 \\
1011000 & 0100111 & 1011001 & 1011011 & 1011011 & 1010000 \\
1001111 & 1101101 & 1001111 & 1101010 & 1010010 & 0111110
\end{bmatrix}$$

7. Choose the rows/columns $C_1$, $R_2$, $C_3$, $R_4$, $C_5$, $R_6$ (selected by sender to perform the logical XOR operation with the column of the matrix $D_5$ and send by a separate communication) from the matrix $A_1$ at random to perform the logical XOR Operation with each column of matrix $D_5$.

(a) Select elements of $C_1$ of $A_1$ and performs logical operator XOR with first column of matrix $D_5$, we get

$$00001010000110000100000100010011001101101$$

XOR

$$10000001010010001101000111011000100101111$$

we get
(a) Select elements of \( R_1 \) of \( A_1 \) and perform logical operator XOR with first column of matrix \( M_1 \), we get

\[
0010110101001000111101001101001000100011101001001
\]

which is the first column of matrix \( M_1 \) in 7-bit form.

(b) Select elements of \( R_2 \) of \( A_1 \) and performs logical operator XOR with second column of matrix \( M_1 \), we get

\[
00001100001000010010001010000011101001000
\]

XOR

\[
10010001001101011001111101001101001011101101000
\]

we get

\[
1001110100100110010010001100100000110001010001
\]

which is the second column of matrix \( M_1 \) in 7-bit form.

(c) Select elements of \( C_3 \) of \( A_1 \) and performs logical operator XOR with third column of matrix \( M_1 \), we get

\[
000111000011001000100011001000100011101001111
\]

XOR

\[
1010110100110110011011010011010011111011010010
\]

we get

\[
101001011101110011000010100001000011011010001
\]

which is the third column of matrix \( M_1 \) in 7-bit form.

(d) Select elements of \( R_4 \) of \( A_1 \) and performs logical operator XOR with fourth column of matrix \( M_1 \), we get

\[
0010001000100100010011001010000101010100010
\]

XOR

\[
1000111010010111010010010100010110011011101010
\]

we get

\[
10100101001110110100100101001000110010010001
\]

which is the fourth column of matrix \( M_1 \) in 7-bit form.

(e) Select elements of \( C_5 \) of \( A_1 \) and performs logical operator XOR with fifth column of matrix \( M_1 \), we get

\[
0001110000110010000001010100110100011111
\]

XOR

\[
1010111010011110101011011001011011011010010
\]

we get

\[
1011001010000010001011000110011111001101101
\]

which is the fifth column of matrix \( M_1 \) in 7-bit form.
(f) Select elements of $R_6$ of $A_1$ and performs logical operator XOR with sixth column of matrix $M_1$, we get

\[ \begin{array}{cccccccc}
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
\end{array} \]

\[ \text{XOR} \]

\[ \begin{array}{cccccccc}
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\end{array} \]

we get

\[ \begin{array}{cccccccc}
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\end{array} \]

which is the sixth column of matrix $M_1$ in 7-bit form.

Hence matrix $M_1$ is

\[
M_1 = \begin{bmatrix}
1000101 & 1001110 & 1000011 & 1010010 & 1011001 & 1010000 \\
1010100 & 1001001 & 1000100 & 1001111 & 1001111 & 0100000 \\
1001110 & 1000100 & 0100000 & 1000100 & 1000101 & 1000011 \\
1010010 & 1011001 & 1010000 & 1010100 & 1001001 & 1001111 \\
1001110 & 0100000 & 1000001 & 1001100 & 1000111 & 1000001 \\
1010010 & 1001001 & 1010100 & 1001000 & 1001101 & 0100000 \\
\end{bmatrix}
\]

8. Convert these binary entries of $M_1$ into corresponding numerical values from ASCII-7 table, we get

\[
M = \begin{bmatrix}
69 & 78 & 67 & 82 & 89 & 80 \\
84 & 73 & 79 & 78 & 32 & 65 \\
78 & 68 & 32 & 68 & 69 & 67 \\
82 & 89 & 80 & 84 & 73 & 79 \\
78 & 32 & 65 & 76 & 71 & 79 \\
82 & 73 & 84 & 72 & 77 & 32 \\
\end{bmatrix}
\]

9. The matrix $M$ is ASCII code of characters. Converting ASCII codes to characters we will get our plain text. Reading Row-wise each character we will get final plain text message as follows:

**ENCRYPTION AND DECRYPTION ALGORITHM**

**5. Result and Discussion**

The technique discussed in this article to encrypt/decrypt messages cannot be broken by brute force attack. The matrix $6 \times 6$ makes it more difficult. Due to absence of direct relation between plain text and cipher text, cracking encrypted messages is very difficult
even if key matrix information is known. Several iterations of logical operations of XOR displace all the bits which increase strength of these methods by many folds.

6. Conclusion

This study proposes a strong algorithm for encrypting/decrypting messages for private key cryptography. For increasing security level, the technique uses finite fields with logical operator XOR. As private key is used for encrypting/decrypting messages it is very hard to crack the message without the key by known plain text attacks. Randomization is also possible which may include encrypting different data block with different key, which are produced from the secret key and shared between communicating parties. Hence the proposed algorithm had strength.

References

1. Behrouz AF. Cryptography & network security. McGraw Hill Education. 2007, 1–239.
2. Atul K. Cryptography and network security. Tata McGraw Hill: New Delhi. 2008, 1–480.
3. William S. Cryptography and network security principles and practices. Prentice Hall. 2005, 592.
4. Hun-Chen C, Cheng YJ. A new cryptography systems and its VISI realization. Journal of Systems Architecture. 2003; 49(7–9), 355–367.
5. Martine H, Thomas J. Breaking the stream cipher F-FCSR-H and F-FCSR-16 in real time. Journal of Cryptology. 2011; 24(3), 427–445.
6. Stakhov AP. The golden matrices and a new kind of cryptography. Chaos, Solution and Fractals. 2007; 32(3), 1138–1146.
7. Hun-Chen C, Cheng YJ, Juin G. Design a new cryptography system. Lecture Notes in Computer Science. 2002; 2532, 211–219.
8. Sharma PL, Rehan M. On security of hill cipher using finite fields. International Journal of Computer Applications. 2013; 71(4), 30–33.
9. Sharma PL, Sharma S. An application of finite field in hill cipher. International Journal of Technology. 2014; 4(1), 248–251.
10. Savas E, Koc C. Finite field arithmetic for cryptography. IEEE Circuits and Systems Magazine. 2010; 10(2), 40–56.