ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1205

A Cryptographic Technique Involving Substitution Algorithm and Logical Operator

Lalit Chaurasiya[1]; S.S. Shrivastava[2]

[1,2]Institute for Excellence in Higher Education, Bhopal

Publication Date: 2025/03/27

Abstract: The purpose of this paper is to implement an encryption and decryption algorithm for symmetric key cryptography that combines substitution method and a newly developed cryptographic technique utilizing the Xor cipher. An Xor cipher is exceptionally useful for encryption and decryption processes because it transforms bits in relation to another string of binary data (the key) that bears no relation to the primary data. This fact leads to considerable ambiguity on the part of attackers trying to guess the value of the key or the plain text, making the system more robust against security attacks.

Keywords: *Substitution algorithm, Xor cipher, Encryption and Decryption.*

How to Cite: Lalit Chaurasiya; S.S. Shrivastava (2025). A Cryptographic Technique Involving Substitution Algorithm and Logical Operator. *International Journal of Innovative Science and Research Technology,* 10(2), 1201-1205. https://doi.org/10.38124/ijisrt/25mar1205

I. INTRODUCTION

The growth in digital communication and the growing threat of unauthorized access, data transmitted protection has become a necessary requirement. Cryptography safeguards information based on mathematical principles, primarily through encryption, which transforms data into an unreadable form, and decryption, which transforms it back to its original form. Cryptographic algorithms are typically divided into two categories: symmetric and asymmetric. Symmetric encryption relies on a single key for encoding and decoding data, whereas asymmetric encryption relies on a pair of keys—a public key to encrypt and a private key to decrypt. With digital interactions characterizing modern life, cryptography is the linchpin of protecting data from unauthorized access and ensuring confidentiality. Various encryption and decryption methods are employed to protect digital data, deterring unauthorized alteration upon receipt. Apart from conventional approaches, we propose a new encryption and decryption process using a substitution algorithm in conjunction with logical operator to improve data protection by a unique and advanced mechanism.

➢ *Substitution Algorithm:*

A substitution algorithm in cryptography is a way to disguise a message by replacing each letter or symbol in the original message with another letter or symbol. A substitution algorithm (also known as a substitution cipher) is a method of encryption where each element (usually a letter) of the plaintext is replaced with another element (typically another letter, symbol, or group of symbols).

The basic idea is to substitute one set of symbols for another in a way that makes the original message hard to decipher without knowing the specific substitution scheme. The goal is to make the message unreadable to anyone who doesn't know the rule used for substitution.

- *Xor Cipher:*

Xor operations are most importance in cryptography because they have the potential to alter bits according to another binary string (the key) without any direct relation to the original information. The uncertainty makes it difficult for attackers to deduce the key or plaintext, particularly when Xor rounds are invoked repeatedly. One of the key advantages of Xor is that it alters each bit uniquely, making encryption effective

II. LITERATUEW REVIEW

- Mittal et. al. [7] established an algorithm for encrypting and decrypting messages using a symmetric key cryptosystem based on a Genetic Algorithm. In their proposed algorithm, they used a substitution method along with genetic crossover and mutation techniques.
- Garg [4] established a new encryption technique employing XOR Cipher for data encryption and decryption in four steps. The method first maps characters to binary, then generates a structured binary string, applies the XOR operation, and finally converts the result back to text. The security of the process is enhanced using Extended ASCII, ensuring reliable encryption.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1205

Experimental results demonstrate the technique's excellent encryption capability.

- Sindhuja et. al. [9] developed a new cryptographic technique for securing information. In their paper, they proposed a symmetric key cryptosystem based on Genetic Algorithms (GA) for encryption and decryption. They converted the plain text and the user input (key) into text and key matrices, respectively. They then created an additive matrix by adding the text matrix and key matrix. A linear substitution function was applied to this additive matrix to generate an intermediate cipher. GA functions, including crossover and mutation, were then used on the intermediate cipher to produce the final ciphertext. The proposed algorithm involved two main steps: substitution followed by genetic crossover and mutation.

III. METHODOLOGY

- To encrypt and decrypt the message, we will use substitution algorithm, which gives the intermediate cipher and then incorporate Xor cipher. We get the cipher text.
- To decrypt the message firstly we will use Xor cipher, after that applying substitution algorithm and matrix addition operation, we get the original plain text.

➢ *Additionally, We Have Used the Tables Which Are as Follows:*

Table 1 Conversion Table

Alphabet/symbol	Numerical value	Alphabet/ symbol	Numerical value
#	0	P	16
A	1	Q	17
B	2	R	18
C	3	S	19
D	4	T	20
E	5	U	21
F	6	V	22
G	7	W	23
H	8	X	24
I	9	Y	25
J	10	Z	26
K	11	@	27
L	12	&	28
M	13	%	29
N	14	&	30
O	15		

Table 2 Xor Operation

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

- Apart from these, another table was used in this paper whose name is EBCID Code (table 2.1)

IV. ALGORITHM

➢ *Encryption*:

- *Generate a key matrix.*

✓ Take block size of matrix (say n = 4)
✓ Choose the input key.
✓ We convert the input key which is taken by us into equivalent to EBCDIC code.
✓ After this we convert the EBCID code into equivalent to binary code.
✓ Now we apply the right shift operation by 2 bits on the binary string.
✓ After performing the operation, we get the binary set which we convert into equivalent to decimal number.
✓ After doing this we get the key matrix (say K).

- *Convert The Chosen Plaintext into A Text Matrix.*

✓ Consider the plain text.
✓ Convert the plaintext into block size of n.
✓ We get their EBCID equivalent text matrix.

- *By Subtracting Key Matrix from Plain Text Matrix Produce a Subtractive Matrix.*

✓ Apply substitution algorithm on the subtractive matrix.
✓ we apply the substitution algorithm y = c(x) = (ax + b)(mod n) on the subtractive matrix.
✓ After applying the subtractive algorithm, we get the intermediate ciphertext.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1205

- ✓ Arrange the characters of intermediate ciphertext in the matrix form.
- ✓ Convert each character of obtained matrix into corresponding equivalent EBCDIC binary code say Ic(binary).
- ✓ Calculate K (XOR) $I_{C[binary]} = C_{ipher}$ (say).
- ✓ Convert each element of C_{ipher} into equivalent EBCDIC character. We obtained the final ciphertext.

➢ *Decryption*:

- Consider the cipher text.
- Convert the cipher text in the matrix form.
- Now convert this character into equivalent EBCDIC binary code, we get another matrix C_{ipher} (Say).
- Calculate C_{ipher}(XOR)K, we arrive at $I_{C[binary]}$ (Say).
- Convert each elements of the matrix $I_{C[binary]}$ into equivalent EBCDIC character code, we get the intermediate ciphertext.
- Convert the intermediate ciphertext into equivalent numeric values using predefined table.
- Arrange the numeric values into a matrix of 4 × 4 orders.
- To generate the original plain text, reverse the substitution algorithm and matrix addition operation.

- ✓ After applying the reverse substitution function $c^{-1}(y) = a^{-1}(y - b) \pmod n$, we move on the matrix (say A) of order n.
- ✓ Add the key matrix (say K) with the matrix A, we get a new matrix (say E).
- ✓ After getting a new matrix, we convert this matrix equivalent to EBCDIC character code.
- ✓ We get the original plain text.

- *Example:*

This example is based on the algorithm given in the section 4.3 involving substitution algorithm, inverse modulo system and logical operator XOR. The substitution algorithm is expressed as $y = C(x) = (ax + b) \bmod n$, where x represents the numerical value of the plaintext letter, and a and b are selected integers. The inverse of this substitution algorithm is given by

$C^{-1}(y) = a^{-1}(y - b) \bmod n$.

➢ *Encryption:*

- *Generation of Key Matrix:*

- ✓ Take block size of matrix (say n = 4).
- ✓ Choose the input key

➢ *Research in Math*

- Convert the input key into corresponding equivalent EBCDIC numeric code: 217 197 226 197 193 217 195 200 64 201 213 64 212 193 227 200

- Convert above numeric code into equivalent binary form:

11011001	11000101	11100010	11000101
11000001	11011001	11000011	11001000
01000000	11001001	11010101	01000000
11010100	11000001	11100011	11001000

- Now we apply the right shift operation by 2 bits on the above binary streams, as follows:

00110110	00110001	00111000	00110001
00110000	00110110	00110000	00110010
00010000	00110010	00110101	00010000
00110101	00110000	00111000	00110010

- After performing the operation, we get the binary set. Now convert this binary set into equivalent EBCDIC numeric code (Table 2.1), we get the key matrix (say K).

$$K = \begin{bmatrix} 54 & 49 & 56 & 49 \\ 48 & 54 & 48 & 50 \\ 16 & 50 & 53 & 16 \\ 53 & 48 & 56 & 50 \end{bmatrix}$$

➢ *Convert the Chosen Plaintext into a Text Matrix:*

- *Consider the Plaintext:*

"Since joining the college in 2021, the teaching assistant has helped to transform the maths department and perceptions of the subject across the school".

- Pick up a part of this message as given below:

- *Maths Department*

Convert the plain text into a matrix of block size n = 4 and write their equivalent EBCDIC numeric code, we get the following matrix (say E):

$$E = \begin{bmatrix} 212 & 193 & 227 & 200 \\ 226 & 64 & 196 & 197 \\ 215 & 193 & 217 & 227 \\ 212 & 197 & 213 & 227 \end{bmatrix}$$

- *Obtaining the Subtractive Matrix:*

By subtracting key matrix K from plaintext matrix E produce a subtractive matrix (say A) as follows:

$$A = \begin{bmatrix} 158 & 144 & 171 & 151 \\ 178 & 10 & 148 & 147 \\ 199 & 143 & 164 & 211 \\ 159 & 149 & 157 & 177 \end{bmatrix}$$

- *Producing the Ciphertext:*

To produce a ciphertext, apply substitution algorithm on the subtractive matrix A.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1205

✓ Here, we apply the substitution algorithm

C(x) = (ax + b)(mod 31), where a = 5, b = 2.

e.g. C (158) = (5 × 158 + 2) (mod 31) = 17

Similarly, we get

17, 9, 20, 13, 24, 21, 29, 24, 5, 4, 16, 3, 22, 3, 12, 19

✓ Now convert the above numbers into equivalent character/symbol by using Table 2.2, we arrive at following intermediate ciphertext:

➢ *Qitmxu%Xedpcvcls*

- Convert the above character in the matrix form (say intermediate ciphertext matrix I_C):

$$I_C = \begin{bmatrix} Q & I & T & M \\ X & U & \% & X \\ E & D & P & C \\ V & C & L & S \end{bmatrix}$$

- Convert each character of I_C into corresponding equivalent EBCDIC binary code, we get following matrix (say $I_{C[binary]}$:

$$I_{C[binary]} = \begin{bmatrix} 11011000 & 11001001 & 11100011 & 11010100 \\ 11100111 & 11100100 & 01101100 & 11100111 \\ 11000101 & 11000100 & 11010111 & 11000011 \\ 11100101 & 11000011 & 11010011 & 11100010 \end{bmatrix}$$

- Now, we apply XOR operation as follows:

$C_{ipher} = K \text{ (XOR) } I_{C[binary]}$

$$= \begin{bmatrix} 11101110 & 11111000 & 11011011 & 11100101 \\ 11010111 & 11010010 & 01011100 & 11010101 \\ 11010101 & 11110110 & 11100010 & 11010011 \\ 11010000 & 11110011 & 11101011 & 11010000 \end{bmatrix}$$

- Now convert each character of above matrix into equivalent EBCDIC character code (Table 2.1), we get final ciphertext matrix as follows:

$$C_{ipher} = \begin{bmatrix} Ó & 8 & û & V \\ P & K & * & N \\ N & 6 & S & L \\ } & 3 & Ô & } \end{bmatrix}$$

- Now, write characters of above matrix row-wise we arrive at following final ciphertext:

Ó8ûVPK * NN6SL}3Ô}

➢ *Decryption*:

- Consider the ciphertext:

Ó8ûVPK * NN6SL}3Ô}

- Convert the above character in the matrix form as follows:

$$\begin{bmatrix} Ó & 8 & û & V \\ P & K & * & N \\ N & 6 & S & L \\ } & 3 & Ô & } \end{bmatrix} = C_{ipher} \text{ (Say)}$$

- Now convert each character of this matrix into equivalent EBCDIC character code by using Table 2.1, we arrive at:

$$\begin{bmatrix} 11101110 & 11111000 & 11011011 & 11100101 \\ 11010111 & 11010010 & 01011100 & 11010101 \\ 11010101 & 11110110 & 11100010 & 11010011 \\ 11010000 & 11110011 & 11101011 & 11010000 \end{bmatrix} = C_{ipher} \text{ (Say)}$$

- Now apply the following operations:

C_{ipher}(XOR)K

$$= \begin{bmatrix} 11011000 & 11001001 & 11100011 & 11010100 \\ 11100111 & 11100100 & 01101100 & 11100111 \\ 11000101 & 11000100 & 11010111 & 11000011 \\ 11100101 & 11000011 & 11010011 & 11100010 \end{bmatrix}$$

$= I_{C[binary]}$ (Say)

- Now convert the above matrix into equivalent EBCDIC character code by using Table 2.1, we get following matrix of intermediate ciphertext:

$$\begin{bmatrix} Q & I & T & M \\ X & U & \% & X \\ E & D & P & C \\ V & C & L & S \end{bmatrix} = I_C \text{ (say)}$$

- Now arrange the elements of above matrix in row-wise, we arrive at following intermediate ciphertext: QITMXU%XEDPCVCLS

- Convert the above ciphertext into equivalent numeric values using Table 2.2, we get:

17, 9, 20, 13, 24, 21, 29, 24, 5, 4, 16, 3, 22, 3, 12, 19

- Arrange the above numeric values into a matrix of 4 × 4 orders as follows:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar1205

$$\begin{bmatrix} 17 & 9 & 20 & 13 \\ 24 & 21 & 29 & 24 \\ 5 & 4 & 16 & 3 \\ 22 & 3 & 12 & 19 \end{bmatrix}$$

- Applying the inverse substitution function $c^{-1}(y) = a^{-1}(y - b) \pmod{31}$, we move on a matrix (say A) of order 4×4 orders as follows:

$$A = \begin{bmatrix} 158 & 144 & 171 & 151 \\ 178 & 10 & 148 & 147 \\ 199 & 143 & 164 & 211 \\ 159 & 149 & 157 & 177 \end{bmatrix}$$

- Add the key matrix K with the matrix A, we get a new matrix (E).

$$E = \begin{bmatrix} 212 & 193 & 227 & 200 \\ 226 & 64 & 196 & 197 \\ 215 & 193 & 217 & 227 \\ 212 & 197 & 213 & 227 \end{bmatrix}$$

- After getting a new matrix, we keep this matrix equivalent to EBCDIC character code, after this we get the original plain text.

Maths Department

V. RESULT AND DISCUSSION

In this paper, we used a substitution algorithm and Xor cipher for encrypting and decrypting messages. The new encryption method is a substitution technique and the XOR operator to encrypt data by replacing bits or characters with predetermined substitutes according to a given key. The procedure conceals the original data in such a way that it is hard for intruders to read without the proper key. Security in this system relies on how intricate the substitution pattern and key are so that it is immune to attacks by brute force and ciphertext analysis. The algorithm employs main operations like right shift, matrix subtraction, substitution, modulo operation, and XOR to create the key using EBCDIC code. These methods complicate decryption even further. Modulo operation contributes significantly towards increased security because its order of growth is larger, thus encryption is more difficult. Brute force attacks also fail to decrypt it because the encryption employs a 4×4 matrix as a key. Without the inverse substitution function and key matrix, decryption of big messages is not feasible. This keeps stored and sent messages secret, particularly for secret information such as military information. Because of these security properties, the likelihood of attacks such as ciphertext attack, chosen plaintext attack, brute force attack, and known plaintext attack is very slim. Hence, this suggested algorithm is much more secure than other encryption algorithms designed by researchers.

VI. CONCLUSION

In conclusion, the substitution algorithm and logical operator XOR stands as an essential tool for simplifying complex problems and enhancing security, with potential for continued growth and adaptation in the future.

REFERENCES

[1]. Forouzan Behrouz A: Cryptography & Network Security, McGraw Hill Education, 2007.
[2]. Garg Satish Kumar: Cryptography Using XOR Cipher, Research Journal of Science and Technology, Vol. 09, Issue 01, 2017.
[3]. Kahate Atul: Cryptography and Network Security, Tata McGraw Hill, New Delhi, 2008.
[4]. Mittal Ayush and Gupta Ravindra Kumar: Encryption and Decryption of a Message Involving Genetic Algorithm, International Journal of Engineering and Advanced Technology (IJEAT), Volume-9 Issue-2, December, 2019.
[5]. Sindhuja K, Pramela Devi S: A Symmetric Key Encryption Technique Using Genetic Algorithm, International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5, Issue 1, 2014.
[6]. Stallings William: Cryptography and Network Security Principles and Practices, Prentice Hall, 2005.