

# A NOVEL TRAPDOOR ENCODING TECHNIQUE FOR OPTIMIZED BLOCK CIPHER ENCRYPTION

Ms Neha Shinde<sup>1</sup>, Dr C S Satsangi<sup>2</sup>

ME Scholar<sup>1</sup>; Associate Professor<sup>2</sup>

Medicaps Institute of Science & Technology, Indore<sup>1,2</sup>

Department of Information Technology<sup>1,2</sup>

[nehacs442@gmail.com](mailto:nehacs442@gmail.com)<sup>1</sup>; [cssatsangi@gmail.com](mailto:cssatsangi@gmail.com)<sup>2</sup>

## 1. ABSTRACT

Security of data is the prime factor of any communication system. Various techniques have been used to preserve data from unauthorized access. Cryptography is one of the techniques. To further improve the security of the data, two level security technique may be used. In this paper we have shown the encryption method of trapdoor encoded technique to secure a data block of length 5. This is a symmetric cryptography technique for optimized block cipher encryption. The importance of this technique lies within its encryption method, it is almost impossible to decrypt the data without permission if the receiver does not have the key.

**Keywords:** trapdoor, prime number, plain text cipher text, encryption

## 2. INTRODUCTION

Cryptography is a technique used to avoid unauthorized access of data. It has two main components; a) Encryption algorithm, and b) Key. Sometime, multiple keys can also be used for encryption. A number of cryptographic algorithms are available in market such as DES, AES, TDES and RSA. The strength of these encryption algorithms depends upon their key strength.

Strong encryption algorithms and optimized key management techniques always help in achieving confidentiality, authentication and integrity of data and reduce the overheads of the system.

Cryptography is basically divided into two categories; a) Symmetric Cryptography, and b) Asymmetric Cryptography. In symmetric cryptography the key used to encrypt the message is the same as the key decrypting the message whereas in asymmetric cryptography different key is used for encryption and decryption. Asymmetric algorithms are relatively slower than symmetric algorithms but provide a good security level.

### 2.1 Trapdoor Function

A trapdoor (one-way function) is one that maps a domain into a range such that every function value has a unique inverse, with the condition that the calculation of the function is easy, whereas the calculation of the inverse is infeasible: generally, *easy* is defined to mean a problem that can be solved in polynomial time as a function of input length. Thus, if the length of the input is  $n$  bits, then the time to compute the function is proportional to  $na$ , where  $a$  is a fixed constant.

Such algorithms are said to belong to the class  $p$ . The term *infeasible* is a much fuzzier concept. In general, we can say a problem is infeasible if the effort to solve it grows faster than polynomial time as a function of input size. For example, if the length of the input is  $n$  bits and the time to compute the function is proportional to  $2^n$ , the problem is considered infeasible. Unfortunately, it is difficult to determine if a particular algorithm exhibits this complexity. Furthermore, traditional notions of computational complexity focus on the worst-case or average-case complexity of an algorithm. These measures are inadequate for cryptography, which requires that it be infeasible to invert a function for virtually all inputs, not for the worst case or even average case.

$X = F^{-1}(Y)$  Infeasible; Without Knowing Certain Additional Information, E.G. Key Etc.

$Y = F(X)$  Easy

## 2.2 Block Cipher Function

A block cipher is a function which maps  $n$ -bit plaintext blocks to  $n$ -bit cipher-text blocks;  $n$  is called the *block length*. It may be viewed as a simple substitution cipher with large character size. The function is parameterized by a  $k$ -bit key  $K$ ,<sup>1</sup> taking values from a subset  $K$  (the *key space*) of the set of all  $k$ -bit vectors  $V_k$ . It is generally assumed that the key is chosen at random. Use of plaintext and cipher text blocks of equal size avoids data expansion.

To allow unique decryption, the encryption function must be one-to-one (i.e., invertible). For  $n$ -bit plaintext and cipher text blocks and a fixed key, the encryption function is a bijection, defining a permutation on  $n$ -bit vectors. Each key

potentially defines a different bijection. The number of keys is  $|K|$ , and the *effective key size* is  $\lg |K|$ ; this equals the key length if all  $k$ -bit vectors are valid keys ( $K = V_k$ ). If keys are equi-probable and each defines a different bijection, the *entropy* of the key space is also  $\lg |K|$ .

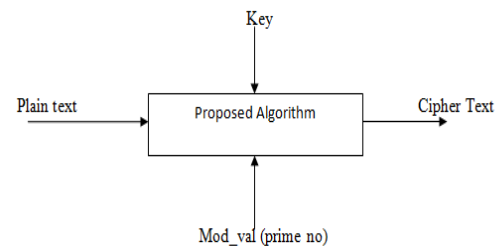


Figure 1: Block Cipher Encryption process

## 3: METHODOLOGY

Enter the message to be encoded in blocks of 5.

Select a prime number whose modular arithmetic will determine the eventual course of encryption and decryption and eventually the course of data transfer. In this case the number has been chosen as 97. The entire character set will produce different residues when used in the algorithm. The block size decides the number of characters to be transferred at a time. Greater the block size greater will be dimensions of the block matrix and harder. It will be to find its inverse by brute force which eventually augments the strength of the algorithm.

Here The Block Size Has Been Chosen As 5. The symbols of the plain text are converted into their corresponding ASCII values using the look-up table. Now select a square invertible matrix 'p' and a diagonal matrix 'e'. The matrix 'e' is a completely random selection which later on decides how immune the algorithm will act against cryptanalysis. The important point to note here is that the matrix should be invertible. Now a secret key is selected which is available only to the transmitter and the intended receiver. Here the key is chosen as '40'.

Now, an iteration of  $e * p$  is calculated for a number of steps equal to the value of the key. After this,

the role of the one way trapdoor function comes into picture as the mod of  $e \cdot p$  and  $\text{mod\_val}$  i.e. The prime number chosen is calculated and is assigned to  $e$ . Then  $e \cdot b^{-1}$  is calculated. This step provides substantial strength to the algorithm since finding the inverse of a matrix is a complex process by brute force. Now, the first stage of code block is generated by again using modular arithmetic. After this, the code expansion and manipulation function is implemented. The vector comprising of 13 symbols is generated, the code is divided by 16(hex modulus) and the remainders collected.

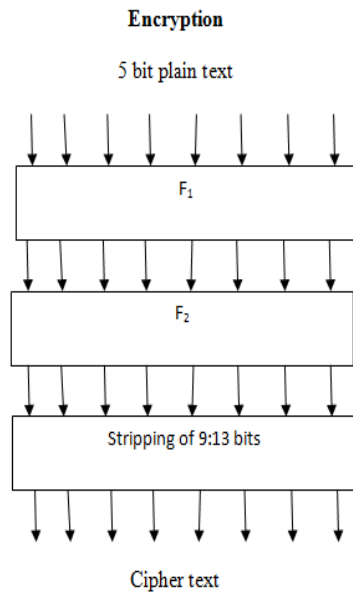


Figure 2: The Function of Proposed Algorithm for Encryption

Then, a quotient polynomial is generated using the prime number. Similarly a remainder polynomial is generated. The quotient generated forms the next dividend polynomial and division is carried out once more and remainder collected. This process is carried on until divisor is larger than the dividend. All the remainders are collected and the array is inverted. This is the manipulated code, using the variables  $j, k$  and  $n$ . This generates a second stage of code termed here as the cipher. The cipher is stored in an array 'b1'. Then the 9<sup>th</sup>

to the 13<sup>th</sup> symbols are dropped to avoid redundancy and for code compression. Finally, the array is converted to binary which gives the 'cipher text'.

Table3.1: Permutation Table used in the Proposed Algorithm

81	33	72	0	14
1	56	91	20	33
20	72	18	39	52
39	91	33	64	71
57	11	49	74	92

### 3.1 Proposed Encryption Algorithm

INPUT: Plaintext  $m_1, m_2, m_3, m_4, m_5$ ;

Prime number value (97 in this case);

( $\text{mod\_val}$ ) which will decide further course of encryption;

Key = 40 (in this case);

OUTPUT = Ciphertext  $\rightarrow 8$  bit;

( $c_1, c_2, \dots, c_8$ );

- 1) Convert the 5 bit plain text into the corresponding ASCII values using look up table and save values in a 'B' matrix;
- 2) Choose a permutation matrix 'P' and a diagonal matrix 'E';
- 3) For the values of the secret key, calculate  $P \cdot E$  and store its value in E;

4) Then implement modular arithmetic on E and mod value & assign the value to E ie  $E \leftarrow \text{mod}(E, \text{Mod\_value})$ ;

5) Calculate initial code using E and inverse of matrix B ie  $\text{code} \leftarrow E * B^{-1}$ ;

Implementation of Trapdoor function

6) Compute modular arithmetic using initial code and mod\_value;

7) Code manipulation function and generation of cipher text;

8) Chose a 13 bit blank array;

9) Calculate the quotient and remainder by dividing it with 16;

10) Generate quotient and remainder polynomials using modular arithmetic;

$$q \leftarrow \text{floor}(((\text{rem} * 97) + \text{code}(i)) / 16);$$

$$r \leftarrow \text{mod}(((\text{rem} * 97) + \text{code}(i), 16));$$

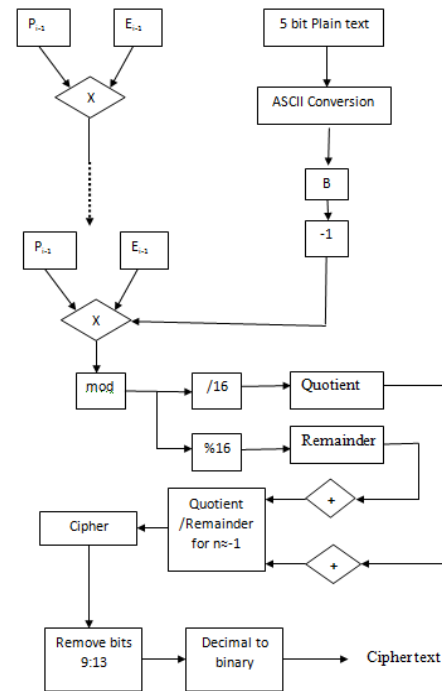
11) Assign remainder to B1 matrix and remainder to code matrix;

12) Continue division until divisor does not reach same order of dividend;

13) Save the cipher in B1 and store the bits in a text file 'cipher.txt';

14) Convert the 13 bit cipher into 8 bit cipher by converting it into a column matrix and stripping off bits 9 to 13 ie  $(9:13, :) = [ ]$ ;

Convert decimal values to binary values to get required cipher text.

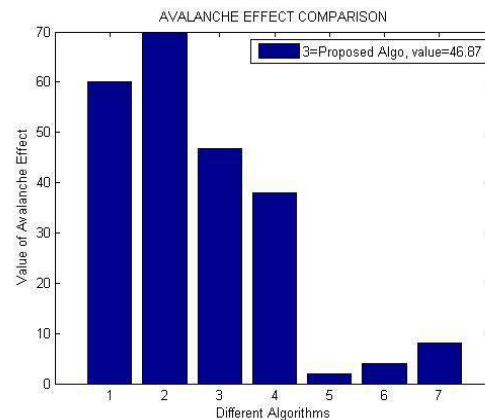


Flow chart for proposed encryption algorithm

Figure 3: Flow chart for proposed encryption algorithm

### 4 RESULTS AND DISCUSSION

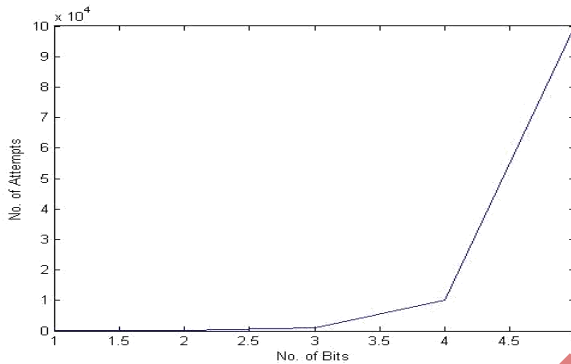
**4.1 Avalanche Effect:** It is the number of flipped bits in the cipher text divided by the total number of bits in the cipher text for one bit change in the plain text.



Graph 1 Avalanche effect comparison

### 4.2 Effect of Key Size variation on proposed algorithm:

The table shown below shows that if the length of the key bit is 1 then we will recover the key surely after the 10 attempts. If the length of the key is increased to 5 then 100000 attempts are required which takes a lot of time recover the key. This means that security keeps on increasing as the key size increases but complexity of the system is also increased. So there is a need to maintain tradeoff between key length and complexity of the system.



Graph 2: Bit Length Effect of Key on Security

The curve shows that if the number of bits of key length is increased than the security of system increases. The system is safe for 10000 attempts in case of 4bit, during this time information is reached at the destination and at the second time transmission the key is changed. If the time is less and not provide full satisfaction of safety, use the 5 bit key length. Depending upon the requirements key length is increased.

The proposed algorithm shows a reasonable amount of avalanche effect owing to the fact that there are several rounds of manipulation of the data block comprising of calculation of arandom invertible matrix, its product with a diagonal matrix, modular arithmetic and generation of expansion code, remainder and quotient polynomials and finally flipping of the array.

Plaintext1= Nehah

Ciphertext1=

10010001011100111000111111001010

Plaintext2= Nehai

Ciphertext2=

01100100010100111101100001101011

### 4.3 The Bit Independence Criteria (Bic):

This is another very important parameter which ultimately decides the difficulty in differential and linear cryptanalysis. It states that two bits in the cipher text must change independently for a change in the plain text. In other words, the changes in the cipher text must not yield any perceptible pattern.

Bit Independence Criteria (Bic)

M1=Nehah

C1=

1001(9)0001011100111000111111001010

M2=Nehai

C2=

0110(6)0100010100111101100001101011

M3=Nehaj

C3=

0100(4)1111110000001001011001011100

M4=Nehak

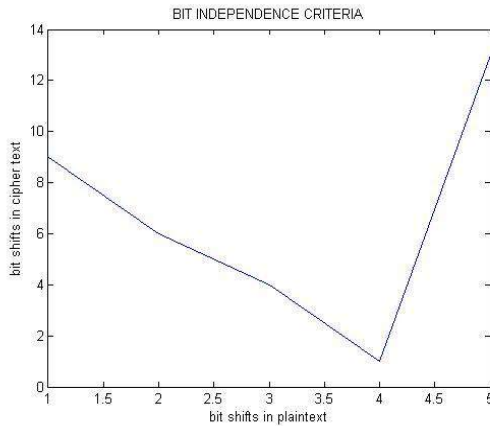
C4=

0001(1)0010101100001110111111101100

M5=Nehal

C5=

1110(13)1110000001000101100010001101

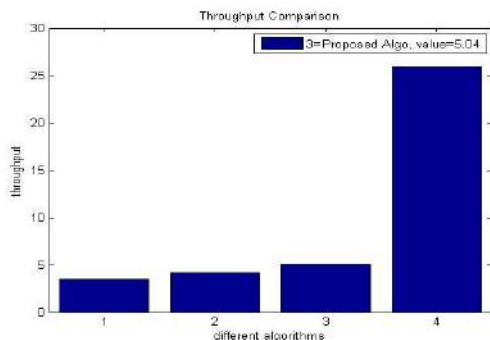


Graph 3 Bit Independence criteria

#### 4.4 Throughput:

Its an important parameter which decides the speed and the efficiency of the algorithm. It plays a crucial role in deciding whether the algorithm is designed well enough to utilize optimal values of space and time if implemented practically. High values of throughput imply more avenues of practical implementation of the algorithm.

Throughput =(No. Of Bytes In Plaintext)/(Total Execution Time) .It Can Be Seen From The Graph Below That The Proposed Algorithm Gives Better Performance Than Both AES And DES Which Have Enjoyed Widespread Popularity.



Graph 4 Throughput Comparison

#### 5: CONCLUSION

It can be said that the proposed algorithm is a novel method for the implementation of an advanced form of encryption standard which must prove to be resistant to linear as well as differential cryptanalysis due to the use of self-invertible matrices as well as multiple stages of one way functions. This increases the randomness, which is exhibited in the parameters like Avalanche Effect and BIC. Still the, simplicity of implementation and high value of throughput indicate towards ease in practical utility and hardware implementation. This makes it a lightweight algorithm. Modular arithmetic involving prime numbers and self-invertible matrices makes it exceedingly resistant to cryptanalysis.

#### 6 REFEREENCES

- [1] A. Khalique, K. Singh and S. Sood, “ A Password-Authenticated Key Agreement Scheme Based on ECC Using Smart Cards”, International Journal of Computer Applications, Vol. 2, No.3, pp. 26-30, 2010.
- [2] C. H. Kim, “Improved Differential Fault Analysis on AES Key Schedule”, IEEE Transactions on Information Forensics and Security, Vol. 7, No. 1, pp. 41-50, 2012.
- [3] H. Yang, E. Osterweil, D. Massey, S. Lu, and L. Zhang, “Deploying Cryptography in Internet-Scale Systems: A Case Study on DNSSEC”, IEEE Transactions on Dependable and Secure Computing, Vol. 8, No. 5, pp. 656-669, 2011.
- [4] K. Bhatele, A. Sinhal and M. Pathak, “A Novel Approach to the Design of a New Hybrid

Security Protocol Architecture”, IEEE International Conference on Advanced Communication Control and Computing Technologies, pp.429-433, 2012.

[5] L. J. G. Villalba, J. G. Matesanz, D. R. Canas and A. L. S. Orozco, “Secure Extension to the Optimized Link State Routing Protocol”, IET Information Security, Vol. 5, No. 3, pp. 163-169, 2011. [11] M. E. Hellman, “An Extension of the Shannon Theory Approach to Cryptography”, IEEE Transactions on Information Theory, Vol. 23, No. 3, pp. 289-294, 1977.

[6] M. Alioto, M. Poli and S. Rocchi, “Differential Power Analysis Attacks to Precharged Buses: A General Analysis for Symmetric-Key Cryptographic Algorithms”, IEEE Transactions on Dependable and Secure Computing, Vol. 7, No. 3, pp. 226-239, 2010.

[7] S. F. Mare, M. Vladutiu and L. Prodan, “Secret data communication system using Steganography, AES and RSA”, IEEE 17th International Symposium for Design and Technology in Electronic Packaging, pp. 339-344, 2011.

[8] M. Y. Wang and C.W. Wu, “A Mesh-Structured Scalable IPsec Processor”, IEEE

[9] Transactions on Very Large Scale Integration Systems, Vol. 18, No. 5, pp. 725-731, 2010.