# A Research paper: An ASCII value based data encryption algorithm and its comparison with other symmetric data encryption algorithms

Akanksha Mathur
Computer Sc. & Engg.
JIET Group of Institutions
Jodhpur, India
akanmamthur@gmail.com

***Abstract*— Encryption is the process of transforming plaintext into the ciphertext where plaintext is the input to the encryption process and ciphertext is the output of the encryption process. Decryption is the process of transforming ciphertext into the plaintext where ciphertext is the input to the decryption process and plaintext is the output of the decryption process. There are various encryption algorithms exist classified as symmetric and asymmetric encryption algorithms. Here, I present an algorithm for data encryption and decryption which is based on ASCII values of characters in the plaintext. This algorithm is used to encrypt data by using ASCII values of the data to be encrypted. The secret used will be modifying o another string and that string is used as a key to encrypt or decrypt the data. So, it can be said that it is a kind of symmetric encryption algorithm because it uses same key for encryption and decryption but by slightly modifying it. This algorithm operates when the length of input and the length of key are same.**

**Keywords- *Encryption, Decryption, ASCII, symmetric encryption, plaintext, ciphertext***

## I. INTRODUCTION

Cryptography is the art and science of study of designing or generating the secret message i.e. code or ciphers of the original message for the secure communication between sender and the receiver. The main goals of cryptography are (1) Authentication, (2) Privacy, (3) Integrity, (4) Non-repudiation [3] and (5) Access Control.

Encryption is basically a process or algorithm to make information hidden or secret. It is considered as the subset of cryptography. It is the actual process of applying cryptography. It is the process to transform or converting the data into some another form that appears to be random, meaningless and unintelligible. It can also be said that encryption is the process of transforming plaintext into the ciphertext where plaintext is the input to the encryption process and ciphertext is the output of the encryption process.

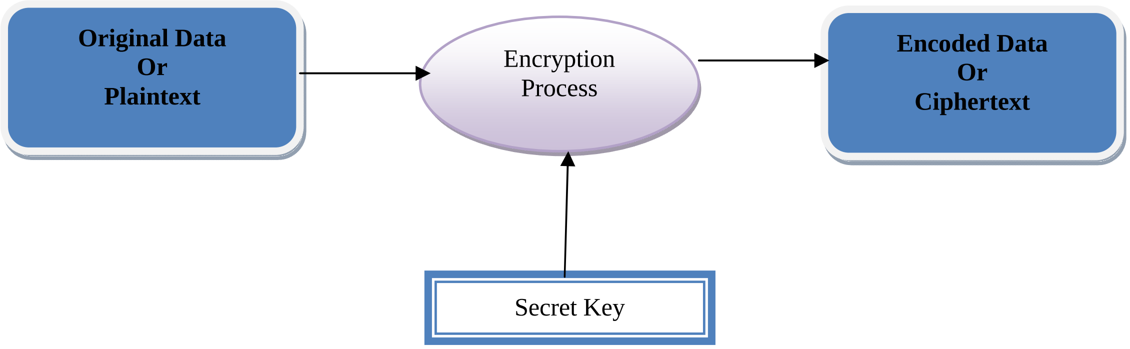


Fig.1 Encryption Process

Conversely, Decryption is the process to transform or converting the encoded data into some meaningful form. It can also be said that decryption is the process of transforming ciphertext into the plaintext where ciphertext is the input to the decryption process and plaintext is the output of the decryption process.

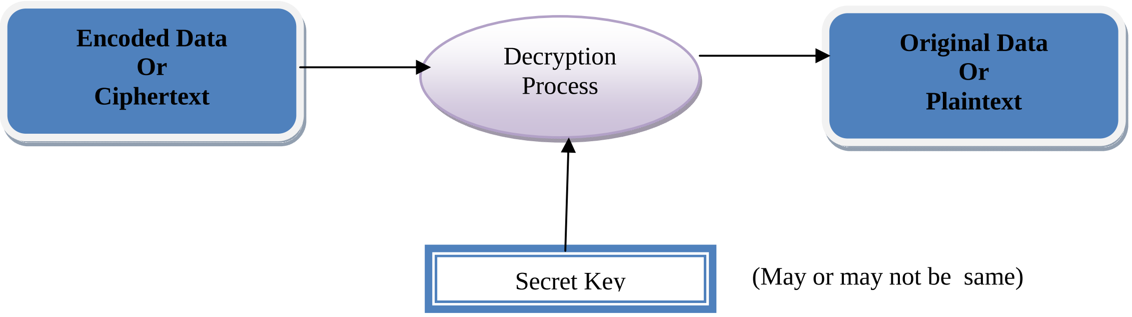


Fig.2 Decryption Process

A cryptographic algorithm is a mathematical functions and unchanging set of steps to perform encryption and decryption of the original data. The main objective of every cryptographic algorithm is to make it as difficult as possible to decrypt the generated ciphertext without using the key. If a really good cryptographic algorithm is used, then there is no technique significantly better than methodically trying every possible combination of key. Encryption Algorithms are categorized into 2 categories as follows:-

(1) **Symmetric Key Encryption -** In symmetric encryption algorithm, only one key is used for both encryption and decryption process. The key is transmitted to both the sender and receiver before. The key is transmitted to both the sender and receiver before the process of encryption and decryption. So, the secret key plays an important role and its strength depends on the length of key (in bits). Symmetric key encryption algorithms are- RC2, DES, 3DES, RC5, Blowfish, and AES et al.

(2) **Asymmetric Key Encryption-** In symmetric key encryption algorithm, it is necessary to distribute the key before the encryption and decryption because the same key is used for both purposes. This problem of distribution of key in symmetric algorithms is solved by asymmetric key encryption algorithm. It uses two types of keys, Private keys and Public Keys. Public Key is used to encrypt the original data or plaintext and generate a ciphertext. This ciphertext is decoded by the receiver as and when it receives by using its own Private Key. [1] Private Key is also known as secret key because it is unknown to all or known only to the person who is intender to receive it or can say authorized person. But public keys can be stored in public databases for anyone to see. [2] Asymmetric key encryption algorithms are RSA, Digital Signatures et al.

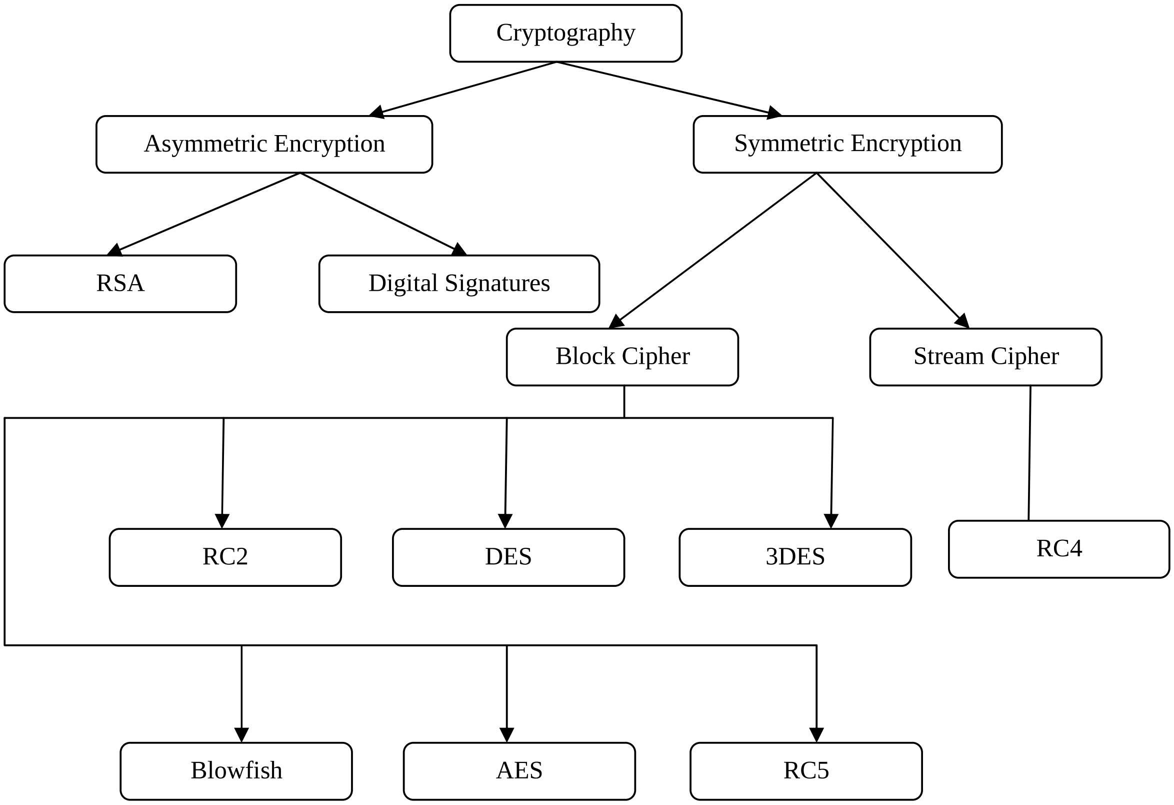


Fig. 3 Overview of Cryptographic Algorithm

## II. INTRODUCTION TO ASCII VALUE BASED DATA ENCRYPTION ALGORITHM

### *A. Introduction*

This algorithm is used to encrypt data by using ASCII values of the data to be encrypted. The secret key used will be modifying to another string and that modified string is used to encrypt or decrypt the data. So, it can be said that it is a kind of symmetric encryption algorithm because it uses same key for both encryption and decryption but by slightly modifying it. This algorithm operates only when the length of input and length of key are same.

### *B. Algorithm to perform Encryption*

Following steps are performed to encrypt the plain text:-

1) Get the ASCII values of each character of input string i.e. plain text and store it in an ASCII content array.

Eg.:-

| **Input** | n | e | h | a |
|---|---|---|---|---|
| **ASCII Content** | 113 | 101 | 104 | 97 |

2) Find out the minimum value *mincontent* from the asciicontent array. Eg.:-Mincontent=97

3) Now perform the modulus operation on each asciicontent value as follows i.e. ASCIIContent[i] %min and save the resultants in modcontent array where the value of I ranges upto the length of input.

| **Input** | n | e | h | a |
|---|---|---|---|---|
| **ASCIIContent** | 110 | 101 | 104 | 97 |
| **modcontent** | 13 | 4 | 7 | 0 |

If the value of mod content is greater than 16, then again perform modcontent %16, and record the places where changes occur or record the positions where the value of mod content is greater than 16.

4) Now perform the modulus operation on each ASCII content value as follows i.e. ASCII Key[i] %min and save the resultants in modcontent array where the value of I ranges upto the length of key.

| **Input** | n | e | h | a |
|---|---|---|---|---|
| **ASCIIContent** | 110 | 101 | 104 | 97 |
| **modcontent** | 13 | 4 | 7 | 0 |
| **Key** | a | b | c | d |
| **ASCIIkey** | 97 | 98 | 99 | 100 |
| **modkey** | 0 | 1 | 2 | 3 |

5) Take the binary values of each value of modkey.

| **Input** | n | e | h | a |
|---|---|---|---|---|
| **ASCIIContent** | 110 | 101 | 104 | 97 |
| **modcontent** | 13 | 4 | 7 | 0 |
| **Key** | a | b | c | d |
| **ASCIIkey** | 97 | 98 | 99 | 100 |
| **modkey** | 0 | 1 | 2 | 3 |
| **binary** | 0000 | 0001 | 0010 | 0011 |

6) Perform the right circular shifts of binary values (n time where n is the length of input).

| | 0000 | 0001 | 0010 | 0011 |
|---|---|---|---|---|
| Right circular shift 1 | 1000 | 0000 | 1001 | 0001 |
| Right circular shift 2 | 1100 | 0000 | 0100 | 1000 |
| Right circular shift 3 | 0110 | 0000 | 0010 | 0100 |
| Right circular shift 4 | 0011 | 0000 | 0001 | 0010 |

Now after circular shifting, key will become- 3 0 1 2

| | | | | |
|---|---|---|---|---|
| **Input** | n | e | h | a |
| **ASCIIContent** | 110 | 101 | 104 | 97 |
| **modcontent** | 13 | 4 | 7 | 0 |
| **Key** | a | b | c | d |
| **ASCIIkey** | 97 | 98 | 99 | 100 |
| **modkey** | 0 | 1 | 2 | 3 |
| **binary** | 0000 | 0001 | 0010 | 0011 |
| **Encrypt key after shifting** | 3 | 0 | 1 | 2 |

7) Now add min value to each ASCII value of each character of encrypt key after shifting. So, final encrypt key is-

| | | | | |
|---|---|---|---|---|
| **Input** | n | e | h | a |
| **ASCIIContent** | 110 | 101 | 104 | 97 |
| **modcontent** | 13 | 4 | 7 | 0 |
| **Key** | a | b | c | d |
| **ASCIIkey** | 97 | 98 | 99 | 100 |
| **modkey** | 0 | 1 | 2 | 3 |
| **binary** | 0000 | 0001 | 0010 | 0011 |
| **Encrypt key after shifting** | 3 | 0 | 1 | 2 |
| **add min value to encrypt key** | 100 | 97 | 98 | 99 |
| **Final Encrypt key** | d | a | b | c |

8) Now to encrypt the original data (input) or plaintext to generate ciphertext, add each mod content value to the ascii values of final encrypt key.

| | | | | |
|---|---|---|---|---|
| **Input** | n | e | h | a |
| **ASCIIContent** | 110 | 101 | 104 | 97 |
| **modcontent** | 13 | 4 | 7 | 0 |
| **Key** | **a** | **b** | **c** | **d** |
| **ASCIIContent** | 97 | 98 | 99 | 100 |
| **modkey** | 0 | 1 | 2 | 3 |
| **binary** | 0 | 1 | 10 | 11 |
| **Encrypt key after shifting** | 3 | 0 | 1 | 2 |
| **add min value to encrypt key** | 100 | 97 | 98 | 99 |
| **Final Encrypt key** | d | a | b | c |
| **ciphertext ascii values** | 113 | 101 | 105 | 99 |

So, final cipher text is-

| | | | | |
|---|---|---|---|---|
| **Input** | n | e | h | a |
| **ASCIIContent** | 110 | 101 | 104 | 97 |
| **modcontent** | 13 | 4 | 7 | 0 |
| **Key** | **a** | **b** | **c** | **d** |
| **ASCIIContent** | 97 | 98 | 99 | 100 |
| **modkey** | 0 | 1 | 2 | 3 |

| | | | | |
|---|---|---|---|---|
| **binary** | 0 | 1 | 10 | 11 |
| **Encrypt key after shifting** | 3 | 0 | 1 | 2 |
| **add min value to encrypt key** | 100 | 97 | 98 | 99 |
| **Final Encrypt key** | d | a | b | c |
| **ciphertext ascii values** | 113 | 101 | 105 | 99 |
| **Ciphertext** | q | e | i | c |

### *C. Algorithm to perform Decryption*

Following steps are performed to decrypt the cipher text:-

1) Take ciphertext and find out the minimum from ASCII values of each character of cipher text.

| | | | | |
|---|---|---|---|---|
| **Ciphertext** | q | e | i | c |
| **Ascii Cipher** | 113 | 101 | 105 | 99 |

Mincipher is:- 99

2) Now Perform the subtraction of ascii values of final encrypt key from asciicipher

| | | | | |
|---|---|---|---|---|
| **Cipher** | q | e | i | c |
| **ASCIICipher** | 113 | 101 | 105 | 99 |
| **asciifinalencrypt** | 100 | 97 | 98 | 99 |
| **difference** | 13 | 4 | 7 | 0 |

Add 16 to the stored positions where the modcontent value is greater than 16.

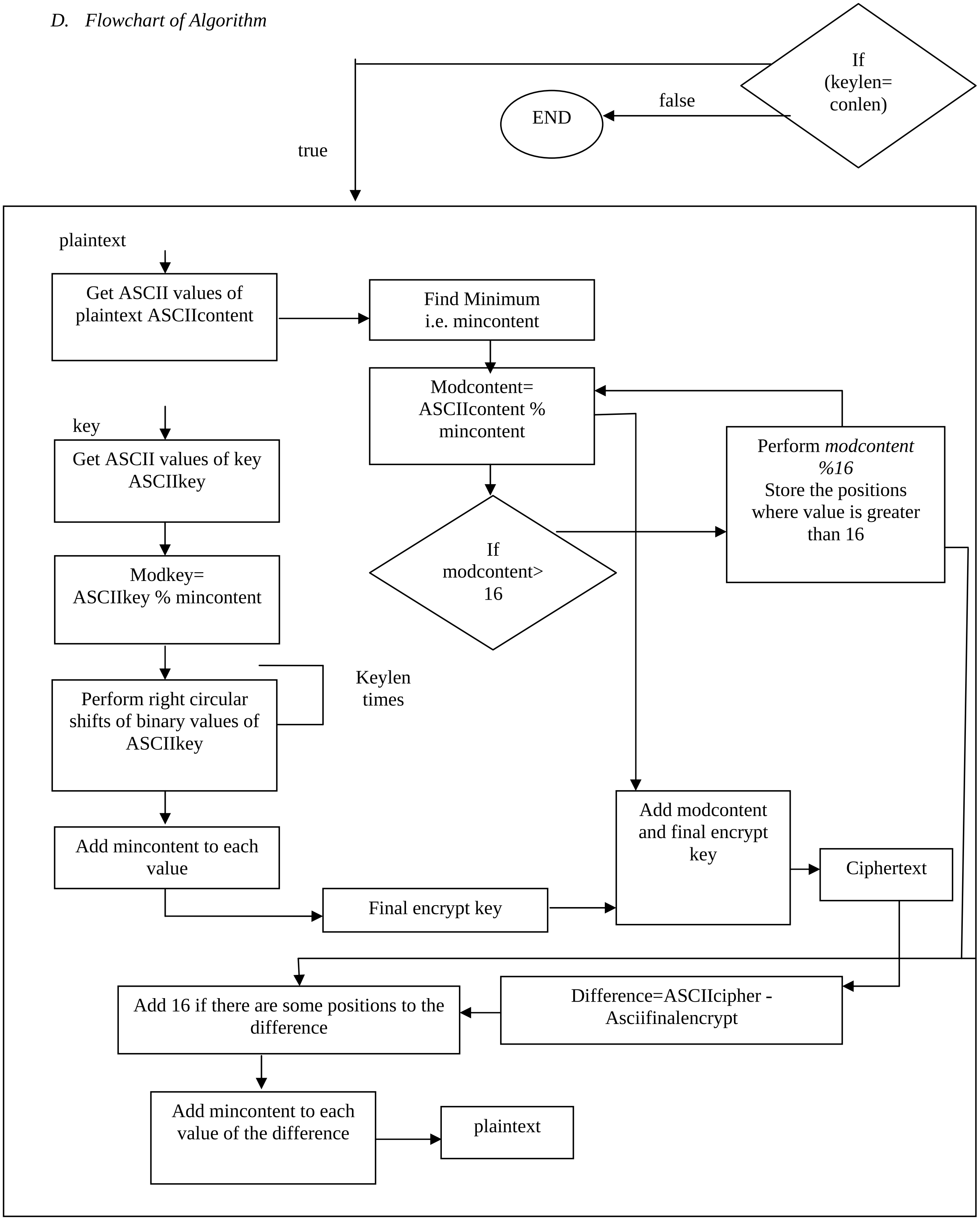
3) Add mincontent to each value of difference to generate plaintext.

| | | | | |
|---|---|---|---|---|
| **Cipher** | q | e | i | c |
| **ASCIICipher** | 113 | 101 | 105 | 99 |
| **asciifinalencrypt** | 100 | 97 | 98 | 99 |
| **difference** | 13 | 4 | 7 | 0 |
| **asciiplain** | 110 | 101 | 104 | 97 |

So, plaintext is-

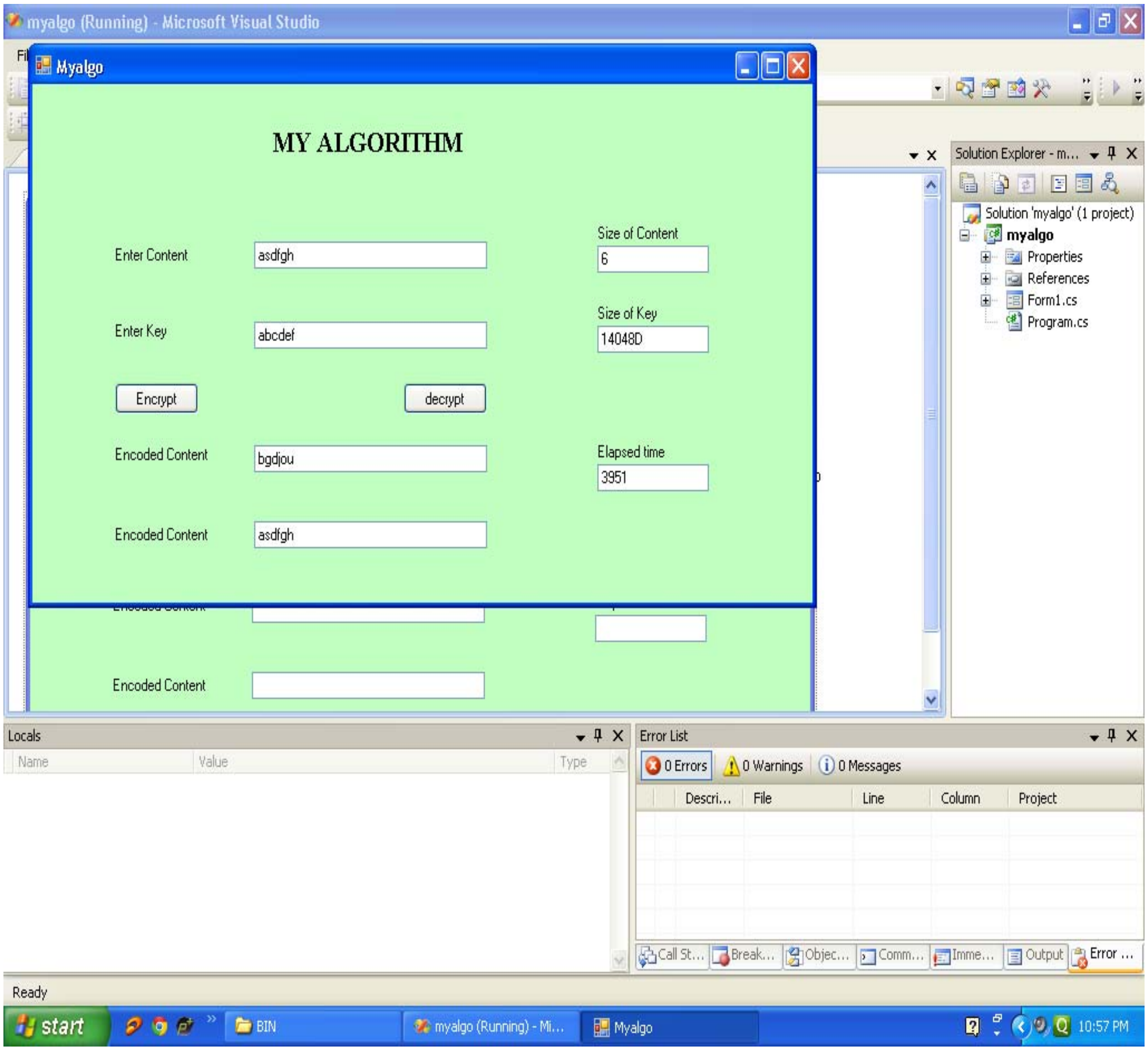
| | | | | |
|---|---|---|---|---|
| **Cipher** | q | e | i | c |
| **ASCIICipher** | 113 | 101 | 105 | 99 |
| **asciifinalencrypt** | 100 | 97 | 98 | 99 |
| **difference** | 13 | 4 | 7 | 0 |
| **asciiplain** | 110 | 101 | 104 | 97 |
| **plaintext** | n | e | h | a |

*D. Flowchart of Algorithm*


If (keylen= conlen)
false
END
true
plaintext
Get ASCII values of plaintext ASCIIcontent
Find Minimum i.e. mincontent
Modcontent= ASCIIcontent % mincontent
key
Get ASCII values of key ASCIIkey
Perform *modcontent %16* Store the positions where value is greater than 16
If modcontent> 16
Modkey= ASCIIkey % mincontent
Perform right circular shifts of binary values of ASCIIkey
Keylen times
Add modcontent and final encrypt key
Add mincontent to each value
Ciphertext
Final encrypt key
Add 16 if there are some positions to the difference
Difference=ASCIIcipher - Asciifinalencrypt
Add mincontent to each value of the difference
plaintext

## III. IMPLEMENTATION & CALCULATING EXECUTION TIME

The below figure represent the implementation of proposed algorithm in C#.net. This implementation takes input as plain text and secret key from the user and encrypts & decrypts the data by following algorithm steps. It also calculates the execution time of the algorithm in microseconds.



MY ALGORITHM

Enter Content: asdfgh

Enter Key: abcdef

Size of Content: 6

Size of Key: 14048D

Encoded Content: bgdjou

Encoded Content: asdfgh

Elapsed time: 3951

The following table represents the data analysis of proposed algorithm baesd on the different size of plaintext. First table represents the execution time on the basis of the size plaintext and second table represents the ciphertext of different plain text. The graph indicates that as the length of input increases, the execution time also increases.

Table 1 PROPOSED ALGO

| Size of plain text | Execution time |
|---|---|
| 2 | 322 |
| 4 | 3679 |
| 6 | 3861 |
| 8 | 4748 |
| 10 | 5543 |

| Plaintext | Key | Ciphertext |
|---|---|---|
| as | sd | mk |
| neha | abcd | qeic |
| asdfgh | abcdef | bgdjou |
| akanksha | abcdefgh | graomflf |
| qwertyuiop | abcdefghij | ~hiv\|veisx |

## IV. LIMITATIONS

The proposed algorithm has the following limitations:-

1) More Execution time
2) Key Length and length of plain text must be same

## V. FUTURE SCOPE

In the future wok related to proposed algorithm, the limitations of propsed algorithm are overcomed by encrypting and decrypting data with may or may not be same key length size in comparison with input size.

## REFERENCES

[1] Gurjeevan Singh, Ashwani Kumar Singla, K.S. Sandha, "Throughput Analysis of Various Encryption Algorithms", International Journal of Computer Science and Technology, Vol. 2, Issue 3, Septemver 2011.

[2] Diaa Salama Abd Elminaam, Hatem Mohamed Abdual Kader, and Mohiy Mohamed Hadhoud, "Evaluating the Performance of Symmetric Encryption Algorithms", International Journal of Network Security, Vol.10, No.3, PP.216–222, May 2010.

[3] Gary C. Kessler "An Overview of Cryptography" , May 1998

## AUTHORS PROFILE

Akanksha Mathur
Assistant Professor
JIET Group of Institutions,
Jodhpur