# Encryption of text characters using ASCII values

Akanksha Mathur
Gujarat University
Ahmedabad, India
akanmamthur@gmail.com

Arshi Riyaz
Department of Computer Science and Engineering
JIET Universe, Jodhpur, India
arshiriyaz@gmail.com

Jyoti Vyas
Department of Computer Science and Engineering
JIET Universe, Jodhpur, India
jyotivyas@jietjodhpur.com

***Abstract*—This paper is demonstrating the encryption and decryption of text characters using their ASCII values. This is a kind of symmetric Encryption algorithm in which same key is used for both encryption and decryption purpose.**

***Keywords*—ASCII, encryption, Decryption, ciphertext, plaintext, cryptographic algorithm**

## I. Introduction TO CRYPTOGRAPHY

A **cryptographic algorithm** is a mathematical functions and unchanging set of steps to perform encryption and decryption of the original data. These algorithms work in combination with a **secret key** which can be a combination of alphabets, numbers, words or phrases. For the purpose of encryption, the algorithm combines the original data or the text to be encoded (plaintext which is input to the encryption process) with the secret key supplied for the encryption. This combination will yield a ciphertext (which is our desired code or we can say output). Similarly, for the purpose of decryption, the algorithm combines the encrypted data or ciphertext with may or may not be the same secret key and this combination will yield again the same plaintext. If there is any modification takes place in any of the secret key or the plaintext, the algorithm will yield a different result than before. 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.

## II. ASCII BASED 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 modified to another string and that string is used as a key to encrypt or decrypt the data.[3] So, it can be said that it is a kind of symmetric encryption algorithm. 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 process of encryption and decryption. So, the secret key plays an important role and its strength depends on the length of key (in bits). The longer the length of key is, it is harder to break it and shorter the length of key is it is even easier to break it. [1] Thus it violates the security purpose of encryption. Similarly .it uses same key for encryption and decryption but by slightly modifying it.

The main limitation of this algorithm is that it will operates when the length of input and the length of key are same. That is, if the length of input is 3 then the length of key must be 3 neither less or nor more than 3.

### *B. Algorithm for encryprion process*

1) Start
2) Input the string (can include numbers, alphabets and special symbols) from the user. This string is known as the *plain text* to be encrypted.
3) Get the ASCII values of each character of *plain text* and store them in an array *asciicontent.*
4) Find out the minimum value *min* from the array *asciicontent*. This *min* value is used further in the algorithm.
5) For I = 1 to n *where n is the length of the input of the plain text*
   *modcontent*[I] = *asciicontent[I] % min*

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 in *record array* where the value of mod content is greater than 16.

*6)* Input the string (can include numbers, alphabets and special symbols) from the user. This string is the *key* which is used to encrypt the *plain text.*\
*7)* Get the ASCII values of each character of *key* and store them in an array *asciikey*.
*8)* For I = 1 to n *where n is the length of the input of the key*
   *modkey[I] = asciikey[I] % min*
*9)* Take the binary values of each value of modkey.
*10)* Perform the right circular shifts of binary values n times (where n is the length of input i.e. *plain text*) and save them in *binary array*.

*11)* Add min value to each ASCII value of each character of encrypt key after shifting.
*Encryptkey[I]=ASCII (Binary[I]) + min*
*Encryptkey* is the final key which is used to encrypt the plain text.

12) To encrypt the original data (input) or plaintext to generate ciphertext, add each mod content value to the ASCII values of final encrypt key.

*13) Ciphertext[I]=ASCII(Excryptley[I])+modcontent[I*
Convert the ASCII values into their corresponding characters to get the cipher text.

*C. Algorithm for decryption purpose*

1) Start
2) Get the ASCII values of each character of cipher text in *asciicipher*.
3) Find out the *minimum* from ASCII values of each character of cipher text.
4) Subtract ASCII values of final encrypt key from asciicipher
*Difference[I]=asciicipher[I]-ASCII(Encryptkey[I])*
Add 16 to the stored positions from *record array* where the modcontent value is greater than 16.
5) Add *minimum* to each value of difference to generate plaintext.

## III. ILLUSTRATION OF ALGORITHMS

Here, representing some of the examples of encryption and decryption process of varying length of input (or key) say 2, 3, 4, 5.

*A. Example 1: Input Length:- 2*

Let Plain text is: - am Key is: - ab

*TABLE 1.* EXAMPLE 1

| **ENCRYPTION** | | |
|---|---|---|
| ***Input (Plain Text)*** | **a** | **m** |
| *asciicontent* | 97 | 109 |
| min=97 | | |
| *modcontent* | 0 | 12 |
| *Key* | a | b |
| *asciikey* | 97 | 98 |
| *modkey* | 0 | 1 |
| *binary* | 0000 | 0001 |
| Right Circular Shifts (2 times) | | |
| Shift 1 | 1000 | 0000 |
| Shift 2 | 0100 | 0000 |
| *Encryptkey* | 4 | 0 |
| *Encryptkey (After adding min)* | 101 | 97 |
| *Encryptkey* | e | a |
| *ASCII(Excryptley)+modcontent* | 101 | 109 |
| ***Ciphertext*** | **e** | **m** |
| **DECRYPTION** | | |
| *Cipher* | e | m |
| *ASCIICipher* | 101 | 109 |
| *minimum=101* | | |
| *asciifinalencryptkey* | 101 | 97 |
| *difference* | 0 | 12 |
| *asciiplain* | 97 | 109 |
| **plaintext`** | **a** | **m** |

Execution time: 320ms.

*B. Example 2: Input Length: - 3*

Let Plain text=bcf Key=cbc

*TABLE 2.* EXAMPLE 2.

| **ENCRYPTION** | | | |
|---|---|---|---|
| *Input (Plain Text)* | b | c | f |
| *asciicontent* | 98 | 99 | 102 |
| min=98 | | | |
| *modcontent* | 0 | 1 | 4 |
| *Key* | c | b | c |
| *asciikey* | 99 | 98 | 99 |
| *modkey* | 1 | 0 | 1 |
| *binary* | 0001 | 0000 | 0001 |
| Right Circular Shifts (3 times) | | | |
| Shift 1 | 1000 | 1000 | 0000 |
| Shift 2 | 0100 | 0100 | 0000 |
| Shift 3 | 0010 | 0010 | 0000 |
| *Encryptkey* | 2 | 2 | 0 |
| *Encryptkey (After adding min)* | 100 | 100 | 98 |
| *Encryptkey* | d | d | b |
| *ASCII(Excryptley)+modcontent* | 100 | 101 | 102 |
| *Ciphertext* | d | e | f |
| **DECRYPTION** | | | |

| | | | |
|---|---|---|---|
| *Cipher* | d | e | f |
| *ASCIICipher* | 100 | 101 | 102 |
| *minimum=100* | | | |
| *asciifinalencryptkey* | 100 | 100 | 98 |
| *difference* | 0 | 1 | 4 |
| *asciiplain* | 98 | 99 | 102 |
| plaintext` | b | c | f |

| | | | | |
|---|---|---|---|---|
| *Difference* | 13 | 4 | 7 | 0 |
| *Asciiplain* | 110 | 101 | 104 | 97 |
| plaintext` | n | e | h | a |

Estimated Time: 3679 ms.

*D. Example 4:Inuput Length: -5*

Let plaintext= pacgl Key=abcde

Execution Time: 2098ms.

*C. Example 3: Input Length: - 4*

Let Plain Text= neha Key= abcd

*TABLE 3.* EXAMPLE 3

| **ENCRYPTION** | | | | |
|---|---|---|---|---|
| *Input (Plain Text)* | n | e | h | a |
| *asciicontent* | 110 | 101 | 104 | 97 |
| min=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 |
| Right Circular Shifts (4 times) | | | | |
| Shift 1 | 1000 | 0000 | 1001 | 0001 |
| Shift 2 | 1100 | 0000 | 0100 | 1000 |
| Shift 3 | 0110 | 0000 | 0010 | 0100 |
| Shift 4 | 0011 | 0000 | 0001 | 0010 |
| *Encryptkey* | 3 | 0 | 1 | 2 |
| *Encryptkey (After adding min)* | 100 | 97 | 98 | 99 |
| *Encryptkey* | d | a | b | c |
| *ASCII(Excryptley)+ modcontent* | 113 | 101 | 105 | 99 |
| *Ciphertext* | q | e | i | c |
| **DECRYPTION** | | | | |
| *Cipher* | q | e | i | c |
| *ASCIICipher* | 113 | 101 | 105 | 99 |
| *minimum=99* | | | | |
| *Asciifinalencryptkey* | 100 | 97 | 98 | 99 |

*TABLE 4.* EXAMPLE 4

| **ENCRYPTION** | | | | | |
|---|---|---|---|---|---|
| *Input (Plain Text)* | p | a | c | g | l |
| *asciicontent* | 112 | 97 | 99 | 103 | 108 |
| min=97 | | | | | |
| *modcontent* | 15 | 0 | 2 | 6 | 11 |
| *Key* | a | b | c | d | e |
| *asciikey* | 97 | 98 | 99 | 100 | 101 |
| *modkey* | 0 | 1 | 2 | 3 | 4 |
| *binary* | 0000 | 0001 | 0010 | 0011 | 0100 |
| Right Circular Shifts (5 times) | | | | | |
| Shift 1 | 0000 | 0000 | 1001 | 0001 | 1010 |
| Shift 2 | 0000 | 0000 | 0100 | 1000 | 1101 |
| Shift 3 | 1000 | 0000 | 0010 | 0100 | 0110 |
| Shift 4 | 0100 | 0000 | 0001 | 0010 | 0011 |
| Shift 5 | 1010 | 0000 | 0000 | 1001 | 0001 |
| *Encryptkey* | 10 | 0 | 0 | 9 | 1 |
| *Encryptkey (After adding min)* | 107 | 97 | 97 | 106 | 98 |
| *Encryptkey* | k | a | a | j | b |
| *ASCII(Excryptley)+' modcontent* | 122 | 97 | 99 | 112 | 109 |
| *Ciphertext* | z | a | c | p | m |
| **DECRYPTION** | | | | | |
| *Cipher* | z | a | c | p | m |
| *ASCIICipher* | 122 | 97 | 99 | 112 | 109 |
| *minimum=97* | | | | | |
| *Asciifinal encryptkey* | 107 | 97 | 97 | 106 | 98 |
| *difference* | 15 | 0 | 2 | 6 | 11 |
| *asciiplain* | 112 | 97 | 99 | 103 | 108 |
| plaintext` | p | a | c | g | l |

Execution Time:: 3780ms.

## IV. Limitations

The proposed algorithm has the following limitations:-

1) More Execution time
2) Key Length and length of plain text must be same.[3]
3) If it is applied on any file then the length of key is equal to the length of file which is not considered as good

## V. Future Scope

In the future wok related to proposed algorithm, the limitations of proposed algorithm are overcome by

1) Encrypting and decrypting data with may or may not be same key length size in comparison with input size.
2) Appling on files of different length
3) Applied on images

## VI. 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.