CIPHER & DECIPHER

Project Manual

Language: 8086 Assembly

Software: emu8086

Members:

1. Md. Ratul Mushfique
2. Mohd. Shadman Ahmed Razeen
3. Sadman Safiur Rahman

Algorithm-1: Updated Vigenere Cipher

Cipher:

User provides an encrypted message

User provides two secret letter

Every letter of the encrypted message and the two secret letters will be converted to its numerical values.

Then the odd places numerical value of the encrypted message will be added to the 1st letter and the even places of the encrypted message will be added to the 2nd letter.

Then each letter will be converted to the encrypted character.

For example:-

Secret Letters:- AB

Message:- ‘this’

Calculation:

Even position= t,i

Odd position= h,s

>t+A-115 =>116+65-115=>B [66]

>h+B-115 =>104+66-115=>7 [55]

>i+A-115 =>105+65-115=>7 [55]

>s+B-115 =>115+66-115=>B [66]

[here, 115 is getting subtracted to get a valid character which is present in the keyboard]

Decipher:

Two ways to do it:

1) By choosing to decipher in insertion of text
2) By deciphering the text that is already ciphered in the emulator

[The encrypted text is B77B]

>B-A +115 => 66-65+115 => t [116]

>7-B +115 => 55-66+115 => h [104]

>7-A +115 => 55-65+115 => i [105]

>B-B +115 => 66-66+115 => s [115]

Algorithm-2: RRS(Ratul-Razeen-Sadman) Algorithm

There will be a certain equation (distance) that will be used to convert the encrypted message. For example: the equation is s=vt. Where v is the position of the character in the uppercase alphabetical word and the t = last iteration number of the text.

Input = ABC; C=t=3 (stored for Decipher)

Output = CFI

In this algorithm we are considering the equation of motion s=vt.

Here, A = 65(ASCII)

We subtracted by 65 and got 0, then we added 1 as in cases like after subtraction we will get 0 even after the multiplication. Moreover, multiplying with the original value instead of subtracting will make it larger to find its cipher value and will not give the correct value while deciphering.

After multiplication, we added 65 to find the sth position in the alphabetical order which in case of A will be C but s is 3 and adding with 65 will give D(68 and 4th in alphabetical order) therefore, we decrease it by 1 always to find its correct position. Example:

(A=v1)--> (65-65) = 0 +1 = 1*3 = 3 = (3-1)+65 = 67 = C = s1

(B=v2)--> (66-65) = 1 +1 = 2*3 = 6 = (6-1)+65 = 70 = F = s2

(C=v3)--> (67-65) = 3 +1 = 3*3 = 9 = (9-1)+65 = 73 = I = s3

In case of deciphering the text every letter of the ciphered text will be considered as ‘s’ and the ‘t’ is stored earlier (t = length of the text). Here we will first increase s by 1 and then subtract it with A(65). After that we divide it with t to find the order of the word and then we decrease the t by 1. Finally, we add 65 to find the correct word.

Here, we will find out the ‘v’ of every letter-

[The encrypted text is CFI]

V1 = s1/t =>(3/3) =1 -1+65 >> A [65]

V2 = s2/t =>(6/3) =2 -1+65 >> B [66]

V3 = s3/t => (9/3) =3 -1+65 >> C [67]

Algorithm-3: Caesar Algorithm

This algorithm takes input from the user. Then shift each letter by 3 characters. We can provide capital and small letter input at the same time. For Cipher, each character in the message is shifted forward by three bits. For Decipher, each character is shifted backward by three bits. The modified message is then displayed on the screen. The examples are given below.

For Cipher:

Input = AbC.

Output = DeF

Input = XYZ

Output = [/]

For Decipher:

Input = DeF

Output = AbC

Algorithm-4: Mirror Cipher-Decipher Algorithm using Stack

Cipher:

A message is provided by the user. Then every letter of the message is pushed into the stack(to make the message in a reversed form). Then we pop every element from the stack and by adding 10h to every element we insert the elements in a defined array. Then we simply view the elements from the array in the interface.

Decipher:

Two ways to do it:

3) By choosing to decipher in insertion of text
4) By deciphering the text that is already ciphered in the emulator

For the first case the inputs will be pushed similarly like Cipher. However, this while popping the values from the stack we will be subtracting the value by 10h that we added while ciphering. Here, push and pop is used once again to reverse order that was reversed while ciphering, thus giving us the actual message.

For the second case we start indexing the values that were stored in an array during cipher. The process will be the same as before for deciphering but instead of typing the inputs we will take the inputs by loop from the range we found earlier and the values that were stored earlier.

Cipher example:

Input: GREAT GOOD

After push and pop : DOOG TAERG

Output: T__W0dQUb

Decipher example:

Input: T__W0dQUbW

After push and pop : WbUQd0W__T

Output: GREAT GOOD