14.08.2020

Ex No: 1.D

IMPLEMENTATION OF VIGENERE CIPHER

AIM:

To implement the Vigen ere Cipher substitution technique using C program.

DESCRIPTION:

To encrypt, a table of alphabets can be used, termed a tabula recta, Vigenère square, or Vigenère table. It consists o f the alphabet written out 26 times in differe nt rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar ciphers. At different points in the encryption process, the cipher uses a different alphabet from one o f the rows. The alphabet used at each point depends on a repeating keyword.

Each row starts with a key letter. The remainder of the row holds the letters A to Z. Although there are 26 key row s shown, you will only use as many keys as there are unique letters in the key string, here just 5 keys, {L, E, M, O, N}. For successive letters of the message, we are going to take successive letters of the key string, and encipher each message letter using its corresponding k ey row. Choose the next letter of the key, go alo ng that row to find the column heading that m atches the message character; the letter at the intersection of [key-row, msg-col] is the encip hered letter.

ALGORITHM:

ALGORITHM:

STEP-1: Arrange the alphabets in row and column of a 26*26 matrix.

STEP-2: Circulate the alphabets in each row to position left such that the first letter isattached to last.

STEP-3: Repeat this process for all 26 rows and construct the final key matrix.

STEP-4: The keyword and the plain text is read from the user.

STEP-5: The characters in the keyword are repeated sequentially so as to match withthat of the plain text.

STEP-6: Pick the first letter of the plain text and that of the keyword as the row indicesand column indices respectively.

STEP-7: The junction character where these two meet forms the cipher character.

STEP-8: Repeat the above steps to generate the entire cipher text.

PROGRAM:

```
// C++ code to implement Vigenere Cipher
#include<bits/stdc++.h>
using namespace std;

// This function generates the key in
// a cyclic manner until it's length isi'nt
// equal to the length of original text
string generateKey(string str, string key)
{
    int x = str.size();

    for (int i = 0; ; i++)
    {
        if (x == i)
            i = 0;
        if (key.size() == str.size())
            break;
        key.push_back(key[i]);
    }
    return key;
}

// This function returns the encrypted text
// generated with the help of the key
string cipherText(string str, string key)
{
    string cipher_text;

    for (int i = 0; i < str.size(); i++)
    {
        // converting in range 0-25
```

```
        char x = (str[i] + key[i]) %26;

        // convert into alphabets(ASCII)
        x += 'A';

        cipher_text.push_back(x);
    }
    return cipher_text;
}

// This function decrypts the encrypted text
// and returns the original text
string originalText(string cipher_text, string key)
{
    string orig_text;

    for (int i = 0 ; i < cipher_text.size(); i++)
    {
        // converting in range 0-25
        char x = (cipher_text[i] - key[i] + 26) %26;

        // convert into alphabets(ASCII)
        x += 'A';
        orig_text.push_back(x);
    }
    return orig_text;
}

// Driver program to test the above function
int main()
{
    string str = "Geedha.India";
    string keyword = "AYUSH";

    string key = generateKey(str, keyword);
    string cipher_text = cipherText(str, key);

    cout << "Ciphertext : "
         << cipher_text << "\n";

    cout << "Original/Decrypted Text : "
         << originalText(cipher_text, key);
    return 0;
}
```

OUTPUT:

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
Ciphertext : GIEBUGFCLQOE
Original/Decrypted Text : GKKJNGHITJOG>
```

RESULT:

Thus the Vigenere Cipher substitution technique had been implemented successfully.