Cryptography and Network Security Lab

PRN: 2019BTECS00090

Name: Udaykumar Gadikar

Batch: B8

Assignment No. 4

Title:

Vigenère Cipher Encryption-Decryption

Aim:

To implement Vigenère Cipher Encryption-Decryption using Console and file input.

Theory:

The Vigenère cipher is a polyalphabetic substitution cipher that is a natural evolution of the Caesar cipher. The Caesar cipher encrypts by shifting each letter in the plaintext up or down a certain number of places in the alphabet. If the message was right shifted by 4, each A would become E, and each S would become W. In the Vigenère cipher, a message is encrypted using a secret key, as well as an encryption table (called a Vigenère square, Vigenère table, or tabula recta).

Procedure:

Encryption:

Let's take this plaintext phrase as an example:

IMPROVE YOUR PUZZLE SOLVING SKILLS

After finalizing the plaintext, the person encrypting would then pick a secret key, which would help encrypt and decrypt the message. Our example secret key here is:

BOXENTRIQ

The next step is repeating the secret key enough times so its length matches the plain text.

IMPROVE YOUR PUZZLE SOLVING SKILLS

BOXENTR IQBO XENTRI QBOXENT RIQBOX

Once the two lines are split into five-letter groups, start encrypting. Take one letter from the plaintext group and a letter from the secret key group (we're going to start with I and B), add them (take mod of 26 if necessary) For this example, the first letter of the encrypted cipher text is J.

Once you've done that for every character, your final encrypted text should look like this:

JAMVB OVGEV FMYMS CMIPZ SMAZJ SYMZP

<u>Decryption</u>

JAMVB OVGEV FMYMS CMIPZ SMAZJ SYMZP

BOXENTR IQBO XENTRI QBOXENT RIQBOX

Subtract given key from cipher text in order to obtain plain text (take mod of 26 if necessary)

You get Plain Text:

IMPROVE YOUR PUZZLE SOLVING SKILLS

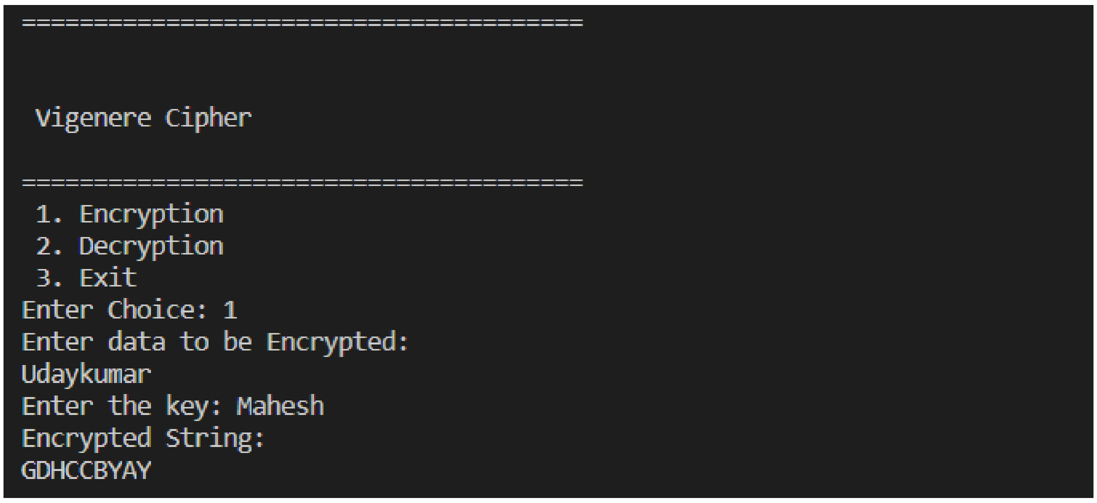
<u>Code</u>:

```
#include <bits/stdc++.h>

using namespace std;

string key;
void func(string k)
{
    key.clear();
    for (int i = 0; i < k.size(); ++i)
    {
        if (k[i] >= 'A' && k[i] <= 'Z')
            key += k[i];
        else if (k[i] >= 'a' && k[i] <= 'z')
            key += k[i] + 32;
    }
}
```

```
string encryption(string t)
{
    string output;
    for (int i = 0, j = 0; i < t.length(); ++i)
    {
        char c = t[i];
        if (c == ' ')
            continue;
        if (c >= 'a' && c <= 'z')
            c += 32;
        else if (c < 'A' || c > 'Z')
            continue;
        output += (c + key[j] - 2 * 'A') % 26 + 'A';
        j++;
        j = j % key.size();
    }
    return output;
}
string decryption(string t)
{
    string output;
    for (int i = 0, j = 0; i < t.length(); ++i)
    {
        char c = t[i];
        if (c >= 'a' && c <= 'z')
            c += 'A' - 'a';
        else if (c < 'A' || c > 'Z')
            continue;
        output += (c - key[j] + 26) % 26 + 'A';
        j = (j + 1) % key.length();
    }
    return output;
}

int main()
{

    int choice;
    int datachoice;
    string sample, key;
    int shift;
    cout << "=======================================\n\n\n Vigenere Cipher \n\
n=======================================";

    while (1)
    {
        cout << "\n 1. Encryption \n 2. Decryption\n 3. Exit\nEnter Choice: ";
        cin >> choice;
        if (choice > 2)
            break;
        switch (choice)
        {
        case 1:

            cout << "Enter data to be Encrypted:\n";
            cin.ignore();
            getline(cin, sample);
            cout << "Enter the key: ";
            getline(cin, key);
            func(key);
            cout << "Encrypted String:\n";
            cout << encryption(sample) << endl;
            break;

        case 2:
            cout << "Enter data to be Decrypted:\n";
            cin.ignore();
            getline(cin, sample);
            cout << "Enter the key: ";
            getline(cin, key);
            func(key);
            cout << "Decrypted String:\n";
            cout << decryption(sample) << endl;
            ;
            break;
        }
    }
    return 0;
}
```

Output:

Console Input:



```
========================================


 Vigenere Cipher

========================================
 1. Encryption
 2. Decryption
 3. Exit
Enter Choice: 1
Enter data to be Encrypted:
Udaykumar
Enter the key: Mahesh
Encrypted String:
GDHCCBYAY
```

Conclusion:

It is a more complex cipher than the Caesar cipher and encrypting a message using the Vigenère cipher is also more secure when compared to that using the Caesar cipher. The Vigenère cipher, just like the Caesar cipher, belongs to a specific subset of encryption scheme called the Polyalphabetic substitution ciphers.