Name: Karuna Bajirao Randive.

Prn: 2020BTECS00024

Batch: B2

Cryptography & Network Security Lab

Assignment No. 3

Theory:

The Vigenere cipher is an algorithm that is used to encrypting and decrypting the text. The Vigenere cipher is an algorithm of encrypting an alphabetic text that uses a series of interwoven Caesar ciphers. It is based on a keyword's letters. It is an example of a polyalphabetic substitution cipher. This algorithm is easy to understand and implement.

Encryption:

- Generate key by repeating its characters of length of the plain text.
- Add each character in key to the corresponding character in plain text.
- Perform Mod26 operation so that number do not exceed the limit.
- Print the cipher text.

Decryption:

- Generate key by repeating its characters of length of the cipher text.
- Subtract each character in key from corresponding character in cipher text.
- Subtract 26 so that number cannot be negative.
- Perform Mod26 operation so that number do not exceed the limit.
- Print the plain text.

Advantages:

- The cipher disguises plaintext letter frequency.
- Like most other polyalphabetic substitution ciphers the main idea was to create a cipher that would disguise letter frequency which greatly interfered with frequency analysis methods.
- Largely uncrackable without knowledge of methods
- It is incredibly difficult to find the key other than through brute-force methods.
- Large key space.

Disadvantages:

- Repeating nature of the key (largest weakness that leads to other weaknesses)
- The security of the Vigenère Cipher is heavily reliant on the length and strength of the keyword. If the keyword is too short or weak, it becomes easier to break the encryption using statistical analysis.
- The Vigenère Cipher is not considered suitable for secure communications in modern times due to its vulnerabilities and the availability of more advanced encryption methods.

Vigenere Cipher:

```
#include<bits/stdc++.h>
using namespace std;

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;
}

string cipherText(string str, string key){
    string cipher_text;

    for (int i = 0; i < str.size(); i++){
        char x = (str[i] + key[i]) %26;
        x += 'A';
        cipher_text.push_back(x);
    }
    return cipher_text;
}

string originalText(string cipher_text, string key){
    string orig_text;

    for (int i = 0 ; i < cipher_text.size(); i++){
        char x = (cipher_text[i] - key[i] + 26) %26;
        x += 'A';
        orig_text.push_back(x);
    }
    return orig_text;
}

int main(){
```

```
    cout<<"Enter 1 for encryption and 2 for decryption: ";
    int x;
    cin>>x;

    string keyword,key;
    cout<<"Enter keyword : ";
    cin>>keyword;

    string s;
    switch (x){
    case 1:
        cout<<"Plain text : ";
        cin>>s;

        key = generateKey(s, keyword);
        cout << "Ciphertext : "<< cipherText(s, key)<< "\n";
        break;

    case 2:
        cout<<"Cipher text : ";
        cin>>s;
        key = generateKey(s, keyword);
        cout << "Original/Decrypted Text : "<< originalText(s, key);
        break;
    default:
        break;
    }
    return 0;
}
```

Output:

```
PS C:\Users\Shree Ram Samarth\Documents\CNS> cd "c:\U
re -lbgi -lgdi32 -lcomdlg32 -luuid -loleaut32 -lole32
Enter 1 for encryption and 2 for decryption: 1
Enter keyword : password
Plain text : HelloKaruna
Ciphertext : CQPPWEDGVZE
PS C:\Users\Shree Ram Samarth\Documents\CNS\Assign03>
```