Name : Ritesh Pawar
PRN : 2020BTECS00068

Cryptography and Network Security Lab

Name: Ritesh Pawar

PRN: 2020BTECS00068

Batch: B5

VIGENERE ALGORITHM

Aim:

To encrypt plain text using vigenere cipher and convert cipher text into plain text by decryption.

Theory:

Vigenere Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets. The encryption of the original text is done using the Vigenère square or Vigenère table.

Code:

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


int main()
{
  int choice;
  cout << "Choose an option:\n";
  cout << "1. Encryption\n";
  cout << "2. Decryption\n";
  cout << "Enter your choice (1 or 2): ";
  cin >> choice;
  cin.ignore(); // Clear the newline character from the input buffer


  if (choice == 1)
  {
    // Encryption
    string plainText, key, cipherText;


    cout << "\nEnter plain text: ";
    getline(cin, plainText);
```

Name : Ritesh Pawar
PRN : 2020BTECS00068

```
    cout << "\nEnter key: ";
    getline(cin, key);

    // Removing spaces and converting to lowercase from plaintext
```

```
    string temp = "";
    for (int i = 0; i < plainText.size(); i++)
    {
      if (plainText[i] != ' ')
        temp += plainText[i];
    }
    plainText = temp;

    for (int i = 0; i < plainText.size(); i++)
    {
      if (plainText[i] >= 'A' && plainText[i] <= 'Z')
        plainText[i] += 32; // Convert to lowercase
    }

    // Removing spaces and converting to lowercase from key
    string temp2 = "";
    for (int i = 0; i < key.size(); i++)
    {
      if (key[i] != ' ')
        temp2 += key[i];
    }
    key = temp2;

    for (int i = 0; i < key.size(); i++)
    {
      if (key[i] >= 'A' && key[i] <= 'Z')
        key[i] += 32; // Convert to lowercase
    }

    // Encryption
    for (int i = 0; i < plainText.size(); i++)
    {
      int val = plainText[i] - 'a' + key[i % key.size()] - 'a';
      cipherText += 'a' + (val % 26);
    }

    cout << "\nCipher Text: " << cipherText << endl;
```

Name : Ritesh Pawar
PRN : 2020BTECS00068

```
}
else if (choice == 2)
{
  // Decryption
  string cipherText, key;


  cout << "\nEnter cipher text: ";
  getline(cin, cipherText);


  cout << "\nEnter key: ";
  getline(cin, key);


  // Removing spaces and converting to lowercase from key
  string temp2 = "";
  for (int i = 0; i < key.size(); i++)
  {
    if (key[i] != ' ')
      temp2 += key[i];
  }
  key = temp2;


  for (int i = 0; i < key.size(); i++)
  {
    if (key[i] >= 'A' && key[i] <= 'Z')
      key[i] += 32; // Convert to lowercase
  }


  // Decryption
  string decrypted = "";
  for (int i = 0; i < cipherText.size(); i++)
  {
    int val = cipherText[i] - 'a' - (key[i % key.size()] - 'a') + 26;
    decrypted += 'a' + (val % 26);
  }


  cout << "\nAfter decryption: " << decrypted << endl;
}
else
{
  cout << "Invalid choice. Please choose 1 or 2." << endl;
}
```

Name : Ritesh Pawar
PRN : 2020BTECS00068

```
  return 0;
}
```

Output:

Encryption:

Decryption:

Limitations:

Periodic Key Repeats: The Vigenère Cipher uses a keyword that repeats throughout the plaintext. The repetition of the keyword introduces patterns in the ciphertext, making it vulnerable to frequency analysis. If the keyword is short or has a recognizable pattern, cryptanalysts can deduce it, compromising the entire encryption.

Kasiski Examination: Cryptanalysts can use the Kasiski examination to identify repeated sequences in the ciphertext. By finding repeated sequences, they can estimate the length of the keyword and potentially recover parts of the key, making decryption easier.

Kerckhoffs's Principle Violation: The Vigenère Cipher violates Kerckhoffs's principle, which states that the security of a cryptographic system should not rely on the secrecy of the algorithm but only on the secrecy of the key. In the Vigenère Cipher, the security depends on both the key and the algorithm, which is a weakness.

Name : Ritesh Pawar
PRN : 2020BTECS00068

Key Length Determination: If the length of the keyword is unknown, determining the correct key length can be challenging. However, techniques such as the Friedman test and the Index of Coincidence can help cryptanalysts estimate the key length, making decryption more feasible.

Frequency Analysis Resistance is Limited: While the Vigenère Cipher is more resistant to simple frequency analysis compared to the Caesar Cipher, it is not entirely immune. Longer messages with repeating keywords can still exhibit frequency patterns that skilled cryptanalysts can exploit.