Experiment 04

Aim: Write a program in Java or Python to perform Cryptanalysis or decoding Vigenere Cipher.

Roll No.	37
Name	Mikil Lalwani
Class	D15-B
Subject	Internet Security Lab
LO Mapped	LO1: To apply the knowledge of symmetric cryptography to implement classical ciphers.

Aim -

Write a program in Java or Python to perform Cryptanalysis or decoding of
Vigenere cipher.

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.
The table consists of the alphabets written out 26 times in different 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 of the rows.
The alphabet used at each point depends on a repeating keyword.

Algorithm-

Encryption-

1. Take plain text and key from the user.
2. Repeat the key and make it equal to the length of the plain text.
3. Now add the value of each element of plain text with the corresponding key element and perform mod 26 to get a new value.
4. Substitute the value with the respective letter to get the ciphered text.

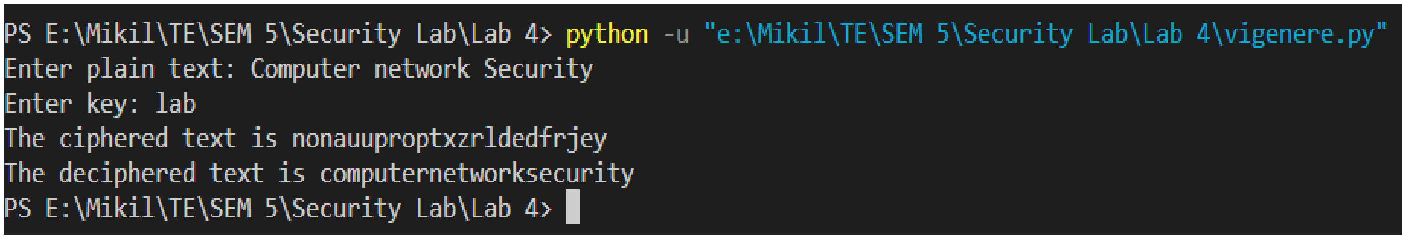
Decryption-

1. Take the ciphered text and repeated key and subtract the value of the corresponding letters and add 26.
2. Now perform mod 26 to get the value of the plain text.

Code-

```
import string

# creating a new key equal to string
def keygenerator(key,plaintext):
    q = len(plaintext)//len(key)
    rem = len(plaintext)%len(key)

    newkey = ""
    newkey = newkey + key*q + key[0:rem]
    return newkey

#Encryption
def encrypt(plaintext,newkey):
    ciphertext = ""
    for i in range(len(plaintext)):
        cipher = ""
        cipher = (letters.index(plaintext[i]) + letters.index(newkey[i]))%26
        ciphertext = ciphertext + letters[cipher]
    return ciphertext

def decrypt(ciphertext,newkey):
    deciphertext = ""
    for i in range(len(ciphertext)):
        decipher = ""
        decipher = (letters.index(ciphertext[i]) - letters.index(newkey[i]) + 26)%26
        deciphertext = deciphertext + letters[decipher]
    return deciphertext

letters = list(string.ascii_lowercase)

# Taking plain text from user and converting to lowercase.
plaintext = input("Enter plain text: ")
plaintext = plaintext.lower()

ptext = ""
# Removing all spaces and punctuation
for i in plaintext:
    if(i in letters):
        ptext = ptext + i

plaintext = ptext
del(ptext)

# Taking key from user.
key = input("Enter key: ")
key = key.lower()

ktext = ""

# Removing all spaces and punctuation
for i in key:
    if(i in letters):
        ktext = ktext + i

key = ktext
del(ktext)
newkey = keygenerator(key,plaintext)
ciphertext = encrypt(plaintext,newkey)
print("The ciphered text is "+ciphertext)

decipher = decrypt(ciphertext,newkey)
print("The deciphered text is "+decipher)
```

Screenshot-



```
PS E:\Mikil\TE\SEM 5\Security Lab\Lab 4> python -u "e:\Mikil\TE\SEM 5\Security Lab\Lab 4\vigenere.py"
Enter plain text: Computer network Security
Enter key: lab
The ciphered text is nonauuproptxzrldedfrjey
The deciphered text is computernetworksecurity
PS E:\Mikil\TE\SEM 5\Security Lab\Lab 4> 
```

Conclusion-

We have successfully implemented Vigenere cipher.