IMPLEMENTING SIMPLE CAESAR CIPHERS AND BREAKING IT USING FREQUENCY ANALYSIS

SUBJECT NAME: CRYPTOGRAPHY AND NETWORK SECURITY

SUBJECT CODE: CS6008

MODULE: 5

NAME	BHUVANESHWAR S
REG.NO	2019103513
DATE	05/04/2022

AIM :

To implement a simple Caesar cipher and crack the cipher text using frequency analysis.

TOOL INVOLVED:

- Python
- Terminal
- Visual Studio Code

PROBLEM DESCRIPTION:

Caesar cipher is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. It is also known as additive cipher.

The encryption can also be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A → 0, B → 1, ..., Z → 25. Encryption of a letter x by a shift n can be described mathematically as

$$E_n(x) = (x + n) \mod 26.$$

For decryption

$$D_n(x) = (x - n) \mod 26.$$

Caesar cipher can be easily broken using

- brute-force attack
- frequency analysis

Brute-force attack:

A brute-force attack tries every possible decryption key for a cipher. Nothing stops a cryptanalyst from guessing one key, decrypting the ciphertext with that key, looking at the output, and then moving on to the next key if they didn't find the secret message.

Frequency analysis:

Frequency analysis is one of the known ciphertext attacks. It is based on the study of the frequency of letters or groups of letters in a ciphertext. The attacker usually checks some possibilities and makes some substitutions of letters in ciphertext. He looks for possible appearing words and based on that makes more substitutions. Using computers, it is possible to try a lot of combinations in relative short time.

KEY = ((index of most frequency letter) – (index of character 'e' i.e. 4))mod 26

INPUT:

Getting an input cipher text from the user in terminal.

OUTPUT:

Crack the cipher text using frequency analysis.

SCREENSHOT:

Encryption.py

```
alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

def encryption(plain_text,key):
    cipher_text = ''

    for i in range(len(plain_text)):
        char = plain_text[i]
        # print(char)
        if char.isupper():
            index = alphabet.index(char.upper())
            cipher_text += alphabet[(index+key)%26].upper()
        elif char.islower():
            index = alphabet.index(char.upper())
            cipher_text += alphabet[(index+key)%26].lower()
        else:
            cipher_text += char
    return cipher_text

plain_text = input("Enter plain text : ")
key = int(input("Enter key : "))
print(encryption(plain_text,key))
```

OUTPUT

```
PS E:\clg 6th sem\crypto&net security\assignment\statisical_attack> python encryption.py
Enter plain text : Cryptography and Network Security
Enter key : 6
Ixevzumxgvne gtj Tkzcuxq Ykiaxoze
PS E:\clg 6th sem\crypto&net security\assignment\statisical_attack>
```

Here, the plain text is ‘Cryptography and Network security’ to encrypt the message using key value 6.

Decryption.py

```
LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

def decryption(cipher_text,key):
    plain_text = ''

    for i in range(len(cipher_text)):
        char = cipher_text[i]

        if char.isupper():
            index = LETTERS.index(char.upper())
            plain_text += LETTERS[(index-key)%26].upper()
        elif char.islower():
            index = LETTERS.index(char.upper())
            plain_text += LETTERS[(index-key)%26].lower()
        else:
```

```
            plain_text += char

    return plain_text

cipher_text = input("Enter cipher text : ")
key = int(input("Enter key : "))
print(decryption(cipher_text,key))
```

OUTPUT

```
PS E:\clg 6th sem\crypto&net security\assignment\statisical_attack> python decryption.py
Enter cipher text : Ixevzumxgvne gtj Tkzcuxq Ykiaxoze
Enter key : 6
Cryptography and Network Security
```

Decrypting the text using the key value 6

attack.py

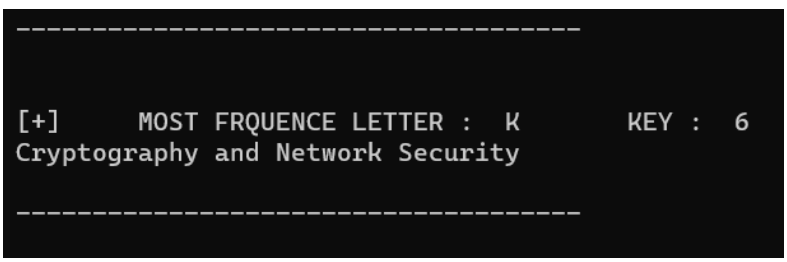
```
from audioop import reverse


LETTERS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

def frequency_analysis(cipher_text):
    cipher_text = cipher_text.upper()

    letter_frequency = {}

    for i in LETTERS:
        letter_frequency[i] = 0;

    for i in cipher_text:
        if i in LETTERS:
            letter_frequency[i] += 1

    return letter_frequency

def decryption(cipher_text,key):
    plain_text = ''

    for i in range(len(cipher_text)):
        char = cipher_text[i]

        if char.isupper():
            index = LETTERS.index(char.upper())
            plain_text += LETTERS[(index-key)%26].upper()
        elif char.islower():
            index = LETTERS.index(char.upper())
            plain_text += LETTERS[(index-key)%26].lower()
        else:
            plain_text += char

    return plain_text
```

```python
def caeser_crack(cipher_text):
    letter_frequency = frequency_analysis(cipher_text)
    print(letter_frequency)

    letter_frequency = {k: v for k, v in sorted(letter_frequency.items(), key=lambda item:
item[1],reverse=True)}

    for x,y in letter_frequency.items():
        if y != 0:
            index_l = LETTERS.index(x)
            index_e = LETTERS.index('E')

            key = (index_l - index_e)%26
            print("\n[+]\tMOST FRQUENCE LETTER : " ,x,"\tKEY : ",key)
            print(decryption(cipher_text,key))
            print("\n-------------------------------------\n")



cipher_text = input("Enter cipher text : ")
print(cipher_text)
caeser_crack(cipher_text)
```

OUTPUT

```
PS E:\clg 6th sem\crypto&net security\assignment\statisical_attack> python attack.py
Enter cipher text : Ixevzumxgvne gtj Tkzcuxq Ykiaxoze
Ixevzumxgvne gtj Tkzcuxq Ykiaxoze
{'A': 1, 'B': 0, 'C': 1, 'D': 0, 'E': 3, 'F': 0, 'G': 2, 'H': 0, 'I': 2, 'J': 1, 'K': 2, 'L': 0, 'M': 1, 'N': 1, 'O': 1, 'P': 0, 'Q': 1, 'R': 0, 'S': 0, 'T'
: 2, 'U': 2, 'V': 2, 'W': 0, 'X': 4, 'Y': 1, 'Z': 3}

[+]     MOST FRQUENCE LETTER :  X       KEY :  19
Pelcgbtencul naq Argjbex Frphevgl

-------------------------------------


[+]     MOST FRQUENCE LETTER :  E       KEY :  0
Ixevzumxgvne gtj Tkzcuxq Ykiaxoze

-------------------------------------


[+]     MOST FRQUENCE LETTER :  Z       KEY :  21
Ncjaezrclasj lyo Ypehzcv Dpnfctej

-------------------------------------


[+]     MOST FRQUENCE LETTER :  G       KEY :  2
Gvctxskvetlc erh Rixasvo Wigyvmxc

-------------------------------------


[+]     MOST FRQUENCE LETTER :  I       KEY :  4
Etarvqitcrja cpf Pgvyqtm Ugewtkva

-------------------------------------


[+]     MOST FRQUENCE LETTER :  K       KEY :  6
Cryptography and Network Security
```

It display the number of frequency for each letter in cipher text.



Hence we crack the plain text using frequency analysis and found the value of key.