Experiment 2

Shashwat Shah
60004220126
TY Btech Comps B

Aim: Study and implement vigenere cipher

Theory: It is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is an cipher based on substitution, using multiple substitution alphabets. The encryption of the original text is done using the vigenere square or vigenere table.

The table consists of the alphabets written out 26 times in different rows, each alphatet shifted cyclically to the left compared to the previous alphabets, corresponding to the 26 possible Caesar ciphers

A more easy implementation could be to visually check vigenere alphabetically by converting (A-Z) into numbers (0-25)

Encryption - $E_i = (P_i + k_i) \bmod 26$

Decryption - $D_i = (E_i - k_i) \bmod 26$

Example

plaintext = PRERAN

Key = BEST

P	R	E	R	A	N
15	17	4	17	0	13

B	E	S	T	B	E
01	04	18	19	01	04

∴ Encrypted.

Q	V	W	K	O	E
16	21	22	10	14	04

∴ The encrypted text is 'QVWKOE'

Conclusion: The time complexity to convert the string into cipher text is O(n) where n is the length of the string. The space complexity is O(n).

Hence, we studied and implemented the Vignere cipher

EXPERIMENT 2

Shashwat Shah
TYBtech Comps B
C22
60004220126

AIM: Study and Implement Vigenere Cipher.

CODE:

```
def generateKey(string, key):
    key = list(key)
    if len(string) == len(key):
        return(key)
    else:
        for i in range(len(string) -
                       len(key)):
            key.append(key[i % len(key)])
    return("" . join(key))

def cipherText(string, key):
    cipher_text = []
    for i in range(len(string)):
        x = (ord(string[i]) +
             ord(key[i])) % 26
        x += ord('A')
        cipher_text.append(chr(x))
    return("" . join(cipher_text))

def originalText(cipher_text, key):
    orig_text = []
    for i in range(len(cipher_text)):
        x = (ord(cipher_text[i]) -
             ord(key[i]) + 26) % 26
        x += ord('A')
        orig_text.append(chr(x))
    return("" . join(orig_text))

if __name__ == "__main__":

    string = input("Enter your message: ")
    keyword = input("Enter key: ")
    key = generateKey(string, keyword)
    cipher_text = cipherText(string,key)
```

```
    print("Ciphertext :", cipher_text)
    print("Original/Decrypted Text :",
        originalText(cipher_text, key))
```

OUTPUT:

```
/BTech/Docs/6th Sem/IS/Code/Exp2/Vigenere.py"
Enter your message: HITHISIS
Enter key: VIGENERE
Ciphertext : CQZLVWZW
Original/Decrypted Text : HITHISIS
```