Lab – 1

Subject : NIS

Aim: Write a program to implement Additive Cipher and Monoalphabetic Substitution Cipher

1. Additive Cipher

Additive cipher is a one of the encryption technique of type substitution cipher.it basically shift each alphabet with key suppose your plaintext's character is 'x' and key is 2 then it will shift two position so your cipher text is 'z'.it also known as Caesar cipher or shift cipher.

Same way decryption takes place each alphabets will shift back as par the key. Cryptanalysis is very much in this kind of encryption technique, only 25 attempt or less and intruder can have most probable valid guess of plaintext.

Program: -

```
def encrypt(msg,key):
    l=list(msg)
    l1=len(l)
    for i in range(l1):
        l[i]=chr((ord(l[i])-97+key)%26+97)
    return ("".join(l))
def decrypt(msg,key):
    l=list(msg)
    l1=len(l)
    for i in range(l1):
        l[i]=chr((ord(l[i])-97-key)%26+97)
    return ("".join(l))
def cryptanalysis(enc_msg):
    l=list(enc_msg)
    l1=len(l)
    for k in range(0,26):
```

```
        for i in range(l1):
            l[i]=chr((ord(l[i])-97-k)%26+97)
        print("".join(l))
        l=list(enc_msg)
    return 0

m,key=input().split()
key=int(key)
enc_msg=encrypt(m,key)
print("Encrypted Message:"+enc_msg)
dec_msg=decrypt(enc_msg,key)
print("Decrypted Message:"+dec_msg)
print("Cryptanalysis:")
final_key=cryptanalysis(enc_msg)
```

Output: -

```
PS D:\DDIT CE\Sem 6\NIS\Lab 1> python .\AdditiveCipher.py
amrut 5
Encrypted Message:frwzy
Decrypted Message:amrut
Cryptanalysis:
frwzy
eqvyx
dpuxw
cotwv
bnsvu
amrut
zlqts
ykpsr
xjorq
winqp
vhmpo
uglon
tfknm
sejml
rdilk
qchkj
pbgji
oafih
nzehg
mydgf
lxcfe
kwbed
jvadc
iuzcb
htyba
gsxaz
PS D:\DDIT CE\Sem 6\NIS\Lab 1>
```

2. Monoalphabetic Substitution Cipher

In monoalphabetic substitution Cipher, there is one to one mapping of the alphabets. Plaintext will be replaced by its corresponding mapped character and this is how encryption takes place. reverse process is done in decryption. cryptanalysis is not practically possible because there are 26! Possible outcomes and it is hard to find plaintext from this large number of outcames.

Program: -

```
def encrypt(msg):

    cipher=[]

    for char in msg:

        cipher.append(map_list2[map_list1.index(char)])

    return "".join(cipher)

def decrypt(msg):

    plain=[]

    for char in msg:

        plain.append(map_list1[map_list2.index(char)])

    return "".join(plain)



map_list1=['a','b','c','d','
','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','
w','x','y','z','e']

map_list2=['c','e','f','j','k','l','n','o','s','p','w','z','y','x','a',
'd','g','h','i','b','m','r','q','t','u','v',' ']
```

```
msg=input("Enter text: ")

enc_msg=encrypt(msg)

print("Encrypted text: "+enc_msg)

dec_msg=decrypt(enc_msg)

print("Decrypted text: "+dec_msg)
```

Output: -

```
PS D:\DDIT CE\Sem 6\NIS\Lab 1> python .\MonoAlphabaticCipher.py
Enter text: hello world
Encrypted text: o zzakqahzj
Decrypted text: hello world
PS D:\DDIT CE\Sem 6\NIS\Lab 1>
```