**Practical – 3**

**Aim : Explain Encryption & Decryption and implement different types of ciphers.**

Sol.

*Encryption :* Encryption is a way of scrambling data so that only authorized parties can understand the information. In technical terms, it is the process of converting plaintext to ciphertext. In simpler terms, encryption takes readable data and alters it so that it appears random.

*Decryption :* The conversion of encrypted data into its original form is called Decryption. It is generally a reverse process of encryption. It decodes the encrypted information so that an authorized user can only decrypt the data because decryption requires a secret key or password.

* ADFGVX Cipher :

Code :

from pycipher import ADFGVX

def func():

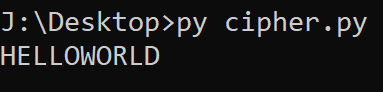
adfgvx = ADFGVX(key='PH0QG64MEA1YL2NOFDXKR3CVS5ZW7BJ9UTI8', keyword='GERMAN')

enc = adfgvx.encipher("Hello world!")

dec = adfgvx.decipher(enc)

print (dec)

func()



* Shift Cipher :

Code :

def encrypt(text,s):

result = ""

for i in range(len(text)):

char = text[i]

if (char.isupper()):

result += chr((ord(char) + s-65) % 26 + 65)

else:

result += chr((ord(char) + s - 97) % 26 + 97)

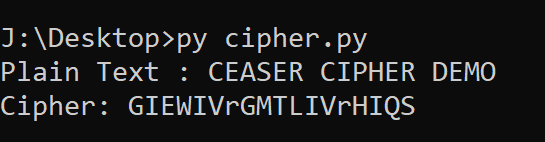
return result

text = "CEASER CIPHER DEMO"

s = 4

print ("Plain Text : " + text)

print ("Cipher: " + encrypt(text,s))



* Substitution Cipher :

Code :

import string

all\_letters= string.ascii\_letters

dict1 = {}

key = 4

for i in range(len(all\_letters)):

dict1[all\_letters[i]] = all\_letters[(i+key)%len(all\_letters)]

plain\_txt= "I am studying Data Encryption"

cipher\_txt=[]

for char in plain\_txt:

if char in all\_letters:

temp = dict1[char]

cipher\_txt.append(temp)

else:

temp =char

cipher\_txt.append(temp)

cipher\_txt= "".join(cipher\_txt)

print("Cipher Text is: ",cipher\_txt)

dict2 = {}

for i in range(len(all\_letters)):

dict2[all\_letters[i]] = all\_letters[(i-key)%(len(all\_letters))]

decrypt\_txt = []

for char in cipher\_txt:

if char in all\_letters:

temp = dict2[char]

decrypt\_txt.append(temp)

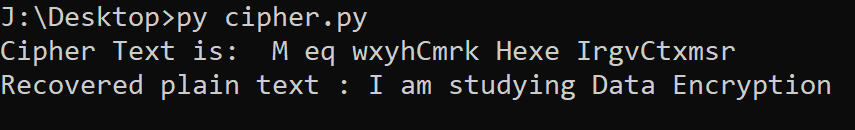
else:

temp = char

decrypt\_txt.append(temp)

decrypt\_txt = "".join(decrypt\_txt)

print("Recovered plain text :", decrypt\_txt)



* Vigenere Cipher :

Code :

from pycipher import Vigenere

def encrypt():

text = "Hello world!"

key = "keykey"

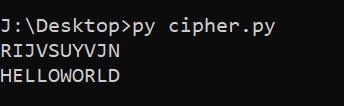
enc\_text = Vigenere(key).encipher(text)

print (enc\_text)

dec\_text = Vigenere(key).decipher(enc\_text)

print (dec\_text)

encrypt()



* *Affine Cipher :*

Code :

from pycipher import Affine

def encrypt():

text = "Hello world!"

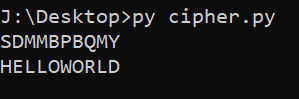
enc\_text = Affine(a=5,b=9).encipher(text)

dec\_text = Affine(a=5,b=9).decipher(enc\_text)

print (enc\_text)

print (dec\_text)

encrypt()



* Permutation cipher :

Code

def split\_len(seq, length):

return [seq[i:i + length] for i in range(0, len(seq), length)]

def encode(key, plaintext):

order = {

int(val): num for num, val in enumerate(key)

}

ciphertext = ''

for index in sorted(order.keys()):

for part in split\_len(plaintext, len(key)):

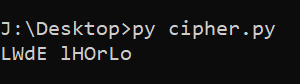
try:ciphertext += part[order[index]]

except IndexError:

continue

return ciphertext

print(encode('3214', 'HELLO World'))

+

* *Playfair cipher :*

Code :

from pycipher import Playfair

def encrypt():

text = "Hello world"

key = "zgptfoihmuwdrcnykeqaxvsbl"

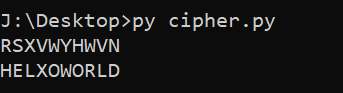
enc\_text = Playfair(key).encipher(text)

dec\_text = Playfair(key).decipher(enc\_text)

print (enc\_text)

print (dec\_text)

encrypt()



* *Enigma Cipher :*

Code :

from pycipher import Enigma

def encrypt():

text = "Hello world!"

enc\_text = Enigma(settings=('A','A','A'),rotors=(1,2,3),reflector='B',ringstellung=('F','V','N'),steckers=[('P','O'),('M','L'),('I','U'),('K','J'),('N','H'),('Y','T'),('G','B'),('V','F'),('R','E'),('D','C')]).encipher(text)

dec\_text =

Enigma(settings=('A','A','A'),rotors=(1,2,3),reflector='B',ringstellung=('F','V','N'),steckers=[('P','O'),('M','L'),('I','U'),('K','J'),('N','H'),('Y','T'),('G','B'),('V','F'),('R','E'),('D','C')]).decipher(enc\_text)

print (enc\_text)

print (dec\_text)

encrypt()

