**Internal Practical Exam**

**AIM: One Time Pad:**

**1.User will enter text file as a plain text. [PT.TXT]**

**2.Load the PT.TXT file and process for input message.**

**3.Key will be randomly generated from 1 to 26 letters which is store into KEY.TXT file**

**4.Encrypt the PT.TXT file by applying the key from KEY.TXT file.**

**5.Display the Cipher-Text to user**

**6.Decrypt the cipher text and stored the Cipher-Text & Plain-Text pair of messagesin separate OUTPUT.TXT file.**

**7.Perform the cryptanalysis by applying brute force attack over cipher-text.**

**Programming:Functions:**

**1.key\_Generation()**

**2.encrypt\_Text()**

**3.decrypt\_Text()**

**4.Cryptanalytics()**

**Program code:**

import string

import random

def encrypttext(text, keyv, characters = string.ascii\_lowercase):

#keyv = keyv[:-1]

keyarr = [int(x) for x in keyv.split()]

#keyarr = list(map(int, keyv.split(" ")))

arr = list(text)

print(arr)

print(keyarr)

for i in range(len(text)):

arr[i] = ord(arr[i]) + keyarr[i]

#print(arr[i])

if arr[i] > 122:

arr[i] = arr[i] - 26

arr[i] = chr(arr[i])

#print(arr[i],ord(arr[i]))

str1 = ""

translated\_text = str1.join(arr)

return translated\_text

def decrypttext(text, keyv, characters = string.ascii\_lowercase):

#keyarr = list(map(int, keyv.split(" ")))

keyarr = [int(x) for x in keyv.split()]

arr = list(text)

for i in range(len(text)):

arr[i] = ord(arr[i]) - keyarr[i]

if arr[i] < 97:

arr[i] = arr[i] + 26

arr[i] = chr(arr[i])

str1 = ""

translated\_text = str1.join(arr)

return translated\_text

def fileCipher(fileName, outputFileName, key , crypt):

with open(key, "r") as f\_in:

keyv = f\_in.readline()

with open(fileName, "r") as f\_in:

with open(outputFileName, "w") as f\_out:

# iterate over each line in input file

for line in f\_in:

if(crypt == "encrypt"):

#encrypt/decrypt the line

lineNew = encrypttext(line, keyv)

print(lineNew)

f\_out.write(lineNew)

else:

f\_out.write(line)

f\_out.write("\n")

lineNew = decrypttext(line, keyv)

print(lineNew)

f\_out.write(lineNew)

print("The file {} has been translated successfully and saved to {}".format(fileName, outputFileName))

def key\_Generation(fileName, key):

with open(fileName, "r") as f\_in:

data = f\_in.read()

leng = len(data)

y =''

for i in range(leng):

x = random.randint(1,26)

y = y + str(x) + ' '

with open(key, "w") as f\_out:

f\_out.write(y)

inputFile = "./plaintext1.txt"

outputFile = "./plaintext1\_encrypted.txt"

key = "./key.txt"

key\_Generation(inputFile, key)

fileCipher(inputFile, outputFile, key,"encrypt")

inputFile = "./plaintext1\_encrypted.txt"

outputFile = "./output.txt"

fileCipher(inputFile, outputFile, key,"decrypt")

#Cryptanalytics(inputFile)

**Output/Screenshot:**



  