LAB Manual

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**Experiment No.01**

**A.1 Aim:**

To implement a simple substitution cipher and a transposition cipher

**A.2 Prerequisite:**

Fundamentals of encryption and decryption process

**A.3 Outcome:**

**After successful completion of this experiment students will be able to** 1. Learn and appreciate the operation of classical ciphers

2. Understand the shortcomings of these ciphers

**A.4 Theory:**

The art and science of keeping messages secure is **cryptography.** A message is **plaintext (**sometimes called **cleartext).** The process of disguising a message in such a way as to hide its substance is **encryption.** An encrypted message is **ciphertext.** The process of turning ciphertext back into plaintext is **decryption.**

A **cryptographic algorithm**, also called a **cipher**, is the mathematical function used for encryption and decryption. Both the encryption and decryption operations use a key. The range of possible values of the key is called the **keyspace.**

* *EK(M) = C*
* *DK(C) = M*
* *DK(EK(M)) = M*

There are two basic types of classical ciphers:

* Transposition ciphers
* Substitution ciphers

A **substitution cipher** is one in which each character in the plaintext is substituted for another character in the ciphertext. The receiver inverts the substitution on the ciphertext to recover the plaintext.

For example:

**Caesar Cipher,** in which each plaintext character is replaced by the character three to the right modulo 26 ("A" is replaced by "D," "B" is replaced by "E,"..., "X“ is replaced by "A," "Y" is replaced by "B," and "Z" is replaced by "C") is a simple substitution cipher.

A **transposition cipher** is obtained by performing some sort of permutation on the plain text symbols. Here, the symbols remain the same but their order is shuffled around.

For example:

In a simple columnar transposition cipher, the plain text is written horizontally row by row in rectangle of fixed width. To form cipher, the text is read off vertically. The order of columns is kept secret and is the key. Decryption involves writing the ciphertext vertically of identical length and then reading the text horizontally. Let me illustrate this with the help of an example:

Consider the sentence: “IT IS RAINING TODAY”

Key value: 4 (rectangular width)

Encryption process:

|  |  |  |  |
| --- | --- | --- | --- |
| I | T | I | S |
| R | A | I | N |
| I | N | G | T |
| O | D | A | Y |

Ciphertext: IRIO TAND IIGA SNTY

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)***

|  |  |
| --- | --- |
| Roll. No. N049 | Name: Tarun Tanmay |
| Class: MBATech CE, 3 Year | Batch: B3 |
| Date of Experiment: | Date of Submission: |
| Grade: | |

**B.1 Software Code written by student:**

**B.2 Input and Output:**

**1) Substitution Cipher:**

en=input("Enter the text to encrypt with no spaces: ")

s=10

r=""

a=0

a=int(input("Enter 1 for encryption and 2 for decryption: "))

if(a==1):

#encryption for loop

for i in range(len(en)):

char=en[i]

if(char.isupper()):

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

else:

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

print("The encrypted string is: ", r)

elif(a==2):

#Decryption for loop

for i in range(len(en)):

char=en[i]

if(char.isupper()):

r+=chr((ord(char) + s+65) % 26+65)

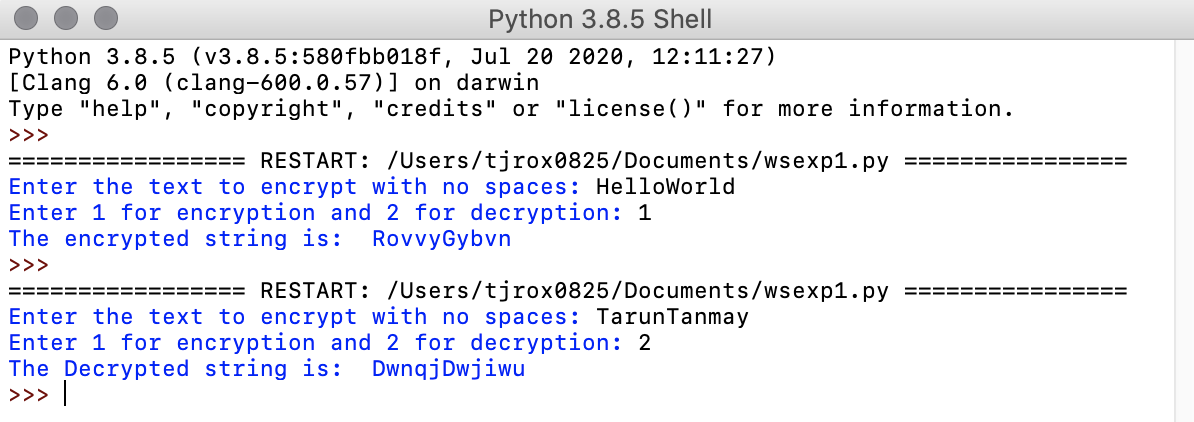
else:

r+= chr((ord(char) + s+97) % 26+97)

print("The Decrypted string is: ",r)

else:

print("Incorrect selection")



**2)**

**Transposition Cipher:**

**#Encryption Transposition Cipher**

def encrypt(message, keyword):

matrix = createEncMatrix(len(keyword), message)

keywordSequence = getKeywordSequence(keyword)

ciphertext = "";

for num in range(len(keywordSequence)):

pos = keywordSequence.index(num+1)

for row in range(len(matrix)):

if len(matrix[row]) > pos:

ciphertext += matrix[row][pos]

return ciphertext

def createEncMatrix(width, message):

r = 0

c = 0

matrix = [[]]

for pos, ch in enumerate(message):

matrix[r].append(ch)

c += 1

if c >= width:

c = 0

r += 1

matrix.append([])

return matrix

def getKeywordSequence(keyword):

sequence = []

for pos, ch in enumerate(keyword):

previousLetters = keyword[:pos]

newNumber = 1

for previousPos, previousCh in enumerate(previousLetters):

if previousCh > ch:

sequence[previousPos] += 1

else:

newNumber += 1

sequence.append(newNumber)

return sequence

msg = input("Enter the Message: ")

key = input("Enter the Keyword: ")

print("The Encrypted Message is: ", encrypt(msg, key))

**A screenshot of a cell phone

Description automatically generated**

**#Decryption Transposition Cipher**

def decrypt(message, keyword):

matrix = createDecrMatrix(getKeywordSequence(keyword), message)

plaintext = "";

for r in range(len(matrix)):

for c in range (len(matrix[r])):

plaintext += matrix[r][c]

return plaintext

def createDecrMatrix(keywordSequence, message):

width = len(keywordSequence)

height = len(message)//width

if height \* width < len(message):

height += 1

matrix = createEmptyMatrix(width, height, len(message))

pos = 0

for num in range(len(keywordSequence)):

column = keywordSequence.index(num+1)

r = 0

while (r < len(matrix)) and (len(matrix[r]) > column):

matrix[r][column] = message[pos]

r += 1

pos += 1

return matrix

def createEmptyMatrix(width, height, length):

matrix = []

totalAdded = 0

for r in range(height):

matrix.append([])

for c in range(width):

if totalAdded >= length:

return matrix

matrix[r].append('')

totalAdded += 1

return matrix

def getKeywordSequence(keyword):

sequence = []

for pos, ch in enumerate(keyword):

previousLetters = keyword[:pos]

newNumber = 1

for previousPos, previousCh in enumerate(previousLetters):

if previousCh > ch:

sequence[previousPos] += 1

else:

newNumber += 1

sequence.append(newNumber)

return sequence

msg = input("Enter the Message: ")

key = input("Enter the Keyword: ")

print("The Decrypted Message is: ", decrypt(msg, key))

**A screenshot of a cell phone

Description automatically generated**

**B.3 Observations and learning:**

In the current age of computers, these ciphers will be cracked easily and are not an acceptable form of encryption for daily use.

**B.4 Conclusion:**

**After successful completion of this experiment I am able to**

1. Learn and appreciate the operation of classical ciphers

2. Understand the shortcomings of these ciphers