**SYSTEM SECURITY**

**LAB 3**

**P.R.N:** 20190802058

**Name:** Krushal Milan Shah

**Experiment:**

1) Write C/C++/Python/JAVA program to implement Rail Fence cipher substitution technique.

2) Write a C/C++/Python/JAVA program to implment row transposi. Consider following information for encipherment.

**Code:**

**1:**

def encryptRailFence(text, key):

rail = [['\n' for i in range(len(text))]

for j in range(key)]

dir\_down = False

row, col = 0, 0

for i in range(len(text)):

if (row == 0) or (row == key - 1):

dir\_down = not dir\_down

rail[row][col] = text[i]

col += 1

if dir\_down:

row += 1

else:

row -= 1

result = []

for i in range(key):

for j in range(len(text)):

if rail[i][j] != '\n':

result.append(rail[i][j])

return ("".join(result))

def decryptRailFence(cipher, key):

rail = [['\n' for i in range(len(cipher))]

for j in range(key)]

dir\_down = None

row, col = 0, 0

for i in range(len(cipher)):

if row == 0:

dir\_down = True

if row == key - 1:

dir\_down = False

rail[row][col] = '\*'

col += 1

if dir\_down:

row += 1

else:

row -= 1

index = 0

for i in range(key):

for j in range(len(cipher)):

if ((rail[i][j] == '\*') and

(index < len(cipher))):

rail[i][j] = cipher[index]

index += 1

result = []

row, col = 0, 0

for i in range(len(cipher)):

# check the direction of flow

if row == 0:

dir\_down = True

if row == key - 1:

dir\_down = False

# place the marker

if (rail[row][col] != '\*'):

result.append(rail[row][col])

col += 1

# find the next row using

# direction flag

if dir\_down:

row += 1

else:

row -= 1

return ("".join(result))

print("Plain text: i love my country")

print("Rows: 3")

print("Encrypted text: " + encryptRailFence("ilovemycountry", 3))

print("Encrypted text: ieorlvmcutyoyn")

print("Rows: 3")

print("Decrypted Text: " + decryptRailFence("ieorlvmcutyoyn", 3))

**2:**

import numpy as np

import string

Plaintext= 'Attack postponed until two am'

Columns= 7

Key= [4 ,3, 1 ,2 ,5 ,6 ,7]

Plaintext = Plaintext.lower()

Plaintext=Plaintext.replace(" ","")

Plaintext\_Array = [char for char in Plaintext ]

Plaintext\_Array= np.array(Plaintext\_Array)

Size\_Array=(int(len(Plaintext\_Array)/Columns))

for i in range (122-Size\_Array+1,123):

Plaintext\_Array=np.append(Plaintext\_Array,chr(i))

Plaintext\_Array

Plaintext\_Array=Plaintext\_Array.reshape(4,7)

Cypher\_Text = []

for i in range (len(Key)):

col = Key[i] -1

Cypher\_Text=np.append(Cypher\_Text,Plaintext\_Array[:,col])

Cypher\_Text

Cypher\_Text\_string=""

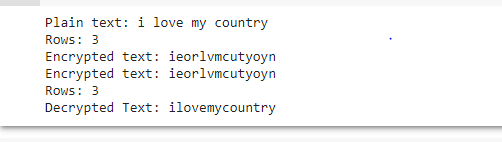
for element in Cypher\_Text:

Cypher\_Text\_string+=element

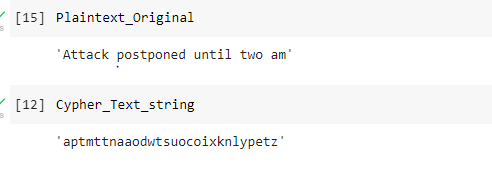
Cypher\_Text\_string

**Results:**

**1**

****

**2**

****

**Conclusion:**

Successfully implemented Rail Fence Cipher and Row Transpose Cipher.