**D15B/37** **Mikil Lalwani** **Security Lab**

# Experiment No 3

## Aim-

Write a program in Java or Python to perform Cryptanalysis or decoding of Playfair Cipher.

## Theory-

The Playfair cipher was the first practical digraph substitution cipher. The scheme was invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher. In Playfair cipher unlike traditional cipher, we encrypt a pair of alphabets(digraphs) instead of a single alphabet.

## Algorithm-

Encryption -

1. Generate the key Square(5×5):
   The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I. The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.

2. Algorithm to encrypt the plain text:
   The plaintext is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter.

Rules for Encryption:

1. If both the letters are in the same column: Take the letter below each one (going back to the top if at the bottom).
2. If both the letters are in the same row: Take the letter to the right of each one (going back to the leftmost if at the rightmost position).
3. If neither of the above rules is true: Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

Decryption -

1. Generate the key Square(5×5) at the receiver's end:
   The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I. The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.

2. Algorithm to decrypt the ciphertext:
   The ciphertext is split into pairs of two letters (digraphs).

Rules for Decryption:

1. If both the letters are in the same column: Take the letter above each one (going back to the bottom if at the top).
2. If both the letters are in the same row: Take the letter to the left of each one (going back to the rightmost if at the leftmost position).
3. If neither of the above rules is true: Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

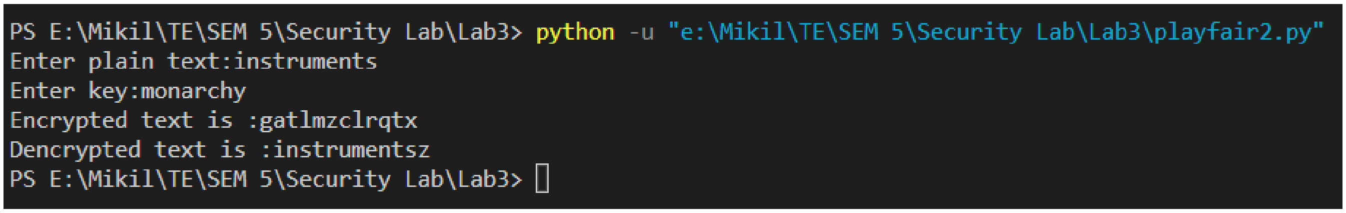
## Code -

```
import string

def modify(txt):
   txt = txt.lower()
   txt = txt.replace(" ","")
   txt = txt.replace("j","i")
   return txt

def pair(plain):
   i=0
   plainpair = []
   while i<len(plain):
      if(i==len(plain)-1):
         plainpair.append((plain[i]+"z"))
         i = i + 1
      elif(plain[i]==plain[i+1]):
         plainpair.append((plain[i]+"x"))
         i = i + 1
```

```
        else:
            plainpair.append((plain[i]+plain[i+1]))
            i = i + 2
    return plainpair

def matrix(key):
    letters = list(string.ascii_lowercase)
    letters.remove("j")
    unique = []
    for i in key:
        if i not in unique:
            unique.append(i)
            letters.remove(i)

    matrix = [["" for i in range(5)] for j in range(5)]

    row = 0
    col = 0
    for i in unique:
        if col == 5:
            row += 1
            col = 0
        matrix[row][col] = i
        col += 1

    for i in letters:
        if col == 5:
            row += 1
            col = 0
        matrix[row][col] = i
        col += 1

    return matrix

def find(z,array):
    for i in range(5):
        for j in range(5):
            if(array[i][j]==z):
                return i,j
```

```
def encrypt(plainpair,array):
    cipher = ""
    for p in plainpair:
        i,j = find(p[0],array)
        m,n = find(p[1],array)

        if (i==m):
            if j==4:
                j=0
            else:
                j+=1
            if n==4:
                n=0
            else:
                n+=1
            cipher += array[i][j]+array[m][n]

        elif (j==n):
            if i==4:
                i=0
            else:
                i+=1
            if m==4:
                m=0
            else:
                m+=1
            cipher += array[i][j]+array[m][n]
        else:
            cipher += array[i][n]+array[m][j]
    return cipher

def dencrypt(cipherpair,array):
    decipher = ""
    for p,q in cipherpair:
        i,j = find(p,array)
        m,n = find(q,array)

        if (i==m):
            if j==0:
                j=4
            else:
                j-=1
            if n==0:
                n=4
            else:
                n-=1
            decipher += array[i][j]+array[m][n]

        elif (j==n):
            if i==0:
                i=4
            else:
                i-=1
            if m==0:
                m=4
            else:
                m-=1
            decipher += array[i][j]+array[m][n]
        else:
            decipher += array[i][n]+array[m][j]
    return decipher

plain = input("Enter plain text:")
plain = modify(plain)

key = input("Enter key:")
key = modify(key)

plainpair = pair(plain)

array = matrix(key)

cipher = encrypt(plainpair, array)
print("Encrypted text is :" + cipher)

cipherpair = pair(cipher)
decipher = dencrypt(cipherpair, array)
print("Dencrypted text is :" + decipher)
```

## Output -



```
PS E:\Mikil\TE\SEM 5\Security Lab\Lab3> python -u "e:\Mikil\TE\SEM 5\Security Lab\Lab3\playfair2.py"
Enter plain text:instruments
Enter key:monarchy
Encrypted text is :gatlmzclrqtx
Dencrypted text is :instrumentsz
PS E:\Mikil\TE\SEM 5\Security Lab\Lab3> 
```

## Conclusion-

Thus we have successfully implemented the Playfair cipher.