## CRYPTOGRAPHY AND NETWORK SECURITY LAB

## MID LAB

Write a Program for encrypting a plain text and decrypting a cipher text using PlayFair Cipher.

**CONCEPT**

**- Playfair cipher: First practical digraph substitution cipher.**

**- Invented in 1854 by Charles Wheatstone, named after Lord Playfair.**

**- Encrypts pairs of alphabets (digraphs) instead of single alphabets.**

**- Reasonably fast to use, requires no special equipment.**

**To generate the key square matrix (5×5) for the Playfair cipher:**

**1. Create a 5×5 grid of alphabets.**

**2. Ensure each of the 25 alphabets is unique.**

**3. Omit one letter from the alphabet, usually J.**

**4. If the plaintext contains J, replace it with I.**

**5. Arrange the initial alphabets in the key square as per the key provided, followed by the remaining letters of the alphabet in order.**

**6. To encrypt the plaintext:**

**- Split it into pairs of two letters (digraphs).**

**- If there's an odd number of letters, add a Z to the last letter.**

**- Ensure pairs cannot be made with the same letter.**

**- If a letter is alone, add a bogus letter with it.**

**- Apply encryption rules:**

**- If both letters are in the same column, take the letter below each one.**

**- If both letters are in the same row, take the letter to the right of each one.**

**- If neither rule applies, form a rectangle with the two letters and take the letters on the horizontal opposite corners of the rectangle.**

**ALGORITHM**

**1. Read the key as input from the user.**

**2. Then create the key table of 5x5 grids of alphabets.**

**3. Get the plain text as an input.**

**4. Then split the plain text message into pairs of two letters(digraphs).**

**5. Pair cannot be made with same letter. Break the letter in single and add a bogus letter z to the previous letter**

**6. If both the letters are in the same column, take the letter below each one.**

**7. If both letters are in the same row, take the letter to the right of each one.**

**8. If neither of the preceding two rules are true, form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.**

**CODE**

```
import java.util.Scanner;

public class PlayfairCipher {
  public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    System.out.print("Enter keyword: ");
    String key = in.nextLine();
    System.out.print("Enter message to encrypt: ");
    String msg = in.nextLine();
```

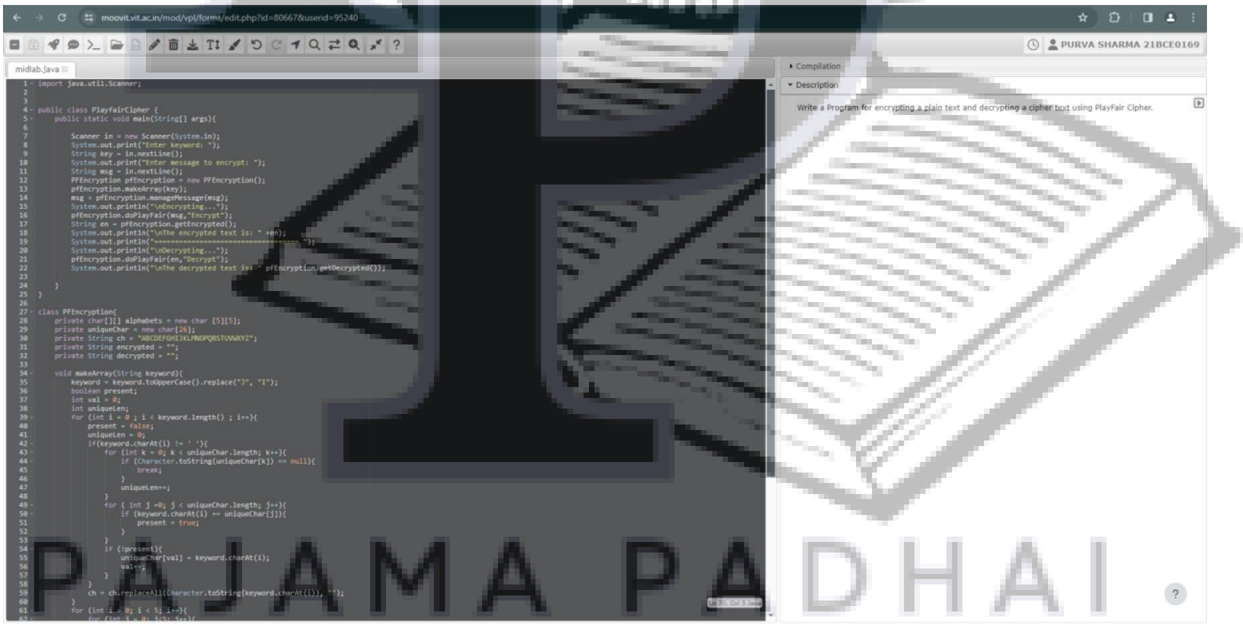
```
        PFEncryption pfEncryption = new PFEncryption();
        pfEncryption.makeArray(key);
        msg = pfEncryption.manageMessage(msg);
        System.out.println("\nEncrypting...");
        pfEncryption.doPlayFair(msg, "Encrypt");
        String en = pfEncryption.getEncrypted();
        System.out.println("\nThe encrypted text is: " + en);
        System.out.println("==============================");
        System.out.println("\nDecrypting...");
        pfEncryption.doPlayFair(en, "Decrypt");
        System.out.println("\nThe decrypted text is: " + pfEncryption.getDecrypted());
    }
}

class PFEncryption {
    private char[][] alphabets = new char[5][5];
    private char[] uniqueChar = new char[26];
    private String ch = "ABCDEFGHIKLMNOPQRSTUVWXYZ";
    private String encrypted = "";
    private String decrypted = "";

    void makeArray(String keyword) {
        keyword = keyword.toUpperCase().replace("J", "I");
        boolean present;
        int val = 0;
        int uniqueLen;
        for (int i = 0; i < keyword.length(); i++) {
            present = false;
            uniqueLen = 0;
            if (keyword.charAt(i) != ' ') {
                for (int k = 0; k < uniqueChar.length; k++) {
```

```
                if (Character.toString(uniqueChar[k]) == null) {
                    break;
                }
                uniqueLen++;
            }
            for (int j = 0; j < uniqueChar.length; j++) {
                if (keyword.charAt(i) == uniqueChar[j]) {
                    present = true;
                }
            }
            if (!present) {
                uniqueChar[val] = keyword.charAt(i);
                val++;
            }
        }
        ch = ch.replaceAll(Character.toString(keyword.charAt(i)), "");
    }
    for (int i = 0; i < ch.length(); i++) {
        uniqueChar[val] = ch.charAt(i);
        val++;
    }
    val = 0;
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 5; j++) {
            alphabets[i][j] = uniqueChar[val];
            val++;
            System.out.print(alphabets[i][j] + "\t");
        }
        System.out.println();
    }
}
```

```
String manageMessage(String msg) {
    int val = 0;
    int len = msg.length();
    String newTxt = "";
    String intermediate = "";
    while (val < len) {
        if (val + 1 < len && msg.charAt(val) == msg.charAt(val + 1)) {
            intermediate = msg.substring(val, val + 2);
            newTxt = intermediate.charAt(0) + "x" + intermediate.charAt(1);
            msg = msg.replaceFirst(intermediate, newTxt);
            len++;
        }
        val += 2;
    }
    if (msg.length() % 2 != 0) {
        msg = msg + 'x';
    }
    return msg.toUpperCase().replaceAll("J", "I").replaceAll(" ", "");
}

void doPlayFair(String msg, String tag) {
    int val = 0;
    while (val < msg.length()) {
        searchAndEncryptOrDecrypt(msg.substring(val, val + 2), tag);
        val += 2;
    }
}

void searchAndEncryptOrDecrypt(String doublyCh, String tag) {
    char ch1 = doublyCh.charAt(0);
```

```
   char ch2 = doublyCh.charAt(1);
   int row1 = 0, col1 = 0, row2 = 0, col2 = 0;
   for (int i = 0; i < 5; i++) {
      for (int j = 0; j < 5; j++) {
         if (alphabets[i][j] == ch1) {
            row1 = i;
            col1 = j;
         } else if (alphabets[i][j] == ch2) {
            row2 = i;
            col2 = j;
         }
      }
   }
   if (tag.equals("Encrypt"))
      encrypt(row1, col1, row2, col2);
   else if (tag.equals("Decrypt"))
      decrypt(row1, col1, row2, col2);
}

void encrypt(int row1, int col1, int row2, int col2) {
   if (row1 == row2) {
      col1 = col1 + 1;
      col2 = col2 + 1;
      if (col1 > 4)
         col1 = 0;
      if (col2 > 4)
         col2 = 0;
      encrypted += (Character.toString(alphabets[row1][col1]) +
            Character.toString(alphabets[row1][col2]));
   } else if (col1 == col2) {
      row1 = row1 + 1;
```

```
    row2 = row2 + 1;
    if (row1 > 4)
      row1 = 0;
    if (row2 > 4)
      row2 = 0;
    encrypted += (Character.toString(alphabets[row1][col1]) +
        Character.toString(alphabets[row2][col1]));
  } else {
    encrypted += (Character.toString(alphabets[row1][col2]) +
        Character.toString(alphabets[row2][col1]));
  }
}

void decrypt(int row1, int col1, int row2, int col2) {
  if (row1 == row2) {
    col1 = col1 - 1;
    col2 = col2 - 1;
    if (col1 < 0)
      col1 = 4;
    if (col2 < 0)
      col2 = 4;
    decrypted += (Character.toString(alphabets[row1][col1]) +
        Character.toString(alphabets[row1][col2]));
  } else if (col1 == col2) {
    row1 = row1 - 1;
    row2 = row2 - 1;
    if (row1 < 0)
      row1 = 4;
    if (row2 < 0)
      row2 = 4;
    decrypted += (Character.toString(alphabets[row1][col1]) +
                Character.toString(alphabets[row2][col1]));
        } else {
            decrypted += (Character.toString(alphabets[row1][col2]) +
                Character.toString(alphabets[row2][col1]));
        }
    }

    String getEncrypted() {
        return encrypted;
    }

    String getDecrypted() {
        return decrypted;
    }
}
```



moovit.vit.ac.in/mod/vpl/forms/edit.php?id=80667&userid=95240

PURVA SHARMA 21BCE0169

midlab.java

```
import java.util.Scanner;


public class PlayfairCipher {
    public static void main(String[] args){

        Scanner in = new Scanner(System.in);
        System.out.print("Enter keyword: ");
        String key = in.nextLine();
        System.out.print("Enter message to encrypt: ");
        String msg = in.nextLine();
        PFEncryption pfEncryption = new PFEncryption();
        pfEncryption.makeArray(key);
        msg = pfEncryption.manageMessage(msg);
        System.out.println("\nEncrypting...");
        pfEncryption.doPlayFair(msg,"Encrypt");
        String en = pfEncryption.getEncrypted();
        System.out.println("\nThe encrypted text is: " +en);
        System.out.println("==================================");
        System.out.println("\nDecrypting...");
        pfEncryption.doPlayFair(en,"Decrypt");
        System.out.println("\nThe decrypted text is: " + pfEncryption.getDecrypted());

    }
}

class PFEncryption{
    private char[][] alphabets = new char [5][5];
    private uniqueChar = new char[26];
    private String ch = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    private String encrypted = "";
    private String decrypted = "";

    void makeArray(String keyword){
        keyword = keyword.toUpperCase().replace("J", "I");
        boolean present;
        int val = 0;
        int uniqueLen;
        for (int i = 0 ; i < keyword.length() ; i++){
            present = false;
            uniqueLen = 0;
            if(keyword.charAt(i) != ' '){
                for (int k = 0; k < uniqueChar.length; k++){
                    if (Character.toString(uniqueChar[k]) == null){
                        break;
                    }
                    uniqueLen++;
                }
                for ( int j =0; j < uniqueChar.length; j++){
                    if (keyword.charAt(i) == uniqueChar[j]){
                        present = true;
                    }
                }
                if (!present){
                    uniqueChar[val] = keyword.charAt(i);
                    val++;
                }
            }
            ch = ch.replaceAll(Character.toString(keyword.charAt(i)), "");
        }
        for (int i = 0; i < 5; i++){
            for (int j = 0; j<5; j++){
```

Compilation

Description

Write a Program for encrypting a plain text and decrypting a cipher text using PlayFair Cipher.

SAMPLE INPUT OUTPUT

**Input& Output:**

**Run1:**

**Enter keyword: scope**

Enter message to encrypt: security

S C O P E

A B D F G

H I K L M

N Q R T U

V W X Y Z

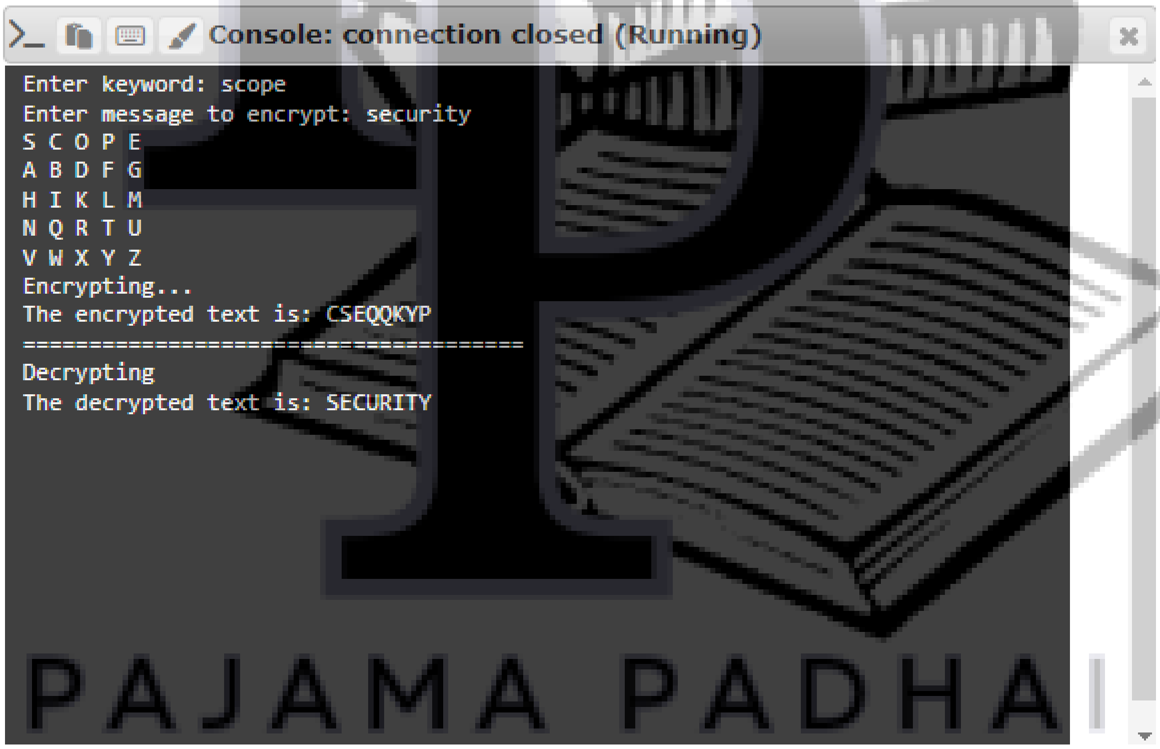
Encrypting. ..

The encrypted text is: **CSEQQKYP**

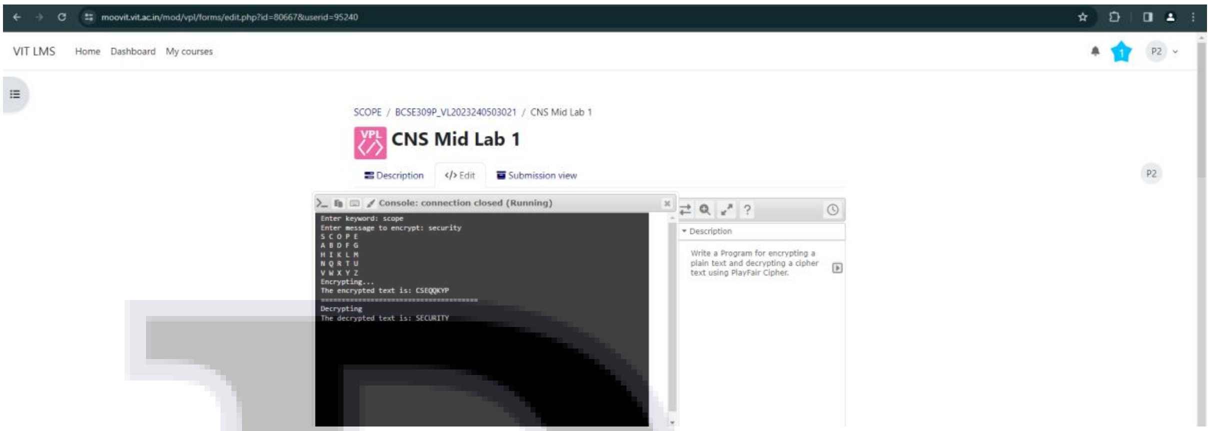
==============================

Decrypting...

The encrypted text is: **SECURITY**



```
Enter keyword: scope
Enter message to encrypt: security
S C O P E
A B D F G
H I K L M
N Q R T U
V W X Y Z
Encrypting...
The encrypted text is: CSEQQKYP
======================================
Decrypting
The decrypted text is: SECURITY
```



moovit.vit.ac.in/mod/vpl/forms/edit.php?id=80667&userid=95240
VIT LMS
Home
Dashboard
My courses
SCOPE / BCSE309P_VL2023240503021 / CNS Mid Lab 1
CNS Mid Lab 1
Description
</> Edit
Submission view
Console: connection closed (Running)
Description
P2

```
Enter keyword: scope
Enter message to encrypt: security
S C O P E
A B D F G
H I K L M
N Q R T U
V W X Y Z
Encrypting...
The encrypted text is: CSEQQKYP
===================================
Decrypting
The decrypted text is: SECURITY
```

Write a Program for encrypting a plain text and decrypting a cipher text using PlayFair Cipher.

**Run2:**

**Enter keyword: network**

**Enter message to encrypt: securitylabvit**

**N E T W O**

**R K A B C**

**D F G H I**

**L M P Q S**

**U V X Y Z**

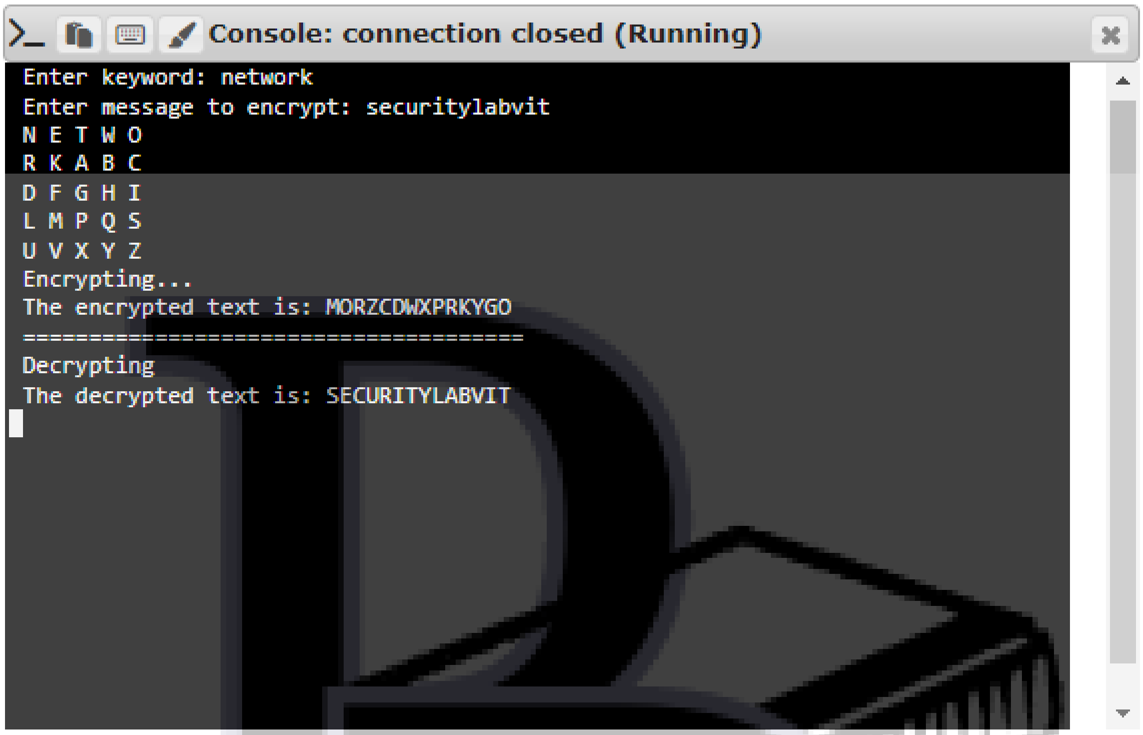
**Encrypting. ..**

**The encrypted text is: MORZCDWXPRKYGO**

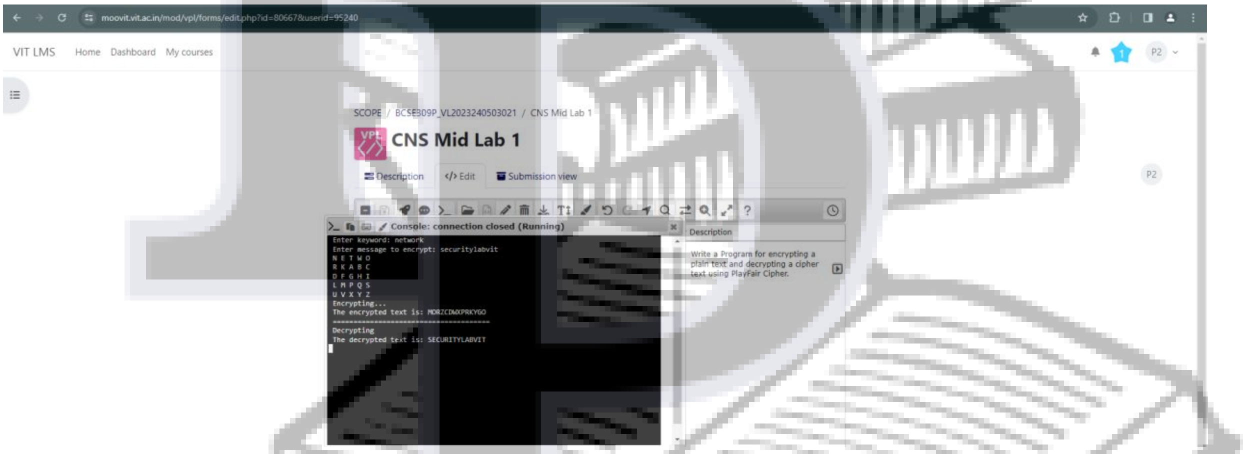
**==============================**

**Decrypting...**

**The encrypted text is: SECURITYLABVIT**



```
Enter keyword: network
Enter message to encrypt: securitylabvit
N E T W O
R K A B C
D F G H I
L M P Q S
U V X Y Z
Encrypting...
The encrypted text is: MORZCDWXPRKYGO
======================================
Decrypting
The decrypted text is: SECURITYLABVIT
```



moovit.vit.ac.in/mod/vpl/forms/edit.php?id=80667&userid=95240
VIT LMS
Home
Dashboard
My courses
P2

SCOPE / BCSE309P_VL2023240503021 / CNS Mid Lab 1

CNS Mid Lab 1

Description
</> Edit
Submission view
Console: connection closed (Running)
Description

```
Enter keyword: network
Enter message to encrypt: securitylabvit
N E T W O
R K A B C
D F G H I
L M P Q S
U V X Y Z
Encrypting...
The encrypted text is: MORZCDWXPRKYGO
======================================
Decrypting
The decrypted text is: SECURITYLABVIT
```

Write a Program for encrypting a plain text and decrypting a cipher text using PlayFair Cipher.