

Ex. No : 1(a)

Date :

Playfair Cipher

AIM:

To implement a program to encrypt a plain text and decrypt a cipher text using play fair Cipher substitution technique.

ALGORITHM:

1. To encrypt a message, one would break the message into digrams (groups of 2 letters)
2. For example, "HelloWorld" becomes "HE LL OW OR LD".
3. These digrams will be substituted using the key table.
4. Since encryption requires pairs of letters, messages with an odd number of characters usually append an uncommon letter, such as "X", to complete the final digram.
5. The two letters of the digram are considered opposite corners of a rectangle in the key table. To perform the substitution, apply the following 4 rules, in order, to each pair of letters in the plaintext:

PROGRAM:

playfairCipher.java

```
import java.awt.Point;
```

```
class playfairCipher {  
    private static char[][] charTable;  
    private static Point[] positions;  
  
    private static String prepareText(String s, boolean chgJtoI) {  
        s = s.toUpperCase().replaceAll("[^A-Z]", "");  
        return chgJtoI ? s.replace("J", "I") : s.replace("Q", "");  
    }  
  
    private static void createTbl(String key, boolean chgJtoI) {  
        charTable = new char[5][5];  
        positions = new Point[26];  
        String s = prepareText(key + "ABCDEFGHIJKLMNOPQRSTUVWXYZ",  
chgJtoI);  
        int len = s.length();  
        for (int i = 0, k = 0; i < len; i++) {  
            char c = s.charAt(i);
```

```

        if (positions[c - 'A'] == null) {
            charTable[k / 5][k % 5] = c;
            positions[c - 'A'] = new Point(k % 5, k / 5);
            k++;
        }
    }
}

private static String codec(StringBuilder txt, int dir) {
    int len = txt.length();
    for (int i = 0; i < len; i += 2) {
        char a = txt.charAt(i);
        char b = txt.charAt(i + 1);
        int row1 = positions[a - 'A'].y;
        int row2 = positions[b - 'A'].y;
        int col1 = positions[a - 'A'].x;
        int col2 = positions[b - 'A'].x;
        if (row1 == row2) {
            col1 = (col1 + dir) % 5;
            col2 = (col2 + dir) % 5;
        } else if (col1 == col2) {
            row1 = (row1 + dir) % 5;
            row2 = (row2 + dir) % 5;
        } else {
            int tmp = col1;
            col1 = col2;
            col2 = tmp;
        }
        txt.setCharAt(i, charTable[row1][col1]);
        txt.setCharAt(i + 1, charTable[row2][col2]);
    }
    return txt.toString();
}

private static String encode(String s) {
    StringBuilder sb = new StringBuilder(s);
    for (int i = 0; i < sb.length(); i += 2) {
        if (i == sb.length() - 1) {

```

```

        sb.append(sb.length() % 2 == 1 ? 'X' : "");
    } else if (sb.charAt(i) == sb.charAt(i + 1)) {
        sb.insert(i + 1, 'X');
    }
}
return codec(sb, 1);
}

private static String decode(String s) {
    return codec(new StringBuilder(s), 4);
}

public static void main(String[] args) throws java.lang.Exception {
    String key = "CSE";
    String txt = "Security Lab"; /* make sure string length is even */ /* change J
to I */
    boolean chgJtoI = true;
    createTbl(key, chgJtoI);
    String enc = encode(prepareText(txt, chgJtoI));
    System.out.println("Simulating Playfair Cipher\n ----- ");
    System.out.println("Input Message : " + txt);
    System.out.println("Encrypted Message : " + enc);
    System.out.println("Decrypted Message : " + decode(enc));
}
}

```

OUTPUT:

Simulating Playfair Cipher

Input Message : Security Lab

Encrypted Message : EABPUGYANSEZ

Decrypted Message : SECURITYLABX

RESULT:

Thus the program for playfair cipher encryption and decryption algorithm has been implemented and the output verified successfully.