

Indian Institute of Information Technology Surat



Lab Report on Network Security (CS 702) Practical

Submitted by

[RAHUL KUMAR SINGH] (UI21CS44)

Course Faculty

Dr. Reema Patel

**Department of Computer Science and Engineering
Indian Institute of Information Technology Surat
Gujarat-394190, India**

Aug-2024

Lab No: 4

Aim:

To implement a Playfair Cipher for secure file encryption and decryption using Java.

Description:

Write the menu driven program for 6X6 Playfair Cipher.

1. Encryption and Decryption of large input text. (Input: File can contain large text (including numbers, characters, symbols, spaces, etc.)

Input Key: Key may be a statement which can contain spaces, symbols, etc.

Input File Name: plaintext.txt

Encrypted File Name: cipher.txt

Decrypted Filename: recover.txt

Note: Encrypt only alphabets and digits. In the program, write logic to remove all extra characters, symbols etc. (except from alphabets and digits).

Code:

```
import javax.swing.*;
import java.awt.event.*;
import java.io.*;
import java.nio.file.*;
import java.util.*;

public class PlayfairCipherGUI {

    public static void main(String[] args) {

        JFrame frame = new JFrame("Playfair Cipher Encryption and Decryption");
        frame.setSize(600, 400);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JButton encryptButton = new JButton("Encrypt (Playfair Cipher)");
        JButton decryptButton = new JButton("Decrypt (Playfair Cipher)");

        encryptButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                performEncryption();
            }
        });

        decryptButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                performDecryption();
            }
        });
    }
}
```

```

    }
});

JPanel panel = new JPanel();
panel.add(encryptButton);
panel.add(decryptButton);

frame.add(panel);
frame.setVisible(true);
}

public static void performEncryption() {
    try {
        String inputFileName = "plaintext.txt";
        String outputFileName = "cipher.txt";
        String content = new
String(Files.readAllBytes(Paths.get(inputFileName)));
        String key = JOptionPane.showInputDialog("Enter the Key for Playfair
Cipher:");
        String cleanedContent = cleanInput(content);
        String encrypted = encryptPlayfairCipher(cleanedContent, key);
        Files.write(Paths.get(outputFileName), encrypted.getBytes());
        JOptionPane.showMessageDialog(null, "Encryption Completed! Encrypted
file saved as " + outputFileName);
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, "Error: " + ex.getMessage());
    }
}

public static void performDecryption() {
    try {
        String inputFileName = "cipher.txt";
        String outputFileName = "recover.txt";
        String content = new
String(Files.readAllBytes(Paths.get(inputFileName)));
        String key = JOptionPane.showInputDialog("Enter the Key for Decryption
(Playfair Cipher):");
        String decrypted = decryptPlayfairCipher(content, key);
        Files.write(Paths.get(outputFileName), decrypted.getBytes());
        JOptionPane.showMessageDialog(null, "Decryption Completed! Decrypted
file saved as " + outputFileName);
    } catch (Exception ex) {

```

```

        JOptionPane.showMessageDialog(null, "Error: " + ex.getMessage());
    }
}

public static String cleanInput(String text) {
    return text.replaceAll("[^A-Za-z0-9]", "").toUpperCase();
}

public static String encryptPlayfairCipher(String text, String key) {
    char[][] matrix = generatePlayfairMatrix(key);
    StringBuilder result = new StringBuilder();
    for (int i = 0; i < text.length(); i += 2) {
        char a = text.charAt(i);
        char b = (i + 1 < text.length()) ? text.charAt(i + 1) : 'X';
        result.append(encryptPair(a, b, matrix));
    }
    return result.toString();
}

public static String decryptPlayfairCipher(String text, String key) {
    char[][] matrix = generatePlayfairMatrix(key);
    StringBuilder result = new StringBuilder();
    for (int i = 0; i < text.length(); i += 2) {
        char a = text.charAt(i);
        char b = (i + 1 < text.length()) ? text.charAt(i + 1) : 'X';
        result.append(decryptPair(a, b, matrix));
    }
    return result.toString();
}

public static char[][] generatePlayfairMatrix(String key) {
    StringBuilder sb = new StringBuilder(key);
    Set<Character> usedChars = new LinkedHashSet<>();
    for (char c : sb.toString().toUpperCase().toCharArray()) {
        if (Character.isLetterOrDigit(c) && !usedChars.contains(c)) {
            usedChars.add(c);
        }
    }
    for (char c = 'A'; c <= 'Z'; c++) {
        if (!usedChars.contains(c)) {
            usedChars.add(c);
        }
    }
}

```

```

    }

    for (char c = '0'; c <= '9'; c++) {
        if (!usedChars.contains(c)) {
            usedChars.add(c);
        }
    }

    char[][] matrix = new char[6][6];
    int index = 0;
    for (char c : usedChars) {
        matrix[index / 6][index % 6] = c;
        System.out.print(c);
        index++;
    }
    System.out.println();
    return matrix;
}

public static String encryptPair(char a, char b, char[][] matrix) {
    int[] pos1 = findPosition(a, matrix);
    int[] pos2 = findPosition(b, matrix);
    if (pos1[0] == pos2[0]) {
        return "" + matrix[pos1[0]][(pos1[1] + 1) % 6] +
matrix[pos2[0]][(pos2[1] + 1) % 6];
    } else if (pos1[1] == pos2[1]) {
        return "" + matrix[(pos1[0] + 1) % 6][pos1[1]] + matrix[(pos2[0] + 1) %
6][pos2[1]];
    } else {
        return "" + matrix[pos1[0]][pos2[1]] + matrix[pos2[0]][pos1[1]];
    }
}

public static String decryptPair(char a, char b, char[][] matrix) {
    int[] pos1 = findPosition(a, matrix);
    int[] pos2 = findPosition(b, matrix);
    if (pos1[0] == pos2[0]) {
        return "" + matrix[pos1[0]][(pos1[1] + 5) % 6] +
matrix[pos2[0]][(pos2[1] + 5) % 6];
    } else if (pos1[1] == pos2[1]) {
        return "" + matrix[(pos1[0] + 5) % 6][pos1[1]] + matrix[(pos2[0] + 5) %
6][pos2[1]];
    }
}

```

```

    } else {
        return "" + matrix[pos1[0]][pos2[1]] + matrix[pos2[0]][pos1[1]];
    }
}

public static int[] findPosition(char c, char[][] matrix) {
    for (int i = 0; i < 6; i++) {
        for (int j = 0; j < 6; j++) {
            if (matrix[i][j] == c) {
                return new int[] { i, j };
            }
        }
    }
    throw new IllegalArgumentException("Character not found in matrix.");
}
}

```

Output:

Plaintext.txt

In the vast and ancient world, where time flows like a river and the stars paint the sky, there existed a land of great beauty and mystery. This land, rich with rolling hills, deep forests, and winding rivers, was home to a people whose lives were intertwined with the natural rhythms of the earth. They cultivated the land, worshipped the sun and the moon, and passed down stories from one generation to the next. These stories spoke of heroes and gods, of love and betrayal, of battles won and lost. But among all these tales, one legend stood out. It was the tale of a hidden treasure, buried deep within the heart of the land. A treasure so vast and powerful that it could change the course of history. Many had searched for it, but none had found it. Some said it was guarded by a fierce dragon, others believed it was cursed. But despite the dangers, the lure of the treasure was too great. And so, the search continued, driven by hope, greed, and the desire for glory. As the years passed, the legend grew, taking on a life of its own. But in the end, it was not the treasure that mattered, but the journey itself.

Cipher.txt [Key:]

JGP1HWCQQDMFFIAK2IQZLS1BVELQ1QJN2A01YQEJIOBQGWLFIMZELTUBQTQIQMUP10QS7P1LQLLWJTU1ABEFIC13NQLDQALFQSZFIMT
7YQ1SXP1KQEBMFQJ305QP1SLOOJGP3JEORA1HQC2QLTUQCMF5QMFJGJPGWLQQYCQE1I1S13QL1RHA5E1QOEJWHQYLQAQMULQQZJG1A5Q
P1P12IDQPSBEPLZS1GYCDUELAI5U1PEL7KR2QMW3Q1MZELEBMFYESTEGQQ1AP10Q0UFIMZELK12KFI3TCQQ0FFEYKUS1QJQ0BU1K2KHI
2ILQDQKEMU1SELI2ZRP10Q0QS1QJ0QTQCSL132LQ1LQCMFKHCT2C01WHFIFC1QQBWC013CDQR10QYEIFMFO1TUFRQDK1GIBE1RELQ0QD
OLYCI2OLIHMFU11C1PUMQ5ITUELQDOL2C3EMAA1MUQLCQPSLAPSQAFFLLQVMQEGMUELELBQ51DUELEBMFDQQLCQPSOQHVCQDMF5H5A
UBR2P1DQMQR2FDE3GI1QELKCP5Q02CEGTULSZKFIV0BFQ0BQ301AC2QJRDPUK2I2E3F32SMFMQYCI1QCMAMQ5IPKQFTB1ACXB3QASB
1AQBKHK2P1LQRCLOQAWHAMQZCQFSST1AFRZMQ0QGQ1P11AFIIHSTP1L0PSL1DUELUSAITPQL5ITU11JPAIQDMFYCP10QAISB032KQMU0
1ABTGW2ICXE1QHJPLLFBMF11A0QJQ2ALSJHLSWCTUELWOBQTQCQ0MZEL0LIHMFJPA5QDMJGI2KBENAL1ANUTEYJFPUJGP1LLMFMQ5I
UK1SP11QQLCQPS1QE3ZTDQ01QLFCUP1LI2SUJOWMQQ02B

Recover.txt

INTHEVASTANDANCIENTWORLDWHERE TIME FLOWS LIKE A RIVER AND THE STARS PAINT THE SKY THERE EXISTED A LAND OF GREAT BEAUTY AND MYSTERY THIS LAND RICH WITH ROLLING HILLS DEEP FORESTS AND WINDING RIVERS WAS HOME TO A PEOPLE WHOSE LIVES WERE INTERTWINED WITH THE NATURAL RHYTHMS OF THE EARTH THEY CULTIVATED THE LAND AND WORSHIPPED THE SUN AND THE MOON AND PASSED DOWN STORIES FROM ONE

GENERATION TO THE NEXT THESE STORIES SPOKE OF HEROES AND GODS OF LOVE AND BETRAYAL OF BATTLES WON AND LOST BUT AMONG ALL THESE TALES ONE LEGEND STOOD OUT IT WAS THE TALE OF A HIDDEN TREASURE BURIED DEEP WITHIN THE HEART OF THE LAND A TREASURE SO VAST AND POWERFUL THAT IT COULD CHANGE THE COURSE OF HISTORY MANY HAD SEARCHED FOR IT BUT NONE HAD FOUND IT SOME SAID IT WAS GUARDED BY A FIERCE DRAGON OTHERS BELIEVED IT WAS CURSED BUT DESPITE THE DANGER THE LURE OF THE TREASURE WAS TOO GREAT AND SO THE SEARCH CONTINUED DRIVEN BY HOPE GREED AND THE DESIRE FOR GLORY AS THE YEARS PASSED THE LEGEND GREW TAKING ON A LIFE OF ITS OWN BUT IN THE END IT WAS NOT THE TREASURE THAT MATTERED BUT THE JOURNEY ITSELF

Conclusion:

- The Playfair Cipher offers basic security through letter pairing but is vulnerable to frequency analysis.
- It is efficient for manual encryption but impractical for large-scale digital use.
- Modern encryption methods like AES provide superior security and scalability.
- Playfair Cipher can only be used for educational purposes like in this assignment.