

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: 3

Aim:

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

Description:

Write the Menu Driven Program for Rail Fence Cipher.

1. Encryption
2. Decryption

Input: File - Large Plaintext (file.txt), Key Value

Output: File - Encoded Text (Cipher.txt)

Input File Name: Plaintext.txt

Encrypted File Name: Cipher.txt

Decrypted File Name: Recover.txt

Code:

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

public class RailFenceCipherGUI {

    public static void main(String[] args) {
        JFrame frame = new JFrame("Rail Fence Cipher Encryption and Decryption");
        frame.setSize(1600, 900);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JButton encryptButton = new JButton("Encrypt (Rail Fence Cipher)");
        JButton decryptButton = new JButton("Decrypt (Rail Fence 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)));
            int key = Integer.parseInt(JOptionPane.showInputDialog("Enter the Key
for Rail Fence Cipher:"));
            String encrypted = encryptRailFenceCipher(content, 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)));
            int key = Integer.parseInt(JOptionPane.showInputDialog("Enter the Key
for Decryption (Rail Fence Cipher):"));
            String decrypted = decryptRailFenceCipher(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 encryptRailFenceCipher(String text, int key) {
    char[][] rail = new char[key][text.length()];
    boolean dirDown = false;
    int row = 0, col = 0;

    for (int i = 0; i < text.length(); i++) {
        if (row == 0 || row == key - 1)
            dirDown = !dirDown;

        rail[row][col++] = text.charAt(i);

        if (dirDown)
            row++;
        else
            row--;
    }

    StringBuilder result = new StringBuilder();
    for (int i = 0; i < key; i++) {
        for (int j = 0; j < text.length(); j++) {
            if (rail[i][j] != 0)
                result.append(rail[i][j]);
        }
    }
    return result.toString();
}

public static String decryptRailFenceCipher(String cipher, int key) {
    char[][] rail = new char[key][cipher.length()];
    boolean dirDown = false;
    int row = 0, col = 0;

    for (int i = 0; i < cipher.length(); i++) {
        if (row == 0 || row == key - 1)
            dirDown = !dirDown;

        rail[row][col++] = '*';

        if (dirDown)
            row++;
        else

```

```

        row--;
    }

    int index = 0;
    for (int i = 0; i < key; i++) {
        for (int j = 0; j < cipher.length(); j++) {
            if (rail[i][j] == '*' && index < cipher.length()) {
                rail[i][j] = cipher.charAt(index++);
            }
        }
    }

    StringBuilder result = new StringBuilder();
    dirDown = false;
    row = 0;
    col = 0;

    for (int i = 0; i < cipher.length(); i++) {
        if (row == 0 || row == key - 1)
            dirDown = !dirDown;

        if (rail[row][col] != 0)
            result.append(rail[row][col++]);

        if (dirDown)
            row++;
        else
            row--;
    }

    return result.toString();
}

```

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

```
Itoh,
i,n etmcotoet ff.tolet do so ytce grbtAh shjnn lwtey do
mracs idetvsihhs uwr hdwgnx.eoo o t ato aer, h.An cueo ennesb oh uh ft rac y h stego.Ba ta o ewfs ktnf y
hldwimoi w t yl sde n ee kf l bsBelsdt u ned ap lsdfa ema a
esrtdohoeehn de. nnnuwn teutio dssha ds, le no lw
yoet,hn d ernTo ,o,aouse osbi nt otdr art oid
ras a eoas e
n yt,li wt oe hrhcrelnt elgntdwiedgha ed
hfhidiamesna hphsvltltesdoefauhhartwi uhhcuhsdef sweene tt cv
adrhdekao ttrmtnenlmiaaer raenihpn eretr Tvnp ostoteesedeat h,nuh ertelese
coi hba di, dtgrt .eoih eoeega ii ua e adik rheae rat arspsenh t aapnoso ihs ro yldat ettariia aar
hcsye d.irenbt.ieurs hnro,sl setlsn tsttyv ,terst ayylhgf iaeo ielh.tleunarmotesogaaenm og. tewresvf a
tnd, ttaroei prleaAttdpdigysn it ,hatu ad e edtt. no,vwohiw aehe ds,pion ee nrsaozne sh d thu
utneoa tfi ucgl Bss awn i eer ead,fitdeeebi.snweavptxe u srirse pwntnr tde er osisddt nlelIain p
trolaghrMfio wgeaidue,es d,n,,eerap we hn rr tftahr ianitbaTioletr,l trauer hteadsf
ttr n ewnga
twdedeo es hety o uda reetd hue oud r or ge fee
te,sl erisehlsseetathn
oabo
def t .rnsdv trsegfsro
de
```

Recover.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.

Conclusion:

- Limited Security: Its security is weak, offering around 0.01-0.015% protection compared to modern encryption methods.
- Predictable Patterns: The zigzag pattern is predictable, making the cipher vulnerable to brute-force and pattern-based attacks.
- Key Size Restrictions: The number of rails (key) must be between 2 and the message length, limiting its flexibility.
- Ineffective for Long Messages: The cipher becomes less secure and manageable as the message length increases.