ASSIGNMENT – 4

Information Security – Lab

Code:20CP304P

Name: Smit Sutaria

Roll no: 21BCP142

DIVISION 3 (Group 5)

COMPUTER ENGINEERING

School of Technology, Pandit Deendayal Energy University

Experiment No : 4

Aim : To understand rail-fence cipher.

Introduction :

Encryption

In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.

In the rail fence cipher, the plain-text is written downwards and diagonally on successive rails of an imaginary fence.

When we reach the bottom rail, we traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus the alphabets of the message are written in a zig-zag manner.

After each alphabet has been written, the individual rows are combined to obtain the cipher-text.

Decryption

As we’ve seen earlier, the number of columns in rail fence cipher remains equal to the length of plain-text message. And the key corresponds to the number of rails.

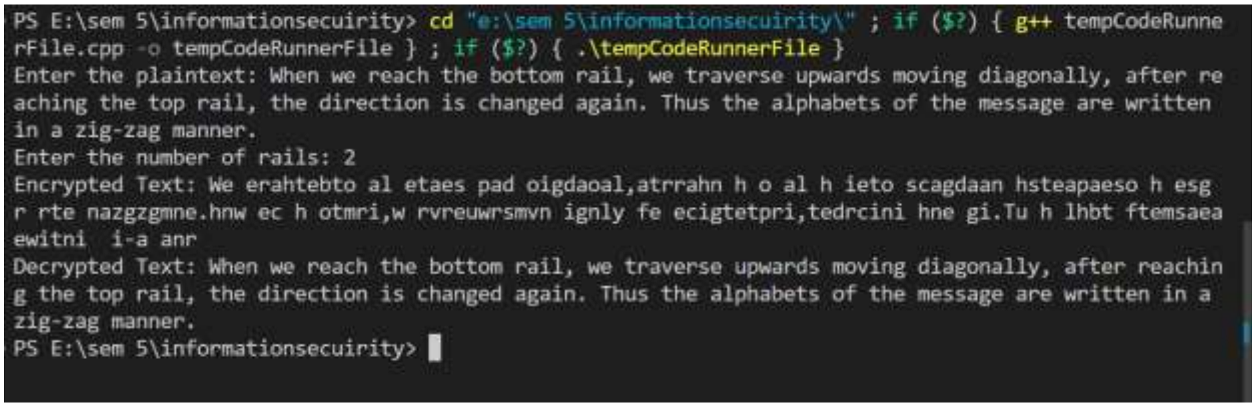
Hence, rail matrix can be constructed accordingly. Once we’ve got the matrix we can figure-out the spots where texts should be placed (using the same way of moving diagonally up and down alternatively).

Then, we fill the cipher-text row wise. After filling it, we traverse the matrix in zig-zag manner to obtain the original text.

Program (Source Code):

```cpp
#include <iostream>
#include <string>

using namespace std;

string encryptRailFence(string plaintext, int rails) {
    string encryptedText;
    string railFence[rails];

    for (int i = 0; i < rails; ++i) {
        railFence[i] = "";
    }

    int rail = 0;
    bool directionDown = false;

    for (char c : plaintext) {
        railFence[rail] += c;

        if (rail == 0 || rail == rails - 1) {
            directionDown = !directionDown;
        }

        if (directionDown) {
            rail++;
        } else {
            rail--;
        }
    }

    for (int i = 0; i < rails; ++i) {
        encryptedText += railFence[i];
    }

    return encryptedText;
}

string decryptRailFence(string ciphertext, int rails) {
    string decryptedText;
    string railFence[rails];

    for (int i = 0; i < rails; ++i) {
        railFence[i] = "";
    }

    int rail = 0;
    bool directionDown = false;

    for (int i = 0; i < ciphertext.length(); ++i) {
        railFence[rail] += '*';

        if (rail == 0 || rail == rails - 1) {
            directionDown = !directionDown;
        }

        if (directionDown) {
            rail++;
        } else {
            rail--;
        }
    }

    int index = 0;
    for (int i = 0; i < rails; ++i) {
        for (int j = 0; j < railFence[i].length(); ++j) {
            railFence[i][j] = ciphertext[index++];
        }
    }

    rail = 0;
    directionDown = false;

    for (int i = 0; i < ciphertext.length(); ++i) {
        decryptedText += railFence[rail][0];
        railFence[rail] = railFence[rail].substr(1);

        if (rail == 0 || rail == rails - 1) {
            directionDown = !directionDown;
        }

        if (directionDown) {
            rail++;
        } else {
            rail--;
        }
    }
    return decryptedText;
}

int main() {
    string plaintext, encryptedText, decryptedText;
    int rails;

    cout << "Enter the plaintext: ";
    getline(cin, plaintext);

    cout << "Enter the number of rails: ";
    cin >> rails;

    encryptedText = encryptRailFence(plaintext, rails);
    decryptedText = decryptRailFence(encryptedText, rails);

    cout << "Encrypted Text: " << encryptedText << endl;
    cout << "Decrypted Text: " << decryptedText << endl;

    return 0;
}
```

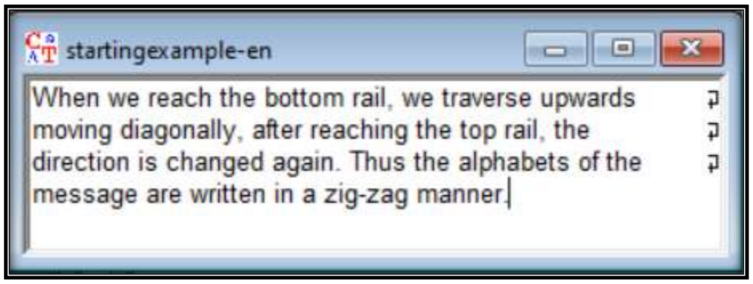
Output (Program):



```
PS E:\sem 5\informationsecuirity> cd "e:\sem 5\informationsecuirity\" ; if ($?) { g++ tempCodeRunne
rFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter the plaintext: When we reach the bottom rail, we traverse upwards moving diagonally, after re
aching the top rail, the direction is changed again. Thus the alphabets of the message are written
in a zig-zag manner.
Enter the number of rails: 2
Encrypted Text: We erahtebto al etaes pad oigdaoal,atrrahn h o al h ieto scagdaan hsteapaeso h esg
r rte nazgzgmne.hnw ec h otmri,w rvreuwrsmvn ignly fe ecigtetpri,tedrcini hne gi.Tu h lhbt ftemsaea
ewitni  i-a anr
Decrypted Text: When we reach the bottom rail, we traverse upwards moving diagonally, after reachin
g the top rail, the direction is changed again. Thus the alphabets of the message are written in a
zig-zag manner.
PS E:\sem 5\informationsecuirity> 
```

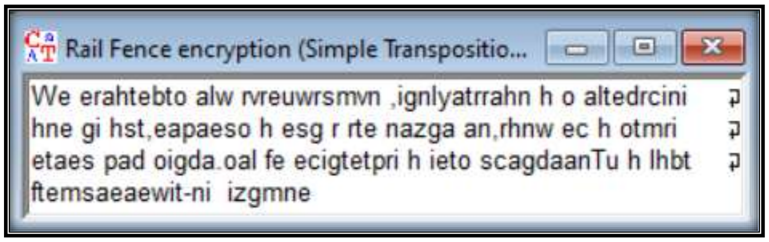
Output (Cryptool):

Plain-Text:



```
When we reach the bottom rail, we traverse upwards
moving diagonally, after reaching the top rail, the
direction is changed again. Thus the alphabets of the
message are written in a zig-zag manner.
```

Cipher-Text:



```
We erahtebto alw rvreuwrsmvn ,ignlyatrrahn h o altedrcini
hne gi hst,eapaeso h esg r rte nazga an,rhnw ec h otmri
etaes pad oigda.oal fe ecigtetpri h ieto scagdaanTu h lhbt
ftemsaeaewit-ni  izgmne
```

Cryptanalysis : The cipher's key is N, the number of rails. If N is known, the ciphertext can be decrypted by using the above algorithm. Values of N equal to or greater than L, the length of the ciphertext, are not usable, since then the ciphertext is the same as the plaintext. Therefore the number of usable keys is low, allowing the brute-force attack of trying all possible keys. As a result, the rail-fence cipher is considered weak

Applications :

The rail fence cipher, also known as the zigzag cipher, is a simple transposition cipher that is used to encrypt a message by rearranging its letters in a zigzag pattern. It is called the "rail fence" cipher because when the letters are written in a zigzag pattern, they resemble the rails of a fence. The rail fence cipher is not very secure and is primarily used for educational or entertainment purposes rather than for secure communication. Here are some common applications of the rail fence cipher:

1. Education and Learning: The rail fence cipher is often used as a teaching tool to introduce students to basic encryption concepts. It provides a hands-on way to understand transposition ciphers and encryption techniques.

2. Puzzles and Games: The rail fence cipher can be used in puzzles, games, and brain teasers. It is sometimes included in escape room challenges or puzzle books, where participants need to decrypt a message to proceed.

3. Pen and Paper Encryption: In situations where a simple form of encryption is needed for fun or novelty, such as writing secret messages between friends or family members, the rail fence cipher can be used. It's easy to encrypt and decrypt by hand.

4. Cryptography History: The rail fence cipher is an interesting historical cipher that dates back to ancient times. It can be used to teach the history of cryptography and its development over the centuries.

5. Illustrating Encryption Principles: The rail fence cipher can be used to illustrate the concept of encryption and decryption, even though it's not suitable for securing sensitive information. Teachers and cryptography enthusiasts may use it as a visual aid to explain how basic encryption works.

6. Creative Writing: Writers and authors sometimes incorporate the rail fence cipher into their works for artistic or narrative purposes. It can add an element of mystery or intrigue to a story when characters need to decipher hidden messages.

7. Online Challenges and Competitions: The rail fence cipher, along with other simple ciphers, may be used in online challenges or competitions to test participants' code-breaking skills and knowledge of cryptography.

It's important to note that the rail fence cipher is not suitable for securing sensitive information because it is relatively easy to decrypt through various methods, including brute force and frequency analysis. For secure communication, more robust encryption techniques should be used, such as modern symmetric or asymmetric encryption algorithms.

References :

1. **Ref-1:** https://www.geeksforgeeks.org/rail-fence-cipher-encryption-decryption/

2. **Ref-2:** https://en.wikipedia.org/wiki/Rail_fence_cipher