Final Year B. Tech. (CSE) – I: 2022-23

4CS451: Cryptography and Network Security Lab

Assignment No. 5

PRN: 2019BTECS00077 **Batch: B7**

Full name: Biradar Avinash Vishnu

<u>Title</u>: Implementation of Rail Fence and Columnar Transposition cipher algorithms.

<u>Objective</u>: write a program to encrypt the plain text and decrypt the cipher text using Rail Fence and Columnar Transposition cipher algorithms.

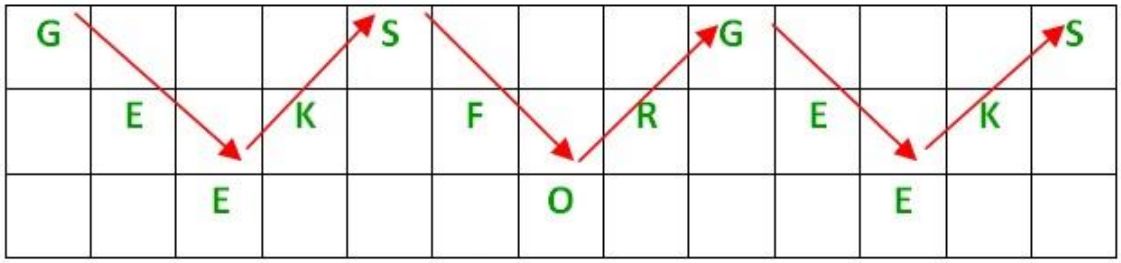
<u>Introduction & Theory</u>:

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

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.

For example, if the message is “GeeksforGeeks” and the number of rails = 3 then cipher is prepared as:



Columnar Transposition:

Columnar Transposition involves writing the plaintext out in rows, and then reading the ciphertext off in columns one by one

RailFence Code:

```
#include<bits/stdc++.h>
using namespace std;

// Capitalize the string
void capitalize(string &str){
    for(char &c:str){
      if(c>=97 && c<=122)
        c-=32;
    }
}

string encrypt(string &plaintext,int depth){

    int n=plaintext.size();
    vector<vector<char>> mat(depth,vector<char>(n,'*'));

    bool dirDown= false;
    int row = 0, col = 0;

    for (int i=0; i<n; i++)
    {
        if (row == 0 || row == depth-1)
            dirDown = !dirDown;
        mat[row][col++] = plaintext[i];
        dirDown?row++ : row--;
    }

    string ans;
    for (int i=0; i < depth; i++)
        for (int j=0; j <n; j++)
            if (mat[i][j]!='*')
                ans.push_back(mat[i][j]);
    return ans;
}

string decrypt(string &cypherText,int depth){

    int n=cypherText.size();
    vector<vector<char>> mat(depth,vector<char>(n,'#'));
```

```
    bool dirDown=false;

    int row=0, col=0;
    for (int i=0;i<n; i++)
    {
        if (row == 0 || row==depth-1)
          dirDown=!dirDown;
        mat[row][col++] = '*';
        dirDown?row++ : row--;
    }

    int index = 0;
    for (int i=0; i<depth; i++){
        for (int j=0; j<n; j++)
            if (mat[i][j] == '*' && index<n)
                mat[i][j] =cypherText[index++];
    }

    string ans;
    row = 0, col = 0;
    for (int i=0; i<n; i++)
    {
        if (row == 0)
            dirDown = true;
        if (row == depth-1)
            dirDown= false;

        if (mat[row][col] != '*')
            ans.push_back(mat[row][col++]);

        dirDown?row++: row--;
    }
    return ans;
}

int main(){

    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);


    string plainText;
    getline(cin,plainText);
    // cout<<plainText<<endl;
    capitalize(plainText);
```

```
    int depth;
    cin>>depth;

    string cypherText=encrypt(plainText,depth);

    cout<<"Cypher Text:\n"<<cypherText<<"\n\n";

    plainText=decrypt(cypherText,depth);

    cout<<"Plain Text:\n"<<plainText<<endl;

    return  0;
}
```

RailFence Result:

File input output:

input.txt

```
Meet me after the party
2
```

output.txt

```
Cypher Text:
ME EATRTEPRYETM FE H AT

Plain Text:
MEET ME AFTER THE PARTY
```

Console Input Output:

```
PS E:\College\Final year\C&NS\practic
ilfence } ; if ($?) { .\Transpo_Rail
this is rail fence cipher
5
Cypher Text:
TRCRH ANEE ISIE HSILFCP  I

Plain Text:
THIS IS RAIL FENCE CIPHER
```

Columnar Code:

```
#include<bits/stdc++.h>
using namespace std;

// Capitalize the string
void capitalize(string &str){
    for(char &c:str){
      if(c>=97 && c<=122)
        c-=32;
    }
}

string encrypt(int n,int m,string &plaintext,vector<int> &key){

 vector<vector<char>> mat(n,vector<char>(m));

    int k=0;
    for(int i=0;i<n;i++){
      for(int j=0;j<m;j++){
        if(k>=plaintext.size())
          mat[i][j]=32;
        else
          mat[i][j]=plaintext[k++];
      }
    }

    // Matrix
    for(int i=0;i<n;i++){
      for(int j=0;j<m;j++){
```

```
        if(mat[i][j]!=32)
            cout<<mat[i][j]<<" ";
      }
      cout<<endl;
    }

    string cypherText="";
    for(int i=0;i<m;i++){
      for(int j=0;j<n;j++){
        if(mat[j][key[i]-1]!=32)
            cypherText.push_back(mat[j][key[i]-1]);
      }
    }
    return cypherText;
}

string decrypt(int m,string &cypherText,vector<int> key){

 int n=(cypherText.size()/m)+1;

 vector<vector<char>> mat(n,vector<char>(m));

    int k=0;
    string plainText="";
    for(int i=0;i<m;i++){
      for(int j=0;j<n;j++){
            if(j==n-1 && key[i]-1>=(cypherText.size()%m))
              mat[j][key[i]-1]=32;
            else
              mat[j][key[i]-1]=cypherText[k++];
      }
    }

     // Matrix
    for(int i=0;i<n;i++){
      for(int j=0;j<m;j++){
        if(mat[i][j]!=32)
            cout<<mat[i][j]<<" ";
      }
      cout<<endl;
    }

    for(int i=0;i<n;i++){
      for(int j=0;j<m;j++){
          if(mat[i][j]!=32)
            plainText.push_back(mat[i][j]);
```

```
      }
    }

    return plainText;
}

int main(){

    // freopen("input.txt", "r", stdin);
    // freopen("output.txt", "w", stdout);

    string plainText;
    getline(cin,plainText);

    int m;
    cin>>m;

    vector<int> key(m);

    for(int i=0;i<m;i++)
       cin>>key[i];


    capitalize(plainText);
    string tmp="";

    for(char c:plainText){
      if(c!=32)
        tmp.push_back(c);
    }

    int n=(tmp.size()/m)+1;

    string Cyphertext=encrypt(n,m,tmp,key);

    cout<<"CypherText:"<<Cyphertext<<"\n\n";

    plainText=decrypt(m,Cyphertext,key);
    cout<<"plainText:"<<plainText<<"\n";

    return  0;
}
```

Columnar Result:
File input output:

input.txt

```
Meet Me after the party
6
2 5 3 1 6 4
```

output.txt

```
M E E T M E
A F T E R T
H E P A R T
Y
CypherText:EFEMRRETPMAHYETTTEA

M E E T M E
A F T E R T
H E P A R T
Y
plainText:MEETMEAFTERTHEPARTY
```

Console Input Output:

```
PS E:\College\Final year\C&NS\practical> cd "e:\College
nar } ; if ($?) { .\Transo_columnar }
this is columnar encryption and decryption
9
6 4 1 9 8 3 7 5 2
T H I S I S C O L
U M N A R E N C R
Y P T I O N A N D
D E C R Y P T I O
N
CypherText:SENPSAIRTUYDNLRDOOCNIINTCCNATIROYHMPE

T H I S I S C O L
U M N A R E N C R
Y P T I O N A N D
D E C R Y P T I O
N
plainText:THISISCOLUMNARENCRYPTIONANDDECRYPTION
```