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

4CS451: Cryptography and Network Security Lab

Assignment No. 3

PRN: 2019BTECS00077 **Batch: B7**

Full name: Biradar Avinash Vishnu

<u>Title</u>: Implementation of Playfair cipher algorithm.

<u>Objective</u>: write a program to encrypt the plain text and decrypt the cipher text using Playfair cipher algorithm.

<u>Introduction & Theory</u>:

The Playfair Cipher Encryption Algorithm:

The Algorithm consists of 2 steps:

Generate the key matrix Square (5×5):

- The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I.

- The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.
-

Encrypt the plain text:

The plaintext is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter.

Rules for Encryption:

- **If both the letters are in the same column**: Take the letter below each one (going back to the top if at the bottom).

- **If both the letters are in the same row**: Take the letter to the right of each one (going back to the leftmost if at the rightmost position).

- **If neither of the above rules is true**: Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

Code:

```
#include<bits/stdc++.h>
using namespace std;
map<char,pair<int,int>> charPos;

// Capitalize the character
void capitalize(string &str){

    for(char &c:str){
      if(c>=97 && c<=122)
        c-=32;
    }
}

// add char pairs to the set
void addToSet(vector<pair<char,char>> &v,char a,char b){
    v.push_back({a,b});
}

// build matrix
void buildMatrix(vector<vector<char>> &mat,string key){

    vector<bool>vis(26,false);

    int i=0,j=0;
    for(int k=0;k<key.length();k++){

        if(!vis[key[k]-65]){
            mat[i][j]=key[k];
            if(key[k]=='I'||key[k]=='J'){
                vis['I'-65]=true;
                vis['J'-65]=true;
            }
           else vis[key[k]-65]=true;
           j++;
           if(j==5){
              j%=5;
             i++;
          }
        }
```

```
    }
    for(int k=0;k<26;k++){

        if(!vis[k]){
            if(k+'A'=='I'||k+'A'=='J'){
                vis['I'-65]=true;
                vis['J'-65]=true;
            }
            mat[i][j]=k+'A';
            j++;
           if(j==5){
             j%=5;
             i++;
            }
        }

    }
    // cout<<"\n \nMatrix:\n";
    for(int i=0;i<5;i++){
        for(int j=0;j<5;j++){
            cout<<mat[i][j]<<" ";
        }
        cout<<'\n';
    }
}

void encrypt(vector<vector<char>> &mat,vector<pair<char,char>>
&pairSet){

    // storing position of chars in the matrix
    for(int i=0;i<5;i++){
      for(int j=0;j<5;j++)
        if(mat[i][j]=='I' || mat[i][j]=='J')
          charPos['J']=charPos['I']=make_pair(i,j);
        else
          charPos[mat[i][j]]=make_pair(i,j);
    }

    // print
    cout<<"\n";
    // cout<<"Plain Text:";
    for(auto it:pairSet)
      cout<<it.first<<it.second<<" ";
    cout<<endl;

    // cout<<"Cypher Text: ";
```

```
    for(auto &it:pairSet){
       int i1=charPos[it.first].first;
       int j1=charPos[it.first].second;
       int i2=charPos[it.second].first;
       int j2=charPos[it.second].second;

       if(i1==i2){
        it.first=mat[i1][(j1+1)%5];
        it.second=mat[i1][(j2+1)%5];
        cout<<mat[i1][(j1+1)%5]<<mat[i1][(j2+1)%5]<<" ";
       }
       else if(j1==j2){
        it.first=mat[(i1+1)%5][j1];
        it.second=mat[(i2+1)%5][j2];
        cout<<mat[(i1+1)%5][j1]<<mat[(i2+1)%5][j2]<<" ";
       }
       else{
        it.first=mat[i1][j2];
        it.second=mat[i2][j1];
        cout<<mat[i1][j2]<<mat[i2][j1]<<" ";
       }

    }
    cout<<"\n\n";
}

void decrypt(vector<vector<char>> &mat,vector<pair<char,char>>
&pairSet){

    // print
    // for(auto it:charPos)
    //   cout<<it.first<<" "<<it.second.first<<"
"<<it.second.second<<endl;
    // cout<<endl;

    // cout<<"Plain Text: ";
    for(auto it:pairSet){
       int i1=charPos[it.first].first;
       int j1=charPos[it.first].second;
       int i2=charPos[it.second].first;
       int j2=charPos[it.second].second;

      //  cout<<i1<<j1<<" "<<i2<<j2<<endl;

       if(i1==i2){
        cout<<mat[i1][((j1-1)+5)%5]<<mat[i1][((j2-1)+5)%5]<<" ";
```

```
        }
        else if(j1==j2){
         cout<<mat[((i1-1)+5)%5][j1]<<mat[((i2-1)+5)%5][j2]<<" ";
        }
        else{
         cout<<mat[i1][j2]<<mat[i2][j1]<<" ";
        }

    }
}

int main(){

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


    string tmp,key,plainText;
    vector<pair<char,char>> pairSet;

    getline(cin,tmp);

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

    // cout<<plainText<<endl;
    capitalize(plainText);

    getline(cin,key);

    capitalize(key);

    // cout<<key<<endl;

    int n=plainText.size();
    for(int i=0;i<n;){
        char a=plainText[i];
        if(i==n-1 || plainText[i+1]==a){
          addToSet(pairSet,a,'X');  // added if x as dummy for same
chars
          i++;
        }
        else{
          addToSet(pairSet,a,plainText[i+1]);
```

```
        i+=2;
      }
    }
  // breaking the plain text into 2chars
  // for(auto it:pairSet)
  //   cout<<it.first<<it.second<<" ";

  // build matrix
  vector<vector<char>> mat(5,vector<char>(5));
  buildMatrix(mat,key);

  // Encrypt the text
  // cout<<"Encryption\n";
  encrypt(mat,pairSet);

  // Decrypt the Cypher Text
  // cout<<"Decryption\n";
  decrypt(mat,pairSet);
}
```

Result:

File input output:

input.txt

```
welcome to cns lab
playfair
```

output.txt

```
P L A Y F
I R B C D
E G H K M
N O Q S T
U V W X Z

WE LC OM ET OC NS LA BX
UH YR TG MN SR OT AY CW

WE LC OM ET OC NS LA BX
```

Console Input Output:

```
PS E:\College\Final year\C&NS\practical> cd "e:\College\Final year\
) { .\playfair }
CnS lab playFair algorithm implementation
walchand
W A L C H
N D B E F
G I K M O
P Q R S T
U V X Y Z

CN SL AB PL AY FA IR AL GO RI TH MI MP LE ME NT AT IO NX
WE RC LD RW CV DH KQ LC IG QK ZF OK GS CB SM FP HQ KG BU

CN SL AB PL AY FA IR AL GO RI TH MI MP LE ME NT AT IO NX
```