



SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMKUR-572103
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CRYPTOGRAPHY AND NETWORK SECURITY LAB (7RCSL01)

Student Name: Pragati Shankar		USN: 1SI19CS090	Batch No: B2	Date: 15-11-2022
Evaluation:				
Write Up (10 marks)	Clarity in concepts (10 marks)	Implementation and execution of the algorithms (10 marks)	Viva (05 marks)	Total (35 marks)
Sl.No	Name of the Faculty In-Charge			Signature
1.	H K Vedamurthy			
2.	Gururaj S P			
Question No: 2				
Write a program to perform the following using Playfair cipher technique				
(i) Encrypt a given message M with different keys $\{k_1, k_2, \dots, k_n\}$. Print key and cipher text pair				
(ii) Decrypt the cipher texts obtained in (i) to get back M.				
Playfair Cipher:				
Construct 5 X 5 matrix using a keyword from left to right and from top to bottom, and then filling in the remainder of the matrix with the remaining letters in alphabetic order. The letters I and J count as one letter.				
Plaintext is encrypted two letters at a time, according to the following rules:				
<ol style="list-style-type: none">1. Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x, so that balloon would be treated as ba lx lo on.2. Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last.3. Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last.4. Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter.				

CODE:

```
#include <bits/stdc++.h>

using namespace std;

typedef struct{
    int row;
    int col;
}position;

char mat[5][5];

void generateMatrix(string key)
{
    int x=0,y=0;
    int flag[26]={0};
    for(int i=0;i<key.length();i++)
    {
        if(key[i]!='j')
            key[i]='i';
        if(flag[key[i]-'a']==0)
        {
            flag[key[i]-'a']=1;
            mat[x][y]=key[i];
            y++;
        }
        if(y==5)
        {
            x++;
            y=0;
        }
    }
    for(char i='a';i<='z';i++)
    {
        if(i=='j')
            continue;
        if(flag[i-'a']==0)
        {
            flag[i-'a']=1;
            mat[x][y]=i;
            y++;
        }
        if(y==5)
        {
            x++;
            y=0;
        }
    }
}

string formatMessage(string text)
```

```

{
    for(int i=0;i<text.length();i++)
    {
        if(text[i]=='j')
            text[i]='i';
        if(text[i]==' ')
            text.erase(text.begin()+i);
    }
    for(int i=1;i<text.length();i+=2)
    {
        if(text[i-1]==text[i])
            text.insert(i,"x");
    }
    if(text.length()%2!=0)
        text+="x";
    return text;
}

position getPosition(char c)
{
    position p;
    for(int i=0;i<5;i++)
    {
        for(int j=0;j<5;j++)
        {
            if(c==mat[i][j])
            {
                p={i,j};
                return p;
            }
        }
    }
    return p;
}

string encrypt(string msg)
{
    string ctext="";

    for(int i=0;i<msg.length();i+=2)
    {
        position p1=getPosition(msg[i]);
        position p2=getPosition(msg[i+1]);
        int x1=p1.row,y1=p1.col;
        int x2=p2.row,y2=p2.col;
        if(x1==x2)
        {
            ctext += mat[x1][(y1+1) % 5];
            ctext += mat[x2][(y2+1)%5];
        }
        else if( y1 == y2 )
        {
            ctext += mat[ (x1+1)%5 ][ y1 ];

```

```

        ctext += mat[(x2+1)%5][y2];
    }
    else
    {
        ctext += mat[x1][y2];
        ctext += mat[x2][y1];
    }
}
return ctext;
}
string decrypt(string msg)
{
    string ptext="";

    for(int i=0;i<msg.length();i+=2)
    {
        position p1=getPosition(msg[i]);
        position p2=getPosition(msg[i+1]);
        int x1=p1.row,y1=p1.col;
        int x2=p2.row,y2=p2.col;
        if(x1==x2)
        {
            ptext += mat[x1][--y1<0 ? 4: y1];
            ptext += mat[x2][--y2<0 ? 4: y2];
        }
        else if( y1 == y2 )
        {
            ptext += mat[--x1<0 ? 4: x1 ][y1];
            ptext += mat[--x2<0 ? 4: x2 ][y2];
        }
        else
        {
            ptext += mat[x1][y2];
            ptext += mat[x2][y1];
        }
    }
    return ptext;
}
int main()
{
    string plaintext;
    cout<<"Enter plaintext: ";
    getline(cin,plaintext);

    int i,n;
    cout<<"Enter the number of keys: ";
    cin>>n;
    string keys[n];
    for(i=0;i<n;i++)
    {
        cout<<"Enter the key: ";

```

```

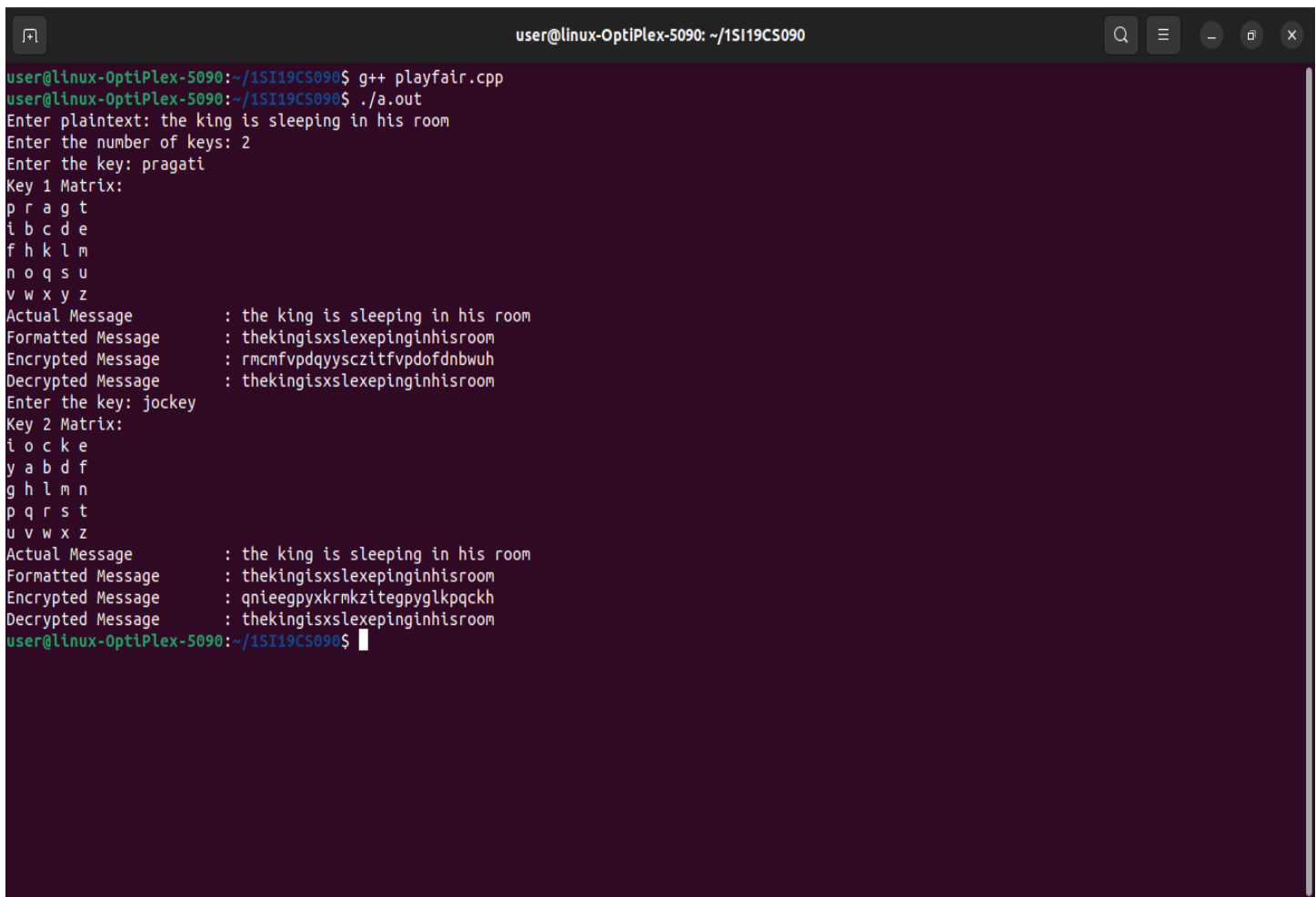
cin>>keys[i];

generateMatrix(keys[i]);

cout<<"Key "<<i+1<<" Matrix:"<<endl;
for(int j=0;j<5;j++)
{
    for(int k=0;k<5;k++)
    {
        cout<<mat[j][k]<<" ";
    }
    cout<<endl;
}
cout << "Actual Message \t\t: " << plaintext << endl;
string fmsg = formatMessage(plaintext);
cout << "Formatted Message \t: " << fmsg << endl;
string ciphertext = encrypt(fmsg);
cout << "Encrypted Message \t: " << ciphertext << endl;
string decryptmsg = decrypt(ciphertext);
cout<<"Decrypted Message \t: " << decryptmsg << endl;
}
return 0;
}

```

Output Screenshot:



```
user@linux-OptiPlex-5090: ~/1SI19CS090
user@linux-OptiPlex-5090:~/1SI19CS090$ g++ playfair.cpp
user@linux-OptiPlex-5090:~/1SI19CS090$ ./a.out
Enter plaintext: the king is sleeping in his room
Enter the number of keys: 2
Enter the key: pragati
Key 1 Matrix:
p r a g t
i b c d e
f h k l m
n o q s u
v w x y z
Actual Message      : the king is sleeping in his room
Formatted Message   : thekingisxslexepinginhisroom
Encrypted Message    : rmcmfvpdqyysczitfvdpdofdnbwuh
Decrypted Message    : thekingisxslexepinginhisroom
Enter the key: jockey
Key 2 Matrix:
i o c k e
y a b d f
g h l m n
p q r s t
u v w x z
Actual Message      : the king is sleeping in his room
Formatted Message   : thekingisxslexepinginhisroom
Encrypted Message    : qnieegpyxkrmkzitegpyglkpqckh
Decrypted Message    : thekingisxslexepinginhisroom
user@linux-OptiPlex-5090:~/1SI19CS090$
```