

Sir M. Visvesvaraya Institute of Technology

Cryptography and Network Security Mini Project Report

Ayushi (1MV17CS022) and Harsh Gahlot (1MV17CS038)

PROBLEM STATEMENT

2. Generate secure transmission of a secret message to the customer using Playfair.
Use the last six letters of the customer name as the key.

PLAYFAIR CIPHER IN CRYPTOGRAPHY

The **Playfair cipher** or **Playfair square** or **Wheatstone-Playfair cipher** is a manual symmetric encryption technique and was the first literal digram substitution cipher. The scheme was invented in 1854 by Charles Wheatstone, but bears the name of Lord Playfair for promoting its use.

The technique encrypts pairs of letters (*bigrams* or *digrams*), instead of single letters as in the simple substitution cipher and rather more complex Vigenère cipher systems then in use. The Playfair is thus significantly harder to break since the frequency analysis used for simple substitution ciphers does not work with it. The frequency analysis of bigrams is possible, but considerably more difficult. With 600 possible bigrams rather than the 26 possible monograms (single symbols, usually letters in this context), a considerably larger cipher text is required in order to be useful.

It was used for tactical purposes by British forces in the Second Boer War and in World War I and for the same purpose by the Australians during World War II. This was because Playfair is reasonably fast to use and requires no special equipment.

Encryption Technique

For the encryption process let us consider the following example:

Key: monarchy

Plaintext: instruments

The Playfair Cipher Encryption Algorithm:

The Algorithm consists of 2 steps:

1. Generate the key 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.

For example:

The key is "**monarchy**"

Thus the initial entries are

'm', 'o', 'n', 'a', 'r', 'c', 'h', 'y'

followed by remaining characters of **a-z(except 'j')** in that order.

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

2. Algorithm to 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.

For example:

PlainText: "instruments"

After Split: 'in' 'st' 'ru' 'me' 'nt' 'sz'

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).

For example:

Diagraph: "me"

Encrypted Text: cl

Encryption:

m -> c

e -> l

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

- 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).

For example:

Diagraph: "st"

Encrypted Text: tl

Encryption:

s -> t

t -> l

- 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.

For example:

Diagraph: "nt"

Encrypted Text: rq

Encryption:

n -> r

t -> q

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

For example:

Plain Text: "instrumentsz"

Encrypted Text: gatlmzclrqtx

Encryption:

i -> g

n -> a

s -> t

t -> l

r -> m

u -> z

m -> c

e -> l

n -> r

t -> q

s -> t

z -> x

in:

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

st:

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

ru:

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

me:

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

nt:

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

sz:

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

DECRYPTION TECHNIQUE

Decrypting the Playfair cipher is as simple as doing the same process in **reverse**. The receiver has the **same key** and can create the **same key table**, and then decrypt any messages made using that key.

in:

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

st:

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

ru:

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

me:

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

nt:

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

sz:

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

Merits and Demerits of Playfair Cipher

1. Merits:

- It is significantly harder to break since the frequency analysis technique used to break simple substitution ciphers is difficult but still can be used on $(25 \times 25) = 625$ digraphs rather than 25 monographs which is difficult.
- Frequency analysis thus requires more cipher text to crack the encryption.

2. Demerits:

- An interesting weakness is the fact that a digraph in the ciphertext (AB) and its reverse (BA) will have corresponding plaintexts like UR and RU (and also ciphertext UR and RU will correspond to plaintext AB and BA, i.e. the substitution is self-inverse). That can easily be exploited with the aid of frequency analysis, if the language of the plaintext is known.
- Another disadvantage is that a playfair cipher is a symmetric cipher; thus the same key is used for both encryption and decryption.

CODE SNIPPET

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

string inputs();
void outputs(string);
string l(string,int);
int r(string,int);
string made(string);
string code(string,string);
string prepare(string,int);
string decode(string, string);
int main()
{
    cout<<"          #####          "<<"\n";
    cout<<"          Customer cipher          "<<"\n";
    cout<<"          #####          "<<"\n";
    int c,lj,ind=0;
    string k;string normal="";
    string ct="";
    string matrix="";
    while(1)
    {
        cout<<"-----"<<"\n";
        cout<<"***Make a choice***"<<"\n";
        cout<<"1: **E/n/c/r/y/p/t**"<<"\n";
        cout<<"2: **D/e/c/r/y/p/t**"<<"\n";
        cout<<"3: **go back**"<<endl;
        cout<<"-----"<<"\n";
        cin>>c;
        switch(c)
        {
            case 1: k=inputs();
                    matrix=made(k);
                    normal=inputs();
                    lj=normal.length();
                    if(lj%2!=0)
                        ind=1;
                    ct=code(normal,matrix);
                    outputs(ct);
                    break;
            case 2: k=inputs();
                    matrix=made(k);
                    ct=inputs();
                    normal=decode(ct,matrix);
                    if(ind==1)
                        normal.erase(lj);
                    outputs(normal);
                    ind=0;
                    break;
            case 3: exit(0);
            default:cout<<"w*r*o*n*g"<<"\n";
        }
    }
}
```

```

    }
}
return 0;
}
string decode(string txtout,string matrix )
{
    int c[2];int a[2];int b[2];
    int s=0;
    int or1=0;
    string a1=txtout;
    int textlen=txtout.length();
    for (int l=0;l<textlen;l=l+2)
    {
        c[0] = 0;c[1] = 0;
        for(int i=0;i<26;i++)
        {
            if (a1[l] == matrix[i])
                c[0] = i;
            if (a1[l+1]==matrix[i])
                c[1] =i;
        }
        for (int m=0;m<2;m++)
        {
            if (c[m] < 5 )
                a[m] = 1;
            else if (c[m]<10&& c[m]>4)
                a[m] = 2;
            else if (c[m]<15 && c[m]>9)
                a[m] = 3;
            else if (c[m]<20 && c[m]>14)
                a[m] = 4;
            else
                a[m] = 5;
            if (c[m]%5==0)
                b[m] = 1;
            else if (c[m]%5==1)
                b[m] = 2;
            else if (c[m]%5==2)
                b[m] = 3;
            else if (c[m]%5==3)
                b[m] = 4;
            else
                b[m] = 5;
        }
        if (a[0]==a[1]||b[0]==b[1])
        {
            if (a[0]==a[1])
            {
                or1=a[0];
                c[0] = (c[0] - 1);
                c[1] = (c[1] - 1);
                for (int m=0;m<2;m++)
                {
                    if (c[m] < 5 )
                        a[m] = 1;
                    else if (c[m]<10&& c[m]>4)
                        a[m] = 2;
                    else if (c[m]<15 && c[m]>9)
                        a[m] = 3;

```



```

        else if (c[m]<20 && c[m]>14)
            a[m] = 4;
        else
            a[m] = 5;
    }
    if (a[0]!=or1||c[0]< 0)
        c[0]=c[0] + 5;
    if (a[1]!=or1||c[1]< 0)
        c[1]=c[1] + 5;
    a1[1] = matrix[c[0]];
    a1[l+1] = matrix[c[1]];
}
else if (b[0]==b[1])
{
    if (c[0]>c[1])
    {
        c[0]=(c[0] - 5) ;
        c[1]= (c[1] - 5) ;
        if(c[1] < 0)
            c[1]=(c[1] + 25) ;
    }
    else if (c[0] < c[1] && c[0] < 4)
    {
        c[0] = (c[0] + 5);c[1] = (c[1] - 5) ;
    }
    else
    {
        if (c[1]-c[0] == 20)
        {
            c[0]=c[0] + 20;
            c[1]=(c[1] - 5) ;
        }
        else
        {
            c[0]=(c[0] - 5) ;
            c[1]=(c[1] - 5) ;
            if (c[0] > 24)
                c[0] = c[0] - 25;
            if (c[1] > 24)
                c[1] = c[1] - 25;
        }
    }
    a1[1] = matrix[c[0]];
    a1[l+1] = matrix[c[1]];
}
}
else if(b[0]<b[1])
{
    s=abs(b[0] -b[1]);
    a1[1] = matrix[c[0]+s];
    a1[l+1] = matrix[c[1]-s];
}

else if (b[0] >b[1])
{
    s= (b[0]-b[1]);
    a1[1] = matrix[c[0]-s];
    a1[l+1] = matrix[c[1]+s];
}
}
}return a1;

```

```

}
string l(string a,int p1)
{
    for (int i=0;i<p1;i++)
    {
        if(a[i]<91&&a[i]>64)
            a[i]+=32;
    }
    return a;
}
int r(string a,int p1)
{
    int k=0;
    for (int i=0;i<p1;i++)
    {
        if(a[i]!=' ')
            a[k++]=a[i];
    }
    a[k]='\0';
    return k;
}
string made(string temp)
{
    int h=temp.length();
    h--;
    string temp1;
    char t;
    for(int i=0;i<6;i++)
    {
        t=temp[h];
        temp1=temp1+t;
        h--;
    }
    string m=temp1;
    int len=temp1.length();
    len=r(temp1,len);
    m=l(temp1,len);string a;
    int i=0,k=0,v=0,c=0;
    bool flag = true;
    for (int p = 0;p<len;p++)
        if (m[p] == 'j')
            m[p] = 'i';
    for (int u=0;u<len;u++)
    {
        for (k=0;k<u;k++)
        {
            if (m[u]==m[k])
                flag= false;
        }
        if (flag == true)
        {
            char z=m[u];
            a = a +z;i++;
        }
        flag=true;
    }
    while(c<26)
    {
        for (k=0;k<len;k++)

```

```

        {
            if (v==m[k]-97)
            {
                flag=false;
            }
        }
        if (flag==true)
        {
            if (v!= 9)
            {
                char z=97+v;
                a = a+z; i++;
            }
        }
        v++;c++;k=0;flag=true;
    }
    return a;
}

string code(string textin1,string matrix)
{
    int e=textin1.length();
    e=r(textin1,e);
    string keyword=l(textin1, e);
    keyword=prepare(keyword,e);
    textin1=keyword;
    string ch= keyword;
    int c[2];int a[2];int b[2];
    int s= 0;
    int or1= 0;
    int textlen =e;
    for (int l= 0; l < textlen; l = l + 2)
    {
        for (int i = 0; i< 26; i++)
        {
            if (textin1[l] == matrix[i])
                c[0] = i;
            if (textin1[l+1] == matrix[i])
                c[1] =i;
        }
        for(int m=0;m<2;m++)
        {
            if(c[m]<5 )
                a[m] = 1;
            else if(c[m]<10&& c[m]>4)
                a[m] = 2;
            else if(c[m]<15 && c[m]>9 )
                a[m] = 3;
            else if(c[m]<20 && c[m]>14)
                a[m] = 4;
            else
                a[m] = 5;
            if (c[m]%5==0)
                b[m] = 1;
            else if(c[m]%5==1)
                b[m] = 2;
            else if(c[m]%5==2)
                b[m] = 3;
            else if (c[m]%5==3)
                b[m] = 4;
        }
    }
}

```

```

        else
            b[m] = 5;
    }
    if(a[0]==a[1]||b[0]==b[1])
    {
        if (a[0]==a[1])
        {
            or1= a[0];
            c[0] = (c[0] + 1);
            c[1] = (c[1] + 1);
            for (int m=0;m<2;m++)
            {
                if (c[m]<5 )
                    a[m] = 1;
                else if (c[m]<10&& c[m]>4)
                    a[m] = 2;
                else if (c[m]<15 && c[m]>9)
                    a[m] = 3;
                else if (c[m]<20 && c[m]>14)
                    a[m] = 4;
                else
                    a[m] = 5;
            }
            if (a[0] != or1)
                c[0] = c[0] - 5;
            if (a[1] != or1)
                c[1] = c[1] - 5;
            ch[1] = matrix[c[0]];
            ch[1+1] = matrix[c[1]];
        }
        if (b[0] == b[1])
        {
            c[0] = (c[0] + 5) ;
            c[1] = (c[1] + 5) ;
            if (c[0] > 24)
                c[0] = c[0]-25;
            if (c[1] > 24)
                c[1] = c[1]-25;
            ch[1] = matrix[c[0]];
            ch[1+1] = matrix[c[1]];
        }
    }
    else if (b[0] < b[1])
    {
        s = abs(b[0] - b[1]);
        ch[1] = matrix[c[0]+s];
        ch[1+1] = matrix[c[1]-s];
    }
    else if (b[0] > b[1])
    {
        s= (b[0] - b[1]);
        ch[1] = matrix[c[0]-s];
        ch[1+1] = matrix[c[1]+s];
    }
}
return ch;
}
string prepare(string txtin,int n)
{

```

```

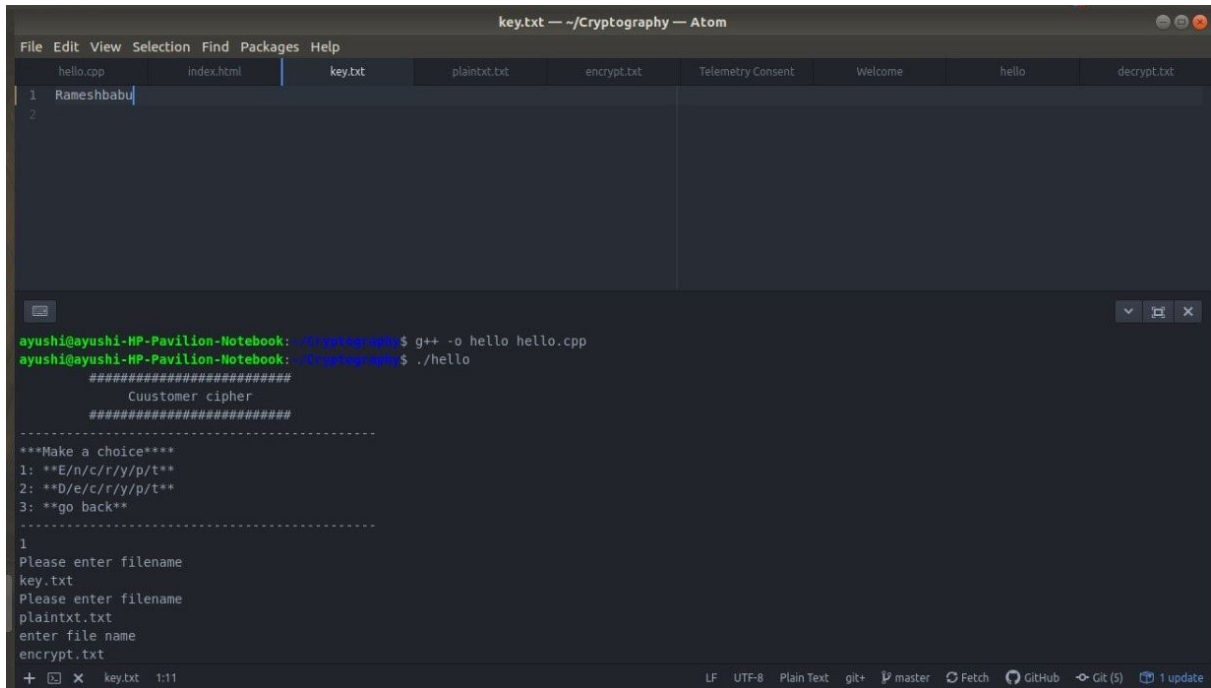
    for (int i=0;i<n;i++)
    {
        if (txtin[i]=='j')
            txtin[i]='i';
    }
    if (n%2!=0)
        txtin= txtin+ "x";
    n=txtin.length();
    for (int i=0;i<n;i=i+2)
    {
        if (txtin[i] == txtin[i+1])
        {
            txtin=txtin+" ";
            n= txtin.length();
            for (int j=n -1;j>i;j--)
                txtin[j]=txtin[j-1];
            txtin[i + 1] = 'x';
        }
    }
    return txtin;
}

string inputs()
{
    ifstream Inputs;string h,e;
    cout<<"Please enter filename"<<"\n";
    cin>> e;
    Inputs.open(e.c_str(),ios::in);
    if(!Inputs)
    {
        cout<<"error"<< "\n";
        return 0;
    }
    while(Inputs)
    {
        Inputs >> h;
    }
    Inputs.close();
    return h;
}

void outputs(string h)
{
    ofstream outs;string e;
    cout<<"enter file name"<<"\n";
    cin>>e;
    outs.open(e.c_str(), ios :: out);
    if(!outs)
        cout<<"error"<<"\n";
    if(outs.is_open())
    {
        outs<<h;
        outs.close();
    }
}

```

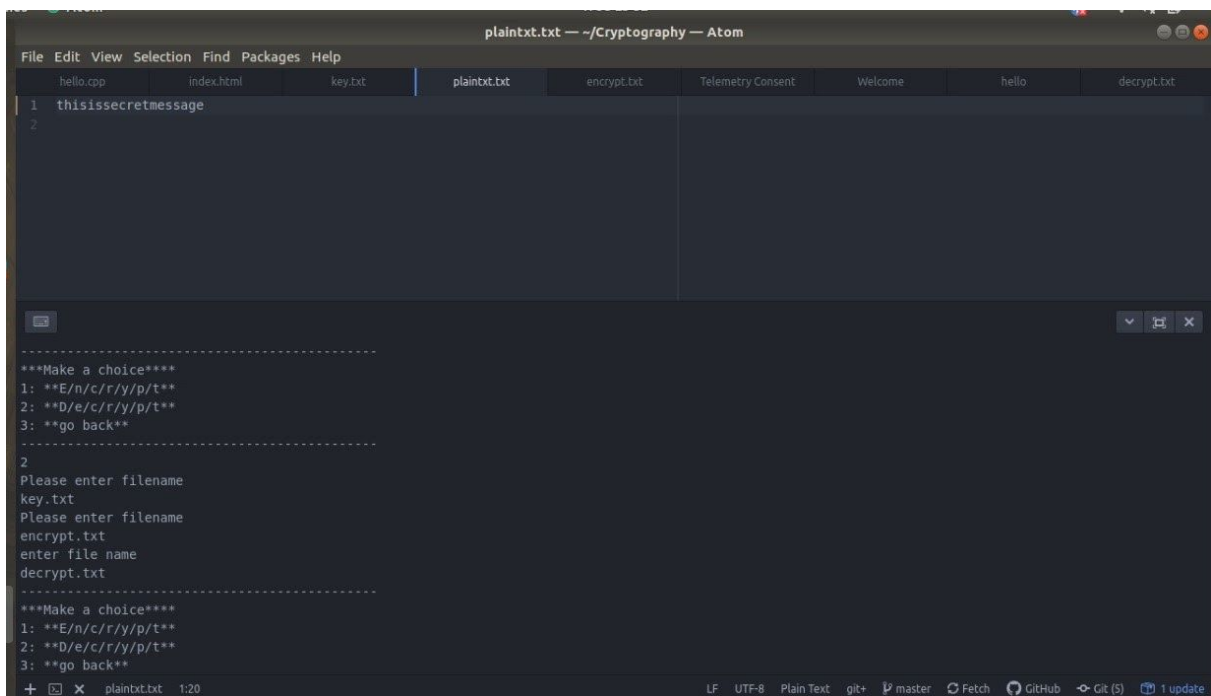
PROGRAM EXECUTION SNAPSHOTS



The screenshot shows the Atom editor with the file `key.txt` open. The file contains the text `Rameshbabu`. Below the editor, a terminal window displays the execution of a C++ program. The program prompts the user to enter a filename, and the user has entered `key.txt`. The terminal output shows the program's execution flow, including prompts for filename and file name, and the resulting output.

```
key.txt — ~/Cryptography — Atom
File Edit View Selection Find Packages Help
hello.cpp index.html key.txt plaintext.txt encrypt.txt Telemetry Consent Welcome hello decrypt.txt
1 Rameshbabu
2

ayushi@ayushi-HP-Pavilion-Notebook: ~/Cryptography$ g++ -o hello hello.cpp
ayushi@ayushi-HP-Pavilion-Notebook: ~/Cryptography$ ./hello
#####
Cuustomer cipher
#####
***Make a choice***
1: **E/n/c/r/y/p/t**
2: **D/e/c/r/y/p/t**
3: **go back**
-----
1
Please enter filename
key.txt
Please enter filename
plaintext.txt
enter file name
encrypt.txt
+ key.txt 1:11
LF UTF-8 Plain Text git+ master Fetch GitHub Git (5) 1 update
```



The screenshot shows the Atom editor with the file `plaintext.txt` open. The file contains the text `thisissecretmessage`. Below the editor, a terminal window displays the execution of a C++ program. The program prompts the user to enter a filename, and the user has entered `key.txt`. The terminal output shows the program's execution flow, including prompts for filename and file name, and the resulting output.

```
plaintext.txt — ~/Cryptography — Atom
File Edit View Selection Find Packages Help
hello.cpp index.html key.txt plaintext.txt encrypt.txt Telemetry Consent Welcome hello decrypt.txt
1 thisissecretmessage
2

ayushi@ayushi-HP-Pavilion-Notebook: ~/Cryptography$ g++ -o hello hello.cpp
ayushi@ayushi-HP-Pavilion-Notebook: ~/Cryptography$ ./hello
#####
Cuustomer cipher
#####
***Make a choice***
1: **E/n/c/r/y/p/t**
2: **D/e/c/r/y/p/t**
3: **go back**
-----
2
Please enter filename
key.txt
Please enter filename
encrypt.txt
enter file name
decrypt.txt
+ plaintext.txt 1:20
LF UTF-8 Plain Text git+ master Fetch GitHub Git (5) 1 update
```

```
encrypt.txt — ~/Cryptography — Atom
File Edit View Selection Find Packages Help
hello.cpp index.html key.txt plaintext.txt encrypt.txt Telemetry Consent Welcome hello decrypt.txt
1 rsnunupgfgqlfazuhcfx

2: **D/e/c/r/y/p/t**
3: **go back**
-----
2
Please enter filename
key.txt
Please enter filename
encrypt.txt
enter file name
decrypt.txt
-----
***Make a choice***
1: **E/n/c/r/y/p/t**
2: **D/e/c/r/y/p/t**
3: **go back**
-----
3
ayushi@ayushi-HP-Pavilion-Notebook: ~/Cryptography$
+ [X] encrypt.txt 1:7 LF UTF-8 Plain Text git+ master Fetch GitHub Git (5) 1 update
```

```
decrypt.txt — ~/Cryptography — Atom
File Edit View Selection Find Packages Help
hello.cpp index.html key.txt plaintext.txt encrypt.txt Telemetry Consent Welcome hello decrypt.txt
1 thisissecretmesxsag

2: **D/e/c/r/y/p/t**
3: **go back**
-----
2
Please enter filename
key.txt
Please enter filename
encrypt.txt
enter file name
decrypt.txt
-----
***Make a choice***
1: **E/n/c/r/y/p/t**
2: **D/e/c/r/y/p/t**
3: **go back**
-----
3
ayushi@ayushi-HP-Pavilion-Notebook: ~/Cryptography$
+ [X] decrypt.txt 1:1 LF UTF-8 Plain Text git+ master Fetch GitHub Git (5) 1 update
```

WEB APPLICATION VIEW

PLAYFAIR CIPHER

Gateway for secure transmission of messages using Encryption and Decryption

Enter Customer Name

your cipher square is:

s,h,b,a,u,c,d,e,f,g,i,k,l,m,n,o,p,q,r,t,v,w,x,y,z

Write the Message

your cipher square is:

s,h,b,a,u,c,d,e,f,g,i,k,l,m,n,o,p,q,r,t,v,w,x,y,z

Write the Message

puococbcfoglfbvuflb

Encrypted Message

PLAGIARISM CHECK REPORT



PLAGIARISM SCAN REPORT

Date May 20, 2020

Exclude URL: NO

	Unique Content	95%
	Plagiarized Content	5%
	Paraphrased Plagiarism	0%
Word Count		1,154
Readability (max. 100)		76
Records Found		15

CONTENT CHECKED FOR PLAGIARISM:

```
#include <bits/stdc++.h>
using namespace std;
string inputs();
void outputs(string);
```