PLAYFAIR CIPHER

By- Ujjwal Shekhar(17BCE0952)
- Baibhav Singh(17BCE2359)
Slot- A2+TA2

INTRODUCTION

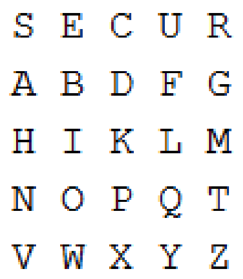
1. The Playfair cipher was the first practical digraph substitution cipher.

2. The scheme was invented in 1854 by Charles Wheatstone, but was named after Lord Playfair who promoted the use of the cipher.

3. The technique encrypts pairs of letters (digraphs), instead of single letters as in the simple substitution cipher.

ALGORITHM

1). The 'key' for a Playfair cipher is generally a word, for the sake of example we will choose ‘secure'. This is then used to generate a 'key square', e.g.



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

Any sequence of 25 letters can be used as a key, so long as all letters are in it and there are no repeats. Note that there is no 'j', it is combined with 'i'. We now apply the encryption rules to encrypt the plaintext.

ALGORITHM....

2). Remove any punctuation or characters that are not present in the key square (this may mean spelling out numbers, punctuation etc.).

3). Identify any double letters in the plaintext and replace the second occurrence with an 'x' e.g. 'hammer' -> 'hamxer’.

4). If the plaintext(which is to be encrypted) has an odd number of characters, append an 'x' to the end to make it even.

5). Break the plaintext into pairs of letters, e.g. 'hamxer' -> 'ha mx er’

6). The algorithm now works on each of the letter pairs

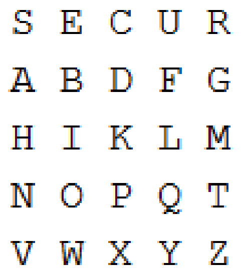
ENCRYPTION ALGORITHM…

Based on the pair of letters position in key square(given in previous slide).

A). If the letters are in different rows and columns, replace the pair with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair.

For example:

If pair of letters is RL then it is replaced by U(for R)M(for L).



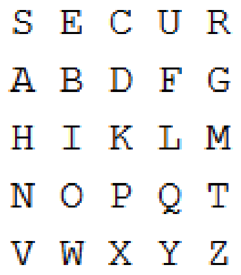
S	E	C	U	R
A	B	D	F	G
H	I	K	L	M
N	O	P	Q	T
V	W	X	Y	Z

ENCRYPTION ALGORITHM…

B). If the letters appear on the same row of the table, replace them with the letters to their immediate right respectively (wrapping around to the left side of the row if a letter in the original pair was on the right side of the row)

For example:

If pair of letters is SC then it is replaced by E(for S)U(for C).



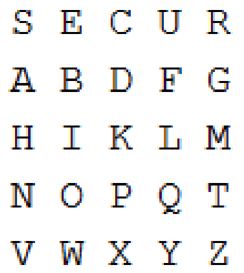
S	E	C	U	R
A	B	D	F	G
H	I	K	L	M
N	O	P	Q	T
V	W	X	Y	Z

ENCRYPTION ALGORITHM….

C). If the letters appear on the same column of the table, replace them with the letters immediately below respectively (wrapping around to the top side of the column if a letter in the original pair was on the bottom side of the column).

For example:

If pair of letters is SH then it is replaced by A(for S)N(for H).



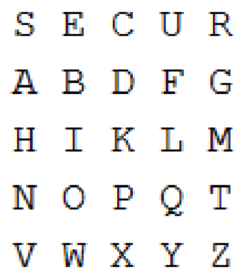
S	E	C	U	R
A	B	D	F	G
H	I	K	L	M
N	O	P	Q	T
V	W	X	Y	Z

DECRYPTION ALGORITHM

A). If the letters are in different rows and columns, replace the pair with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair.

For example:

If pair of letters is UM then it is replaced by R(for U)L(for M).



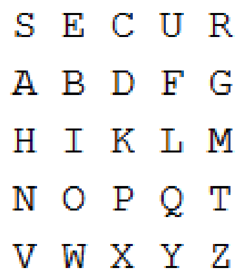
S	E	C	U	R
A	B	D	F	G
H	I	K	L	M
N	O	P	Q	T
V	W	X	Y	Z

DECRYPTION ALGORITHM….

B). If the letters appear on the same row of the table, replace them with the letters to their immediate left respectively (wrapping around to the right side of the row if a letter in the original pair was on the left side of the row)

For example:

If pair of letters is EU then it is replaced by S(for E)C(for U).



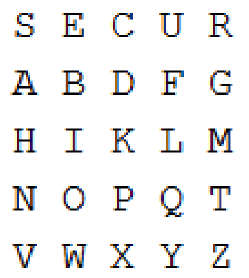
S	E	C	U	R
A	B	D	F	G
H	I	K	L	M
N	O	P	Q	T
V	W	X	Y	Z

DECRYPTION ALGORITHM…

C). If the letters appear on the same column of the table, replace them with the letters immediately above respectively (wrapping around to the down side of the column if a letter in the original pair was on the top side of the column).

For example:

If pair of letters is AN then it is replaced by S(for A)H(for N).



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

JAVA SOURCE CODE

Executable .exe file-

https://drive.google.com/file/d/1o8GZSe7KylV-wtG2TqmPdMvmwaSioG4b/view?usp=sharing

Text file of the source code-

https://drive.google.com/file/d/11rdrtdzm6FTa0h2nMaAWarGSU42F91ZL/view?usp=sharing

REFERENCES

http://practicalcryptography.com/ciphers/playfair-cipher

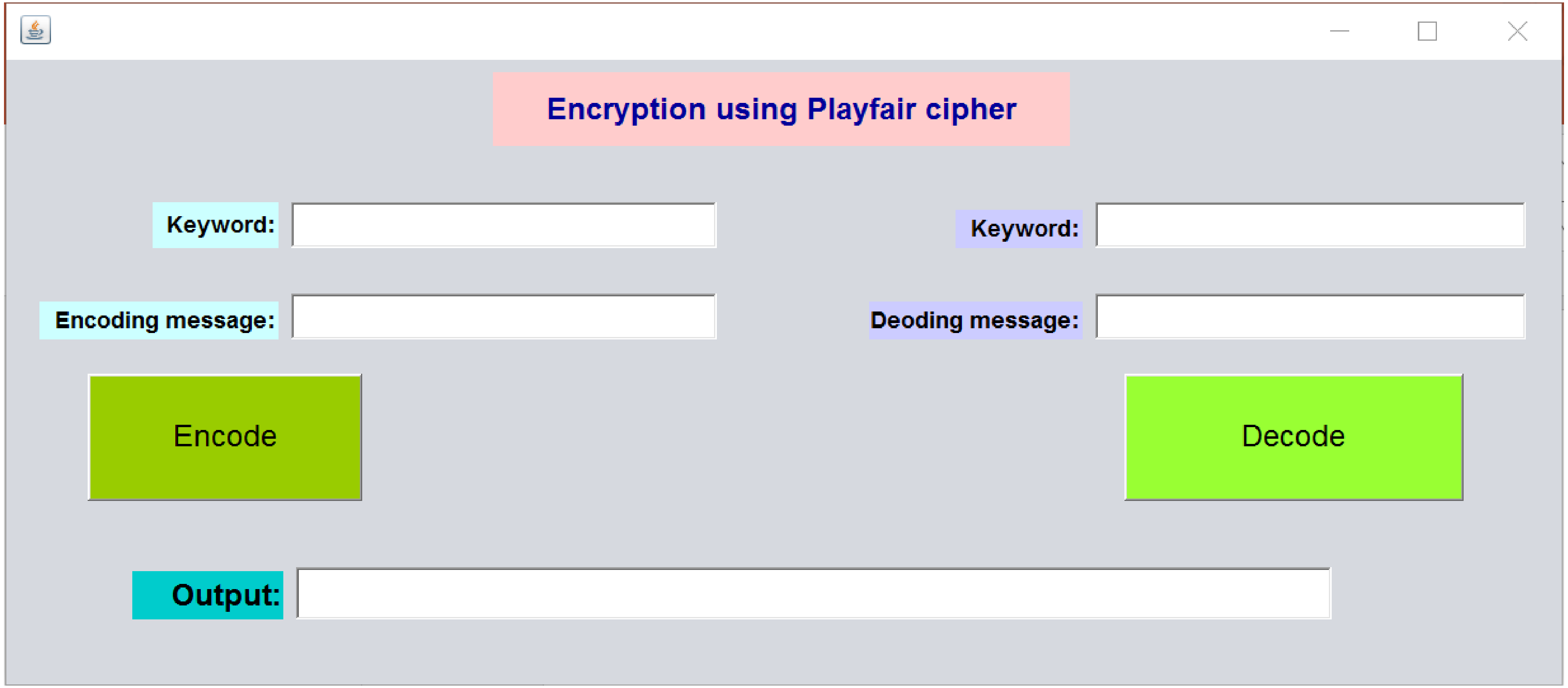
https://learncryptography.com/classical-encryption/playfair-cipher

```java
/*
* To change this license header, choose License Headers in Project Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.
*/
import java.awt.Point;
import java.util.Scanner;

/**
*
* @author lenovo
*/
public class newplayfair {
    private int length = 0;
    private String [][] table;


    private newplayfair(){
        System.out.println("Please input the keyword for the Playfair cipher.");
    Scanner sc = new Scanner(System.in);
    String keyword = parseString(sc);//it removes space from in between of words and is defined in
line 66

     while(keyword.equals(""))
      keyword = parseString(sc);//if user just press enter, that is gave null string then it will again take
key value
    System.out.println();

    table = this.cipherTable(keyword);//instance created in line 25


    int uj=1;
    while(uj==1)
    {
        int choice;
        System.out.println("\n1.Encode \n2.Decode \n3.Exit \nEnter your choice: ");
        choice=sc.nextInt();

        switch(choice)
        {
            case 1:
                encode();
                break;
            case 2:
                decode();
                break;
            case 3:
                System.out.println("Thank you using ");
                uj=0;
                break;
        }
    }


    }


    private String parseString(Scanner s){
    String parse = s.nextLine();
    parse = parse.toUpperCase();
    parse = parse.replaceAll("[^A-Z]", "");//It will go character by character in parse string and if it
matches from any character in A-Z then it is replaced by " "
    parse = parse.replace("J", "I");//replacing I and J
    return parse;
   }

    private void encode(){
    Scanner sc = new Scanner(System.in);
    System.out.println("Please input the message to be encoded using the previously given
keyword");

    String input = parseString(sc);
    while(input.equals(""))
      input = parseString(sc);
    System.out.println("Parsed input messege is "+input);
    System.out.println();

    String output = cipher(input);

    System.out.println("The encoded message is: "+output);


    }






   private String[][] cipherTable(String key){
    String[][] playfairTable = new String[5][5];
    String keyString = key + "ABCDEFGHIKLMNOPQRSTUVWXYZ";//concatanation

    // fill string array with empty string
    for(int i = 0; i < 5; i++)
```

```
   for(int j = 0; j < 5; j++)
    playfairTable[i][j] = "";//just an intiallization part

  for(int k = 0; k < keyString.length(); k++){
   boolean repeat = false;
   boolean used = false;
   for(int i = 0; i < 5; i++){
    for(int j = 0; j < 5; j++){
     if(playfairTable[i][j].equals("" + keyString.charAt(k))){
      repeat = true;//to make sure that character place is not getting used
     }else if(playfairTable[i][j].equals("") && !repeat && !used){
      playfairTable[i][j] = "" + keyString.charAt(k);
      used = true;
     }
    }
   }
  }
  return playfairTable;
 }




private String cipher(String in){
  length = (int) in.length() / 2 + in.length() % 2;

  // insert x between double-letter digraphs & redefines "length"
  for(int i = 0; i < (length - 1); i++){
   if(in.charAt(2 * i) == in.charAt(2 * i + 1)){
    in = new StringBuffer(in).insert(2 * i + 1, 'X').toString();//converting because it is a string buffer
    length = (int) in.length() / 2 + in.length() % 2;
   }
  }

  //Formation of digraph

  String[] digraph = new String[length];
  for(int j = 0; j < length ; j++){
   if(j == (length - 1) && in.length() / 2 == (length - 1))
    in = in + "X";// adds an x to the last digraph, if necessary
   digraph[j] = in.charAt(2 * j) +""+ in.charAt(2 * j + 1);//Using concatanation forming word which
will make
  }

  // encodes the digraphs and returns the output
  String out = "";
  String[] encDigraphs = new String[length];
```

```
   encDigraphs = encodeDigraph(digraph);
   for(int k = 0; k < length; k++)
    out = out + encDigraphs[k];
   return out;
  }



 private String[] encodeDigraph(String di[]){
   String[] enc = new String[length];
   for(int i = 0; i < length; i++){
    char a = di[i].charAt(0);
    char b = di[i].charAt(1);
    int r1 = (int) getPoint(a).getX();
    int r2 = (int) getPoint(b).getX();
    int c1 = (int) getPoint(a).getY();
    int c2 = (int) getPoint(b).getY();
    System.out.println("point for  "+a+" "+r2+","+r1);
    System.out.println("point for  "+b+" "+c2+","+c1);
    System.out.println();

    // case 1: letters in digraph are of same row, shift columns to right
    if(r1 == r2){
     c1 = (c1 + 1) % 5;
     c2 = (c2 + 1) % 5;

    // case 2: letters in digraph are of same column, shift rows down
    }else if(c1 == c2){
     r1 = (r1 + 1) % 5;
     r2 = (r2 + 1) % 5;

    // case 3: letters in digraph form rectangle, swap first column # with second column #
    }else{
     int temp = c1;
     c1 = c2;
     c2 = temp;
    }

    //performs the table look-up and puts those values into the encoded array
    enc[i] = table[r1][c1] + "" + table[r2][c2];
   }
   return enc;
  }



 private Point getPoint(char c){
   Point pt = new Point(0,0);//intialliztion of point array
  for(int i = 0; i < 5; i++)
   for(int j = 0; j < 5; j++)
    if(c == table[i][j].charAt(0))
     pt = new Point(i,j);//updating point array
  return pt;
 }



private void decode(){
  String out;
  Scanner sc=new Scanner(System.in);
  System.out.println("Enter the text you want to decode");
  out=sc.next();

   String decoded = "";
  for(int i = 0; i < out.length() / 2; i++){
   char a = out.charAt(2*i);
   char b = out.charAt(2*i+1);
   int r1 = (int) getPoint(a).getX();

   int r2 = (int) getPoint(b).getX();

   int c1 = (int) getPoint(a).getY();
   int c2 = (int) getPoint(b).getY();
   System.out.println("point for  "+a+" "+r1+","+r2);
   System.out.println("point for  "+b+" "+c1+","+c2);
   System.out.println();
   if(r1 == r2){
    c1 = (c1 + 4) % 5;
    c2 = (c2 + 4) % 5;
   }else if(c1 == c2){
    r1 = (r1 + 4) % 5;
    r2 = (r2 + 4) % 5;
   }else{
    int temp = c1;
    c1 = c2;
    c2 = temp;
   }
   decoded = decoded + table[r1][c1] + table[r2][c2];
  }
  //return decoded;
  System.out.println("Decoded message is "+decoded);
 }


private void decode(String out){
  String decoded = "";
  for(int i = 0; i < out.length() / 2; i++){
   char a = out.charAt(2*i);
   char b = out.charAt(2*i+1);
   int r1 = (int) getPoint(a).getX();

   int r2 = (int) getPoint(b).getX();

   int c1 = (int) getPoint(a).getY();
   int c2 = (int) getPoint(b).getY();
   System.out.println("point for  "+a+" "+r1+","+r2);
   System.out.println("point for  "+b+" "+c1+","+c2);
   System.out.println();
   if(r1 == r2){
    c1 = (c1 + 4) % 5;
    c2 = (c2 + 4) % 5;
   }else if(c1 == c2){
    r1 = (r1 + 4) % 5;
    r2 = (r2 + 4) % 5;
   }else{
    int temp = c1;
    c1 = c2;
    c2 = temp;
   }
   decoded = decoded + table[r1][c1] + table[r2][c2];
  }
  System.out.println("Decoded message is "+decoded);
 }

public static void main(String[] args)
{
  newplayfair pf = new newplayfair();
}

}
```

SNIPPETS



Encryption using Playfair cipher

Keyword:

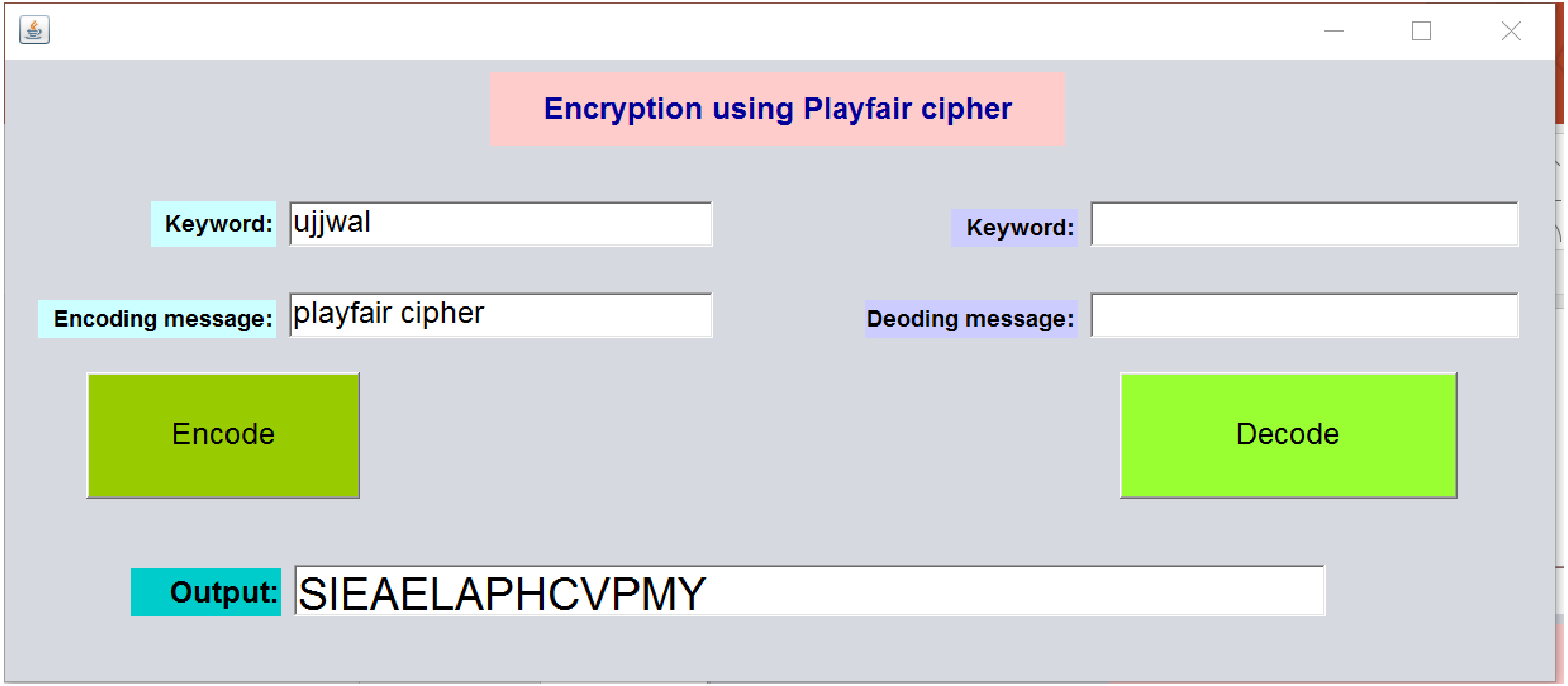
Encoding message:

Keyword:

Deoding message:

Output:

SNIPPETS(ENCRYPTION USING A PUBLIC KEY)



Encryption using Playfair cipher

Keyword: ujjwal

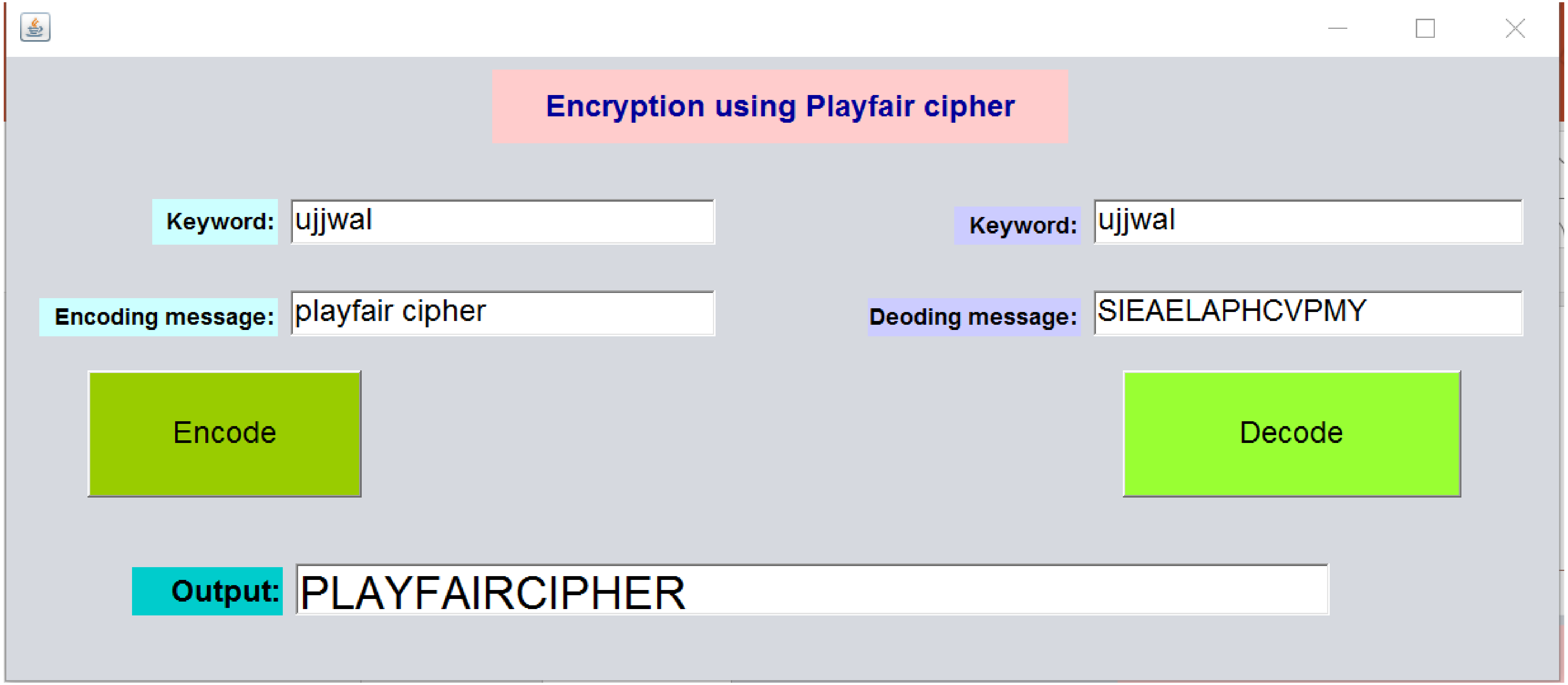
Encoding message: playfair cipher

Keyword:

Deoding message:

Output: SIEAELAPHCVPMY

SNIPPETS(DECRYPTION USING CORRECT PUBLIC KEY)



Encryption using Playfair cipher

Keyword: ujjwal

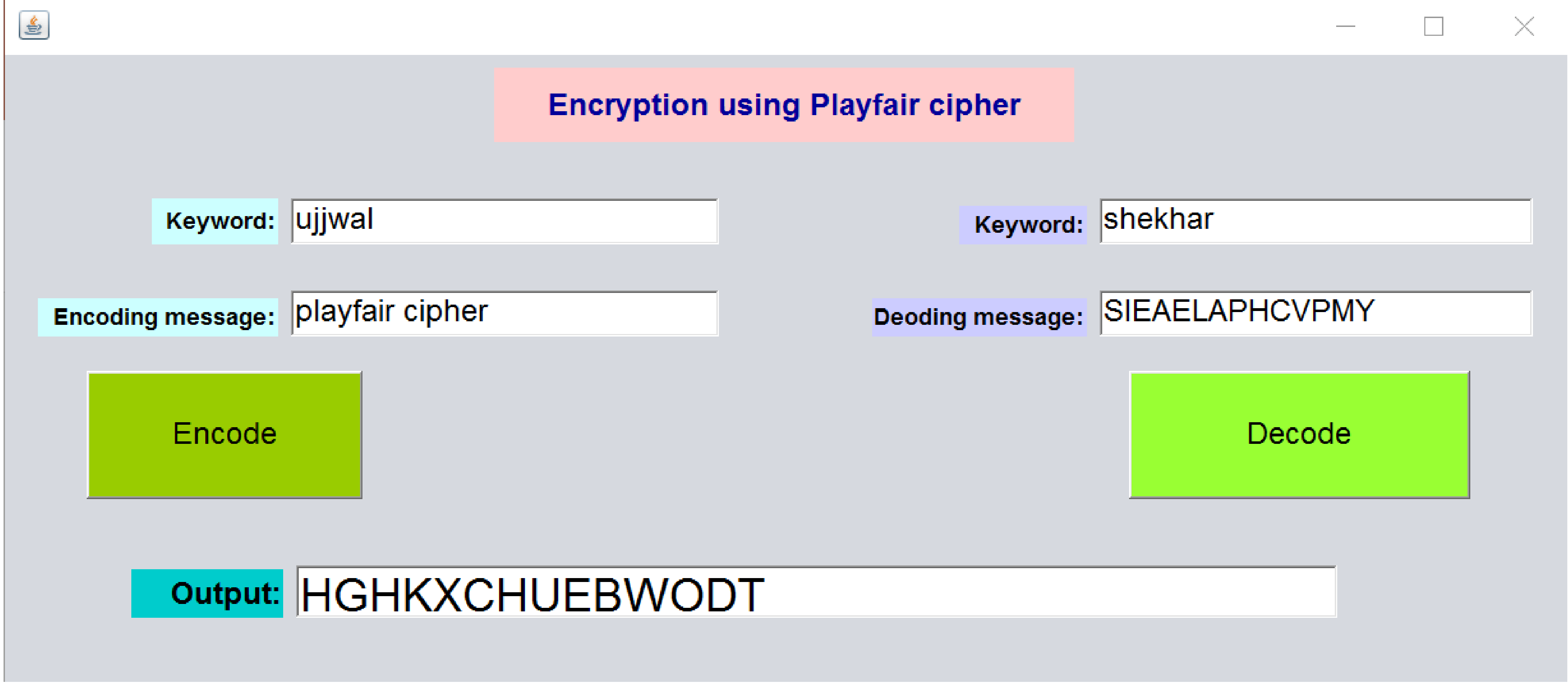
Encoding message: playfair cipher

Keyword: ujjwal

Deoding message: SIEAELAPHCVPMY

Output: PLAYFAIRCIPHER

SNIPPETS(DECRYPTION USING WRONG PUBLIC KEY)



Encryption using Playfair cipher

Keyword: ujjwal

Encoding message: playfair cipher

Keyword: shekhar

Deoding message: SIEAELAPHCVPMY

Output: HGHKXCHUEBWODT