SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

SCHOOL OF COMPUTING

DEPARTMENT OF DATASCIENCE AND BUSINESS SYSTEMS

21CSC202J OPERATING SYSTEMS

MINI PROJECT REPORT

IMAGE ENCRYPTION AND DECRYPTION

Name:Angelin J G

Register Number:RA2112704010009

Mail ID:aj7764@srmist.edu.in

Department: Data science and Business Systems

Specialization: Data Science

Semester: 3

Team Members Name : Charvitha Yerukonda,Angelin JG
RegistrationNumber:RA2112704010019,RA2112704010009

CONTENT PAGE

ABSTRACT----------2
Chapter 1 : Introduction and Motivation ----------3
Chapter 2: Review of Existing methods and their Limitations----------6
Chapter 3 : Proposed Method with Flow Diagram----------7
Chapter 4: Modules Description----------8
Chapter 5: Implementation requirements----------9
Chapter 5: Output Screenshots----------10
Conclusion----------12
References----------13
Appendix A – Source Code----------14
Appendix B – GitHub Profile and Link for the Project----------18

ABSTRACT

The main objective of our project is to provide security of the image-based data with the help of suitable key and protect the image from illegal copying and distribution. These In today's world data security is the major problem which is to be face. In order to secure data during communication, data storage and transmission we use XOR encryption algorithm.

The XOR encryption algorithm is an example of symmetric encryption where the same key is used to both encrypt and decrypt a message.

Encryption has been a longstanding way for sensitive information to be protected. Historically, it was used by militaries and governments. In modern times, encryption is used to protect data stored on computers and storage devices, as well as data in transit over networks. Symmetric ciphers, also referred to as secret key encryption, use a single key. The key is sometimes referred to as a shared secret because the sender or computing system doing the encryption must share the secret key with all entities authorized to decrypt the message.

CHAPTER-1

INTRODUCTION & MOTIVATION

INTRODUCTION:-

Computer has become an essential device now a days. The main use of computer is to store data and send it from one location to other. The information that is shared must be transferred in a secured manner. To ensure secured transmission of information, data is encrypted to unreadable formats by an unauthorized person. Cryptography is the science of information security which has become a very critical aspect of modern computing systems towards secured data transmission and storage. The exchange of digital data in cryptography results in different algorithms that can be classified into two cryptographic mechanisms: symmetric key in which same key is used for encryption and decryption and asymmetric key in which different keys are used for encryption and decryption.

Images are broadly used in numerous processes. As a result, the safety of image data from unauthorized access is crucial at the hands of user. Image encryption plays a significant role in the field of information hiding. Image hiding or encryption methods and algorithms ranges from simple spatial domain methods to more complicated and reliable frequency domain. Image Encryption Using Rubik's Cube Based Algorithm is the process to transform the image securely so that no unauthorized user can be able to decrypt the image. Image encryption have applications in many fields including the internet communication, transmission, medical imaging etc.

First, in order to scramble the pixels of gray-scale original image, the principle of Rubik's cube is deployed which only changes the position of the pixels. Using two random secret keys, the bitwise XOR is applied into the rows and columns. These steps can be repeated until the number of iterations is not reached. Numerical simulation has been performed to test the validity and the security of the proposed encryption algorithm.

XOR ENCRYPTION ALGORITHM:

The XOR Encryption algorithm is a very effective yet easy to implement method of symmetric encryption. Due to its effectiveness and simplicity, the XOR Encryption is an extremely common component used in more complex encryption algorithms used nowadays.

The XOR encryption algorithm is an example of symmetric encryption where the same key is used to both encrypt and decrypt a message.

Reapplying the same XOR mask (using the same key) to the cipher text outputs the original plain text. The following truth table (based on the XOR truth table) demonstrates how the encryption process works.

The XOR encryption algorithm can be applied to any digital/binary information, included text based information encoded using the 8-bit ASCII code. In this case the encryption key can be expressed as a string of characters.

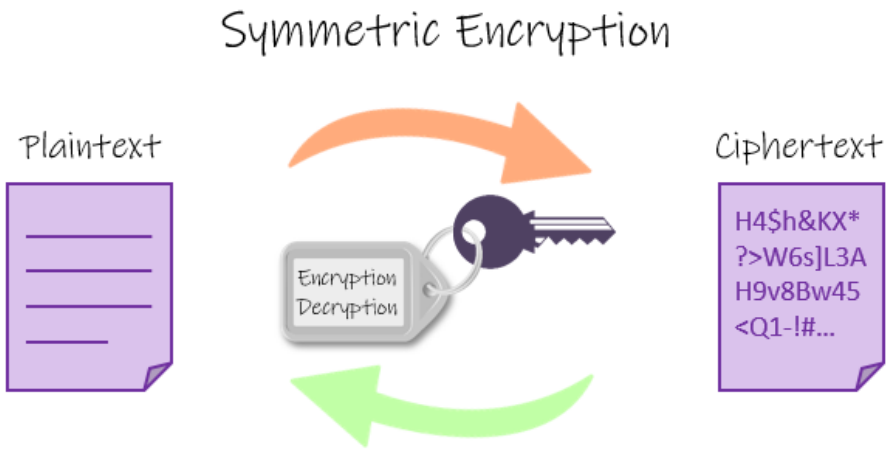
By itself, the XOR encryption can be very robust if:

It is based on a long key that will not repeat itself. (e.g. a key that contains as many bits/characters as the plaintext)

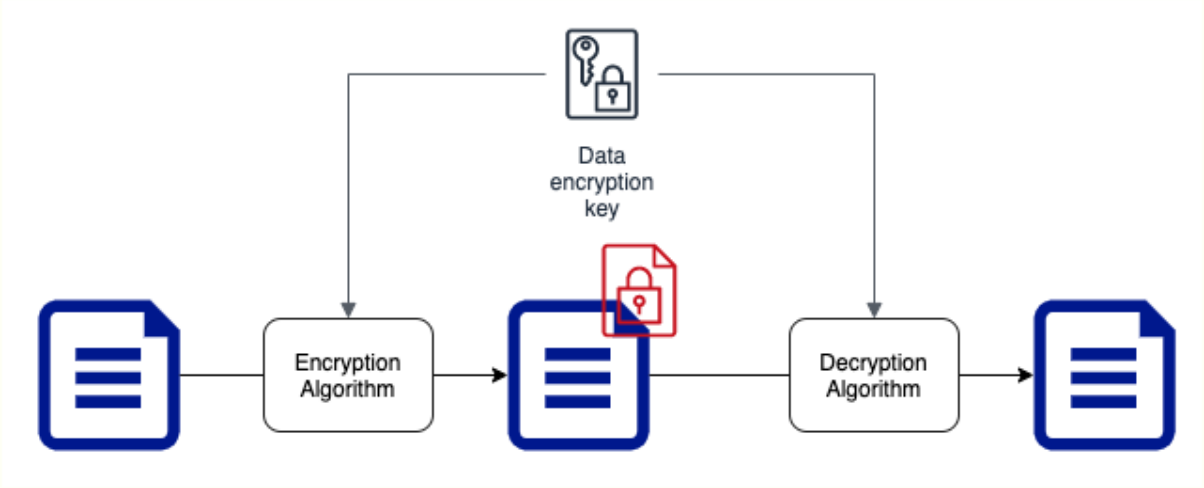
A new key is randomly generated for any new communication.

The key is kept secret by both the sender and the receiver.

When a large quantity of text is to be encrypted, a shorter repeating encryption key is used to match the length of the plain text. However re-using the same key over and over, or using a shorter repeating key results in a less secure method where the cipher text could be decrypted using a frequency analysis.



Algorithm:-



MOTIVATION/PROBLEM STATEMENT: -

The main problems that arise during image transmission process are with respect to the time it takes to reach the destination and its security level. For real time image encryption only those ciphers are preferable which takes lesser amount of computational time.When an original image is been transmitted from one end to another over a network, security is essential. In order to secure the data which we send it has to be encrypted. So that the intruder would not get to know or the data cannot be hacked. The images are split and combined, at the decryption end using the same techniques original images are obtained. Hence, the image transmission takes place safely.

CHAPTER – 2

REVIEW OF EXISTING METHOD & LIMITATION

EXISTING METHOD: -

Now a day's digital images are used extensively. So, digital security is a very important aspect in today's research area.Using visual cryptography we can encrypt an important message in such an unintelligible format that no one can identify the original image from that format and sender can send the original image or message securely to the receiver,because unauthorized access to that original message may create disastrous security issue .Therefore it is important to convert the original message into random like cipher using a secret key, in such a way that the original message can be recovered again. Image scrambling is a very important aspect of visual cryptography technique. There are various methods of scrambling technique, such as blockbased scrambling, pixel scrambling etc. Using these scrambling techniques many researchers had reported many algorithms to create cipher images. It is observed that using XOR cipher we can encrypt an original image into a scrambled image which cannot distinguish. In this scrambling process, we use a secret key. We use the same secret key to unscramble the encrypted image

LIMITATION:-

The problem with XOR encryption is that for long runs of the same characters, it is very easy to see the password. Such long runs are most commonly spaces in text files. Say your password is 8 chars, and the text file has 16 spaces in some line (for example, in the middle of ASCII-graphics table). If you just XOR that with your password, you'll see that output will have repeating sequences of characters. The attacker would just look for any such, try to guess the character in the original file (space would be the first candidate to try), and derive the length of the password from length of repeating groups.

Binary files can be even worse as they often contain repeating sequences of 0x00 bytes. Obviously, XORing with those is no-op, so your password will be visible in plain text in the output! An example of a very common binary format that has long sequences of nulls is .doc.

CHAPTER – 3

PROPOSED ALGORITHM

A. Encryption Algorithm

- Take an image of m x m dimension.
- Select a secret key of the same dimension as per the image dimension such as [X11, X12, x13 ….xmm]
- Converted each element of image and key matrix in binary format
- Apply bitwise XOR operation between an element of image matrix and the corresponding element of the key matrix
- After step 4 we will get the encrypted image

B. Decryption Algorithm

- Take the encrypted image.
- Select the same secret key which was used to encrypt the image.
- Converted each element of encrypted image and key matrix in binary format.
- Apply bitwise XOR operation between an element of encrypted image matrix and the corresponding element of the key matrix.
- After step 2 we will get the decrypted image which will be same as original one.

CHAPTER – 4

MODULES DESCRIPTION

Uploading the image

Encryption

Selecting the same image

Decryption

Uploading the image-

This module will have the user to input the key ,make the user select the image that has to be encrypted.

Encryption:-

This will make sure whether the image is encrypted.

Selecting the same image:-

Come back,again select the image that has to be decrypted.

Decryption :-

Check if the image is visible now.

CHAPTER-5

IMPLEMENTATION REQUIREMENTS

HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system.

- Hard disk : 50 GB
- RAM : 1 GB
- Processor : Intel Core i3

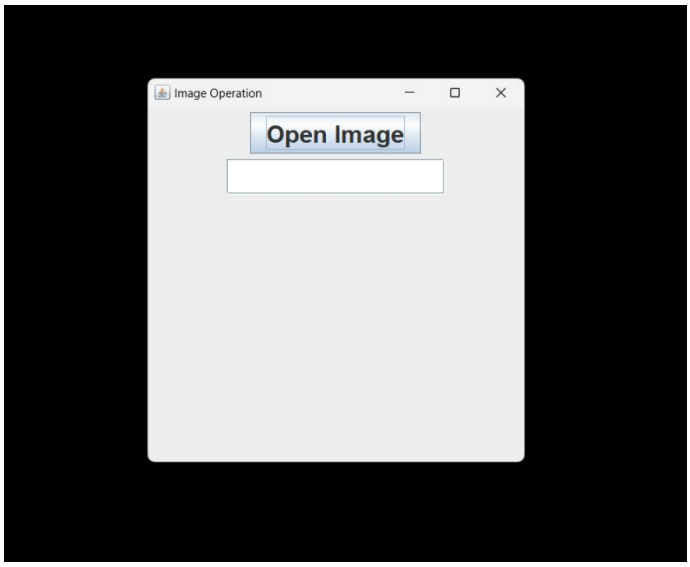
SOFTWARE REQUIREMENTS

- Windows 9 and above
- JDK 19
- VisualStudio Code

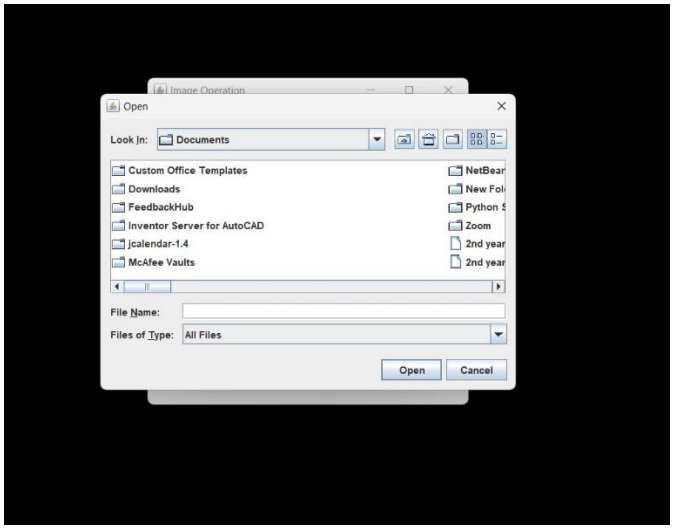
CHAPTER-5

OUTPUT SCREENSHOTS

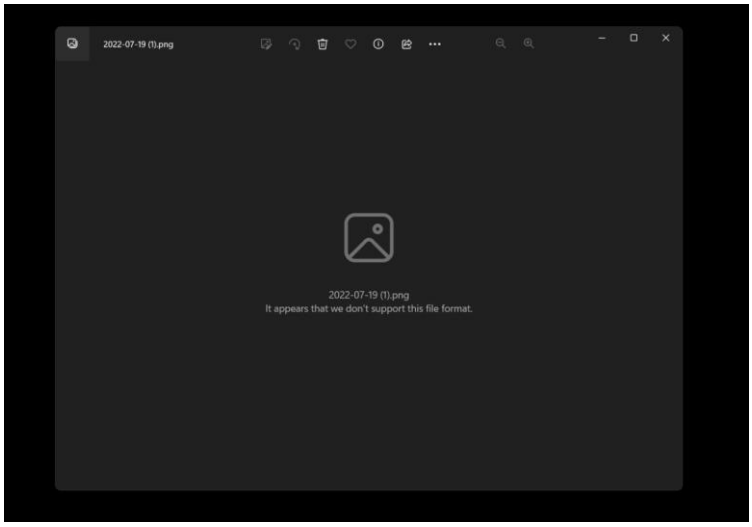
Upload image



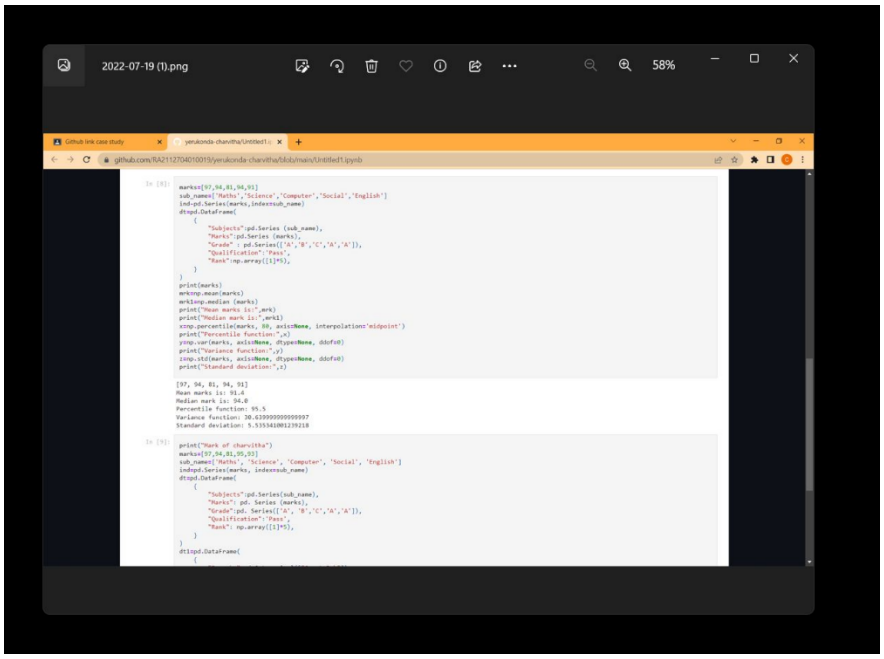
Select image



Encrypted image



After decryption



CONCLUSION

Information, mostly exist in the form of Images. When, information is shared in environment like peer-to peer environment, the images are highly vulnerable to risk of being hacked. Our work proposes a novel encryption algorithm using XOR operation to resolve the security issue.The observation illustrates the generation of cipher text as junk characters or special characters which does not provide information directly and confuses the hacker.

REFERENCES:

1)P. Sharma, D. Mishra, V.K. Sarthi, P. Bhatpahri and R. Shrivastava,"Visual Encryption Using Bit Shift Technique", International Journal of Scientific Research in Computer Science and Engineering, Vol. 5, No. 3, pp. 57-61, June 2017.

2) X. Y. Wang, Y.Q Zhang, and L.T. Liu, "An enhanced sub-image encryption method", Optics and Laser in Engineering, Vol. 86, pp.248-254, November 2016

Appendix-A

Source Code:-

```
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;

import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
public class ImageOperation {

   public static void operate(int key)
   {

      JFileChooser fileChooser=new JFileChooser();
      fileChooser.showOpenDialog(null);
      File file=fileChooser.getSelectedFile();
      //file FileInputStream
      try
      {

         try (FileInputStream fis = new FileInputStream(file)) {
```

```
            byte []data=new byte[fis.available()];
            fis.read(data);
            int i=0;
            for(byte b:data)
            {
               System.out.println(b);
               data[i]=(byte)(b^key);
               i++;
            }

            FileOutputStream fos=new FileOutputStream(file);
            fos.write(data);
            fos.close();
         }
         JOptionPane.showMessageDialog(null, "Done");

      }catch(Exception e)
      {
         e.printStackTrace();
      }
   }

   public static void main(String[] args) {

      System.out.println("this is testing");

      JFrame f=new JFrame();
```

```
f.setTitle("Image Operation");
f.setSize(400,400);
f.setLocationRelativeTo(null);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

Font font=new Font("Roboto",Font.BOLD,25);
//creating button
JButton button=new JButton();
button.setText("Open Image");
button.setFont(font);

//creating text field

JTextField textField=new JTextField(10);
textField.setFont(font);



button.addActionListener((ActionEvent e)->{
   System.out.println("button clicked");
   int temp = 0;
   try {
     temp = Integer.parseInt(textField.getText());
   } catch (NumberFormatException nfe) {
       // handle the exception
   }

   operate(temp);
```

```
        });

        f.setLayout(new FlowLayout());

        f.add(button);
        f.add(textField);
        f.setVisible(true);

    }
}
```

Appendix B

Github link :-

https://github.com/aj7764/os