Steganography and Cryptography



COMPUTER ENGINEERING SEM-V, UNIVERSITY POLYTECHNIC

FACULTY OF ENGINEERING AND TECHNOLOGY

JAMIA MILLIA ISLAMIA (A CENTRAL UNIVERSITY)

NEW DELHI-110025

SESSION: 2015-2016

**CONTIBUTORS: PROJECT GUIDE:**

MOHD.JAHID MS. FARAH JAMAL ANSARI

MOHD.ILYAS QUADRI

NEHA SHAMS

* 1. Introduction
  2. Steganogarphy
     1. Types of Steganogarphy
        1. Image Steganogarphy

1. Algorithms
   1. LSB
   2. RSA

**3**  Methodology

**3.1** Encrypt method

**3.2** Decrypt method

**3.3** Encode method

* 1. Decode meth

**4** Glimpse of Application

**5 5.1** Technical Details

**5.2** Creation of User Space

**5.3** Software Support

**5.4** Application

**6** Conclusion and future scope

**Abstract**

Steganography is the dark cousin of cryptography, the use of codes. While cryptography provides privacy, steganography is intended to provide security.

The software is performing image steganography.

The software takes the inputs an image and data to be hidden. It uses **RSA** algorithm to encrypt the data and use the **LSB** algorithm to hide the encrypted data into the image.

To decode the image, we first extract the data from the image and then decrypt the data.

In future we will hide an image into another image. Moreover, we will make an android application for steganography and cryptography.

Objective is not only to prevent the message being read but also to hide its existence.

**Chapter-1**

**Introduction**

* 1. **What is Steganography?**

Steganographygraphy means “covered writing”.

* Art and science of hiding information in ways that prevent detection.
* Can be used in audio, text, packet headers, or images.

**Steganography**

Steganography is the practice of concealing a file, message, image, or video within another

file, message, image, or video.

The word *steganography* combines the Greekwords *steganos* meaning "covered,concealed,

or protected", and *graphein*  meaning "writing".

The majority of today’s steganographic systems uses multimedia objects like image, audio,

video etc as cover media because people often transmit digital pictures over email and other

Internet communication.

Both hide a message. But Steganography is meant to make the meassage invisible, while

cryptography changes the meassage’s form, by means of replacement and/or algorithm.

* + 1. **Types of Steganography**

1. Image Steganography
2. Sound Steganogarphy
3. Video Steganography

In this we are only interested with Image Steganography.

**1.1.1.2 Image Steganography**

Images are used as the popular cover medium for steganography. A message is embedded

in a digital image using an embedding algorithm, using the secret key. The resulting

stego-image is send to the receiver. On the other side, it is processed by the extraction

algorithm using the same key. During the transmission of stego- image unauthenticated

persons can only notice the transmission of an image but cant see the existence of the

hidden message.

Real Image Data to be hidden



**Fig.1 (Showing Process of Steganography)**

** **

**Fig.1.1 (Original Image)** **Fig.1.2 (Embedded Image)**

**1.2 Cryptography**

Cryptography the art or science encompassing the principles and methods of

transforming an intelligible message into one that is unintelligible, and then

retransforming that message back to its original form.

**Plaintext** - the original intelligible message n.

**Ciphertext**- the transformed message.**Cipher** - an algorithm for transforming an

intelligible message into one that is unintelligible by transposition and

substitution methods .

**Key** - some critical information used by the cipher, known only to the sender &

receiver.

**Encipher (encode) -** the process of converting plaintext to cipher text using a

cipher and a key.

**Decipher (decode)** - the process of converting cipher text back into plaintext using

a cipher and a key.

**Chapter-2**

**Algorithms**

**2.1 Least Significant Bit(LSB)**

☺ One of the most common technique.

☺ Alter LSB of each pixel.

☺Hiding ASCII code, one letter at a time.

**Example**

Message: 10010110

Original Data: 10010011 11000011 10101010 11001100

LSBs are selected: 1001001**1** 1100001**1** 1010101**0** 1100110**0**

Process of Stagenography:

1 0 0 1 0 1 1 0

1001001**1** 1100001**0** 1010101**0** 1100110**0**

**Like:**

Pixels of Images

10010101 00001101 11001001 10010110

00001111 11001010 10011111 00010000

11001011

Now suppose we want to "hide" the following 9 bits of data (the hidden data is usually compressed prior to being hidden): 101101101. If we overlay these 9 bits over the LSB of the 9 bytes above, we get the following (where bits in bold have been changed):

Encoded pixels

10010101 00001100 11001001 10010111

00001110 11001011 10011111 00010000

11001011

**2.2 RSA Algorithm for Cryptography**

1. The most common public-key algorithm is called the RSA method after its inventors

( Rivert, Shamir, and Aldleman).

1. The private key here is a pair of numbers (d,N); the public key is also a pair of numbers

(e,N).

1. The sender uses the following algorithm to encrypt the message:

C=PemodN

In this algorithm, P is the plaintext, which is represented as a number; C is the number

that represents the Ciphertext.

1. The receiver uses the following algorithm to encrypt the message:

P=CdmodN

**Steps to apply RSA**

1 .Select P & Q (P & Q are two large prime numbers)

2. Calculate the product of P & Q.

N=P\*Q

3. Select public key .It is the factor of

Z= (P-1)(Q-1)

4. 1<e<Z

5. (d\*e)mod Z = 1

6. Encrypt C= m pow (e)mod n

7. Decrypt Cpow (d)mod n =m

**Chapter -3**

**Methodology**

**3.1 Encrypt method**

int P=3,Q=11,Z,N,E=7,D=3;

int encrypt(int m)

{

N=P\*Q;

Z=(P-1)\*(Q-1);

int C=((int)Math.pow(m,E))%N;

return C;

}

**3.2 Decrypt method**

int decrypt(int m)

{

int data=(int)Math.pow(x,e.D) )%e.N

retrun data;

}

**3.3 Encode Method**

public void encode\_text(byte[] img,byte[] msg,int start)

{

for(int i=0;i<msg.length;i++)

{

for(int j=7;j>=0;j--,start++)

{

int b=msg[i];

b>>>=j;

b&=1;

img[start]=(byte)((img[start]&0xFE)|b);

}

}

}

Here: **img** represents pixels of images in the form of array of bytes.

**msg** represents messages to be hide in the form of array of bytes.

**start** represents the position of byte of image where the next bit of the message to hide in

the image.

**3.4 Decode Method**

public byte[] decode\_text(byte[] img)

{

for(int i=0;i<32;i++)

{

int bit=(img[i]&1);

bit<<=(31-i);

len|=bit;

}

bytemsg[]=new byte[len];

for(int i=32,k=0;k<len;k++)

{

for(int j=7;j>=0;j--,i++)

{

msg[k]=(byte)((msg[k] << 1) | (img[i] & 1));

}

}

returnmsg;

}

Here: **img** represents the bytes of images from which bytes are decoded.

**msg** represents the bytes of decoded messages from the image.

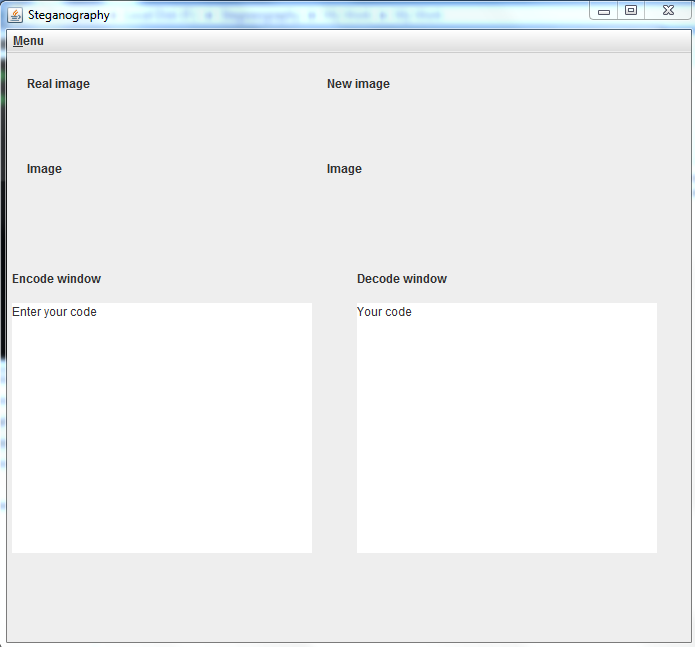
**Chapter-4**

**Glimpse of Application**

1. Interface of the application

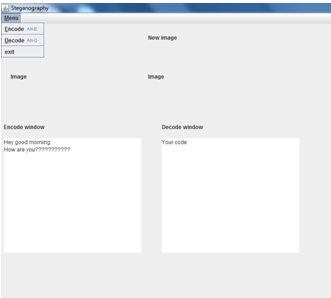
Contains a menu bar with options:

|  |
| --- |
| Encode |
| Decode |
| Exit |



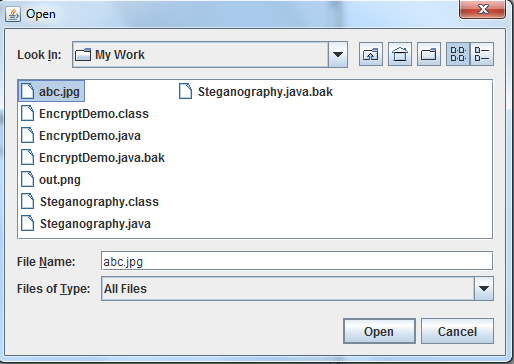
**Fig.4 (Interface of application)**

1. Enter the message to be encoded. And chose the image for the encryption.



**Fig.4.1( Input the message)**

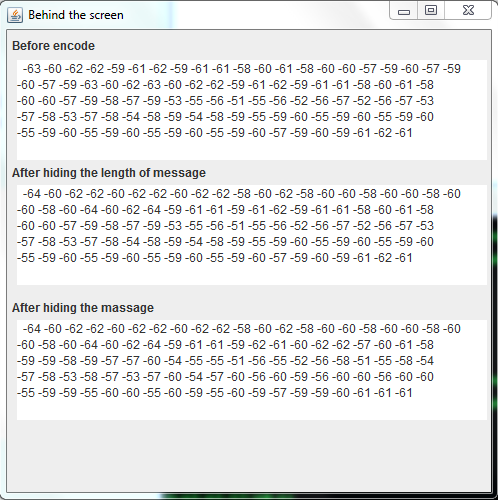
1. Select the file.



**Fig.4.2( Chose a image file)**

1. Now the magic has done.

Pixels of Image before hiding the data.

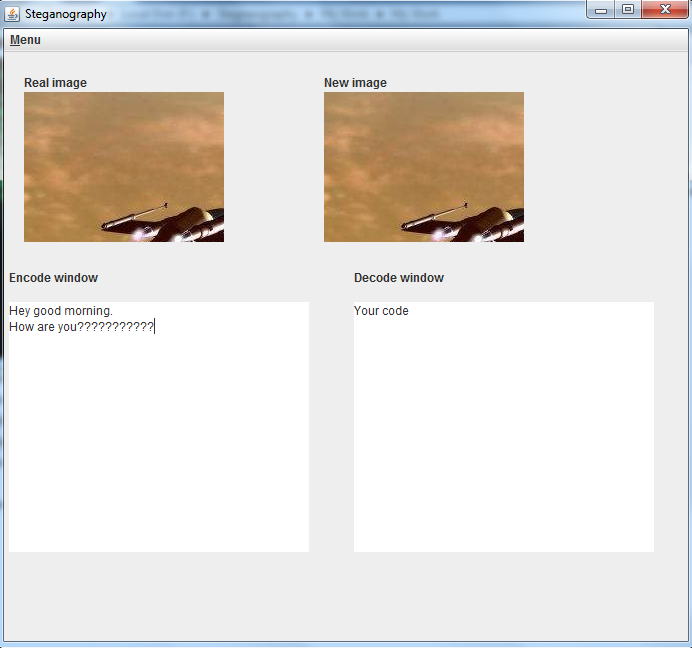


Pixels of image after hiding the message.

Pixels of image after hiding the length of message.

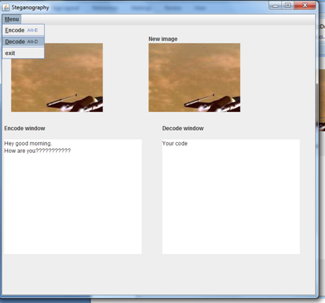
**Fig.4.3 (Pixels of Image in various phases of process)**

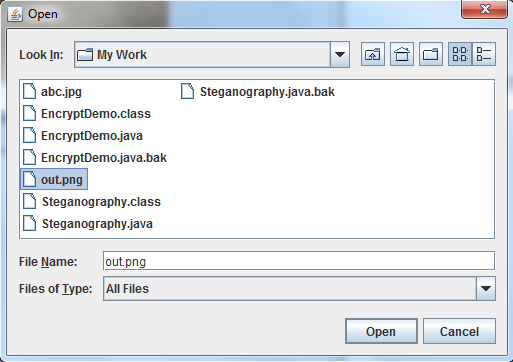
Here two images are present one is the original image and the embedded image with data.



**Fig.4.4( Showing images before and after the process respectively)**

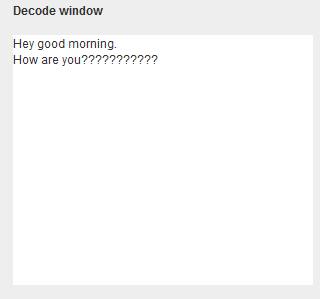
1. Now chose the image to be decoded.





**Fig.4.5 (Selecting a image to be decoded)**

1. Here we got the message from the embedded image.



**Fig.4.6 (Hidden Message)**

**Chapter-5**

**Implementation**

**5.1 Technical Details:**

☺ Using java.awt.Image, ImageIO

☺ The package contain all the necessary classes and method along with interfaces

that are necessary for the manipulation of the images.

**5.2 Creation of User Space:**

☺ User space is created for preserving the original file, so that all the modifications

aredone in the user space.

☺ In the object of BufferedImage, using ImageIO.read method we take the original

image.

☺ Using createGraphics and drawRenderedImage method of Graphics class, we

createour user space in BufferedImage object.

**5.3 Software Support:**

☺Operating System:

Windows or GNU/LINUX

☺Programming Language:

Java

GUI: Swings

☺System Requirement:

JDK 1.4 or Higher

Java being a platform independent language, the projects runs on any platform.

* 1. **Application**

1. Confidential communication and secret data storing.
2. Protection of data alteration.

**Chapter-6**

**Conclusion and future scope**

* 1. Hide Image within another image.



* 1. Mobile application of steganography and cryptography using android.