Tribhuvan University

Institute of Science and Technology

Himalaya College of Engineering

[Code: CSC-404]

A

Final Year Project Report

On

"Image Steganography"

by

Aakash Shrestha [7212/072]

Sandhya Khadka [7251/072]

Simran Dhoju [7255/072]

Supriya Amatya [7259/072]

Submitted to Department of Computer Science and Information Technology in partial fulfillment of the requirement for the Bachelor's degree in Computer Science and Information Technology

Department of Computer Science and Information Technology

August 5, 2019

DECLARATION BY CANDIDATES

We, the final year (seventh semester) B.Sc. CSIT students of Himalaya College of Engineering, hereby declare that the project report entitled **"Image Steganography"** is carried by us under the guidance and supervision of **Mr. Bhupendra Luhar** (Lecturer, Tribhuwan University)

We assure that this project work embedded the result of the original work and the contents of the project have not submitted to anybody else for the award of degree. This project is purely of academic interest.

We have followed the guide lines provided by the university in writing the report.

Signature of the candidates

Mr. Aakash Shrestha

Date: August 5, 2019

Miss Sandhya Khadka

Miss Simran Dhoju

Miss Supriya Amatya

SUPERVISOR'S CERTIFICATE

This is to certify that the project entitled **"Image Steganography"** is done by Aakash Shrestha (7212), Sandhya Khadka (7251), Simran Dhoju (7255) and Supriya Amatya (7259) of fourth year B.Sc. CSIT under the guidance and supervision of **Mr. Bhupendra Luhar**, Lecturer, Himalaya College of Engineering, towards the partial fulfillment of bachelor's degree of Computer Science and Information Technology.

Signature of Supervisor

Mr. Bhupendra Luhar

Visiting Faculty HCOE

Himalaya College of Engineering

Chyasal, Lalitpur

Date:

Seal:

EVALUATION COMMITTEE

This is to certify that the project entitled **“Image Steganography”** is done by Aakash Shrestha (7212), Sandhya Khadka (7251), Simran Dhoju (7255) and Supriya Amatya (7259) of fourth year B.Sc. CSIT under the guidance and supervision of **Mr. Bhupendra Luhar**, Lecturer, Himalaya College of Engineering, towards the partial fulfillment of bachelor’s degree of Computer Science and Information Technology.

Signature of HOD	**Signature of Supervisor**
Mr. Himal Chand Thapa	Mr. Bhupendra Luhar
Head of Department (CSIT)	Visiting Faculty HCOE
Himalaya College of Engineering	Himalaya College of Engineering
Chyasal, Lalitpur	Chyasal, Lalitpur

External Examiner

ACKNOWLEDGEMENT

Firstly, we would like to thank Himalaya College of Engineering for providing us this platform to gain knowledge about different aspects of Computer Science and Information Technology.

We would also like to thank our Head of Department **Er. Himal Chand Thapa** who played an important role in providing us advice and suggestion towards the selection of our project topic as well as for providing us with necessary content and classes regarding the project. We are also grateful for the care and support from our supervisor **Mr. Bhupendra Luhar** and for providing us all necessary suggestions and corrections he provided, for our project.

In addition, we also want to thank our friends and classmates for helping us with the guidelines we needed during the completion of our project.

ABSTRACT

Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Cryptography is a technique associated with the process of converting ordinary plain text into unintelligible text and vice-versa. In contrast to cryptography, steganography is not to keep others from knowing the hidden information but it is to keep others from thinking that the information even exists. Together cryptography and steganography can provide a powerful basis for data security.

The main purpose of this project is to produce a security tool based on steganography and cryptography techniques for sending and receiving sensitive information over the internet. The program first encrypts the message data using AES algorithm and then embeds the result of encrypted data in the provided image file using steganography technique. The system also provides the feature for extracting the hidden data from the corresponding image file and decrypting the extracted data for eventually finding the original message. The embedding process follows image's LSB replacement algorithm. To obtain the hidden message, the process is reversed.

Keyword: Steganography, Cryptography, LSB, AES.

Table of Contents

List of Figures

List of Tables

Abbreviation

AES: Advanced Encryption Standard

API: Application Program Interface

DFD: Data Flow Diagram

ICT: Information and Communication Technology

IDE: Integrated Development Environment

IP: Internet Protocol

LSB: Least Significant Bit

RGB: Red Green Blue

TCP: Transmission Control Protocol

UI: User Interface

Chapter 1. Introduction

1.1 Introduction to Image Steganography

Due to advances in ICT, most information is kept electronically. Consequently, the security of information has become a fundamental issue. Maintaining the integrity and confidentiality of sensitive information, blocking the access of sophisticated hackers is very important. Steganography is a technique of hiding information in digital media [1]. Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular ones because of their frequency on the internet. In contrast to cryptography, it is not to keep others from knowing the hidden information but it is to keep others from thinking that the information even exists.

Steganography hides the secrete message within the host data set in imperceptible way which is to be reliably communicated to a receiver. The host data set is purposely corrupted, but in a covert way, designed to be invisible to an information analysis. The growing possibilities of modern communications need the special means of security especially on computer network. With the increment in exchange of data over in internet, the network security is becoming more important. Therefore, the confidentiality and data integrity require protection against unauthorized access and use. This has resulted in an explosive growth of the field of information hiding.

Steganography is the art of concealing information in ways that prevent the detection of hidden messages. Similarly, Cryptography is a technique associated with the process of converting ordinary plain text into unintelligible text and vice-versa. Together cryptography and steganography can provide a powerful basis for data security. These techniques become more important as more people join the cyberspace revolution.

1.2 Problem Statement

With growing digitization in every field, digital security has become a fundamental aspect. Also, because of development in the internet technology, digital media can be transmitted conveniently over the network. This calls for security over the internet. Throughout history steganography and cryptography have been used to secretly communicate information between people. In the past, means of cryptography and steganography were carried out using traditional methods of pen and paper, using invisible ink, etc.

Therefore, with the increasing use of communication of information over digital medium, security for digital methods are to be developed. Steganography hides data onto a stego medium. Steganography along with cryptographic tools can ensure even more data security.

1.3 Objective

The main objectives of this project are:

- To produce security tool based on steganography and cryptography techniques combined.
- To avoid drawing suspicion to the existence of a hidden message.

1.4 Scope and Limitations

The scope of this project is to limit unauthorized access and provide better security during message transmission. To meet the requirements, simple and basic LSB approach of steganography had been used. The program first encrypts the message data using AES algorithm and embeds the result of encrypted data in the provided image file using steganography technique for sending over the network. The system also provides the feature for extracting the hidden data from the corresponding image file and decrypting the extracted data for eventually finding the original message.

Steganography means hiding data into another data. It can be used to hide data such as text, image, audio, video etc. within a cover image, video etc. While the system program provides a way to hide text data into a cover image file, it is limited only to data of the mentioned types i.e., text data onto image data.

1.5 Report Organization

Chapter 1 includes subtopics as introduction to image steganography, problem statement, objectives, scope and limitations.

Chapter 2 contains literature review.

Chapter 3 comprises of requirement analysis, feasibility study and process models of the system. The requirements analysis further consists of functional and non-functional requirements. Economic, technical and operational feasibilities are some listed feasibilities. Process model includes DFD diagrams.

Chapter 4 consists of system architecture, class diagram activity diagram and algorithm used.

Chapter 5 includes tools used and testing. Testing comprises of unit testing, integrated testing and system testing.

Chapter 6 includes conclusion and future enhancements.

Chapter 2. Literature Review

The word steganography is derived from the Greek words stegos meaning cover and grafia meaning writing [2]. In Image steganography the information is hidden exclusively in images. Steganography is the art and science of secret communication. It is the practice of encoding/embedding secret information in a manner such that the existence of the information is invisible. The actual files can be referred to as cover text and the cover image. After inserting the secret message, it is referred to as stego-medium. A stego-key is used for hiding encoding process to restrict detection or extraction of the embedded data [3].

For more secure data transfer, cryptography is used along with steganography. In cryptography the message is encrypted using encryption algorithm along with secret key and transferred it to the other end, then receiver can decrypt it and get original message by using decryption algorithm. The grouping of these two approaches enhances the security of data. The combination of these two methods will satisfy the requirement such as capacity and security for data transmission over an open channel. If the attacker were able to detect the steganography technique, they would still have to require the cryptographic decoding way to de-cipher the encrypted message and vice versa [4].

The Least Significant Bit Replacement Algorithm is a commonly used straightforward steganographic algorithm used to embed secret information inside a cover medium. In this method, the least significant bits of the original data in the cover medium are altered based on the secret message. In the case of digital images, the alteration is done only at the least significant bits of the original image so as to reduce the effect of degradation of the original image. By inserting the secret message only at the least significant bits, the perceptibility of the original image is not much affected [5].

This system implements the image steganography technique of LSB replacement algorithm along with AES encryption for cryptography.

Chapter 3. System Analysis and Feasibility Study

3.1 Requirement Analysis

3.1.1 Functional Requirements

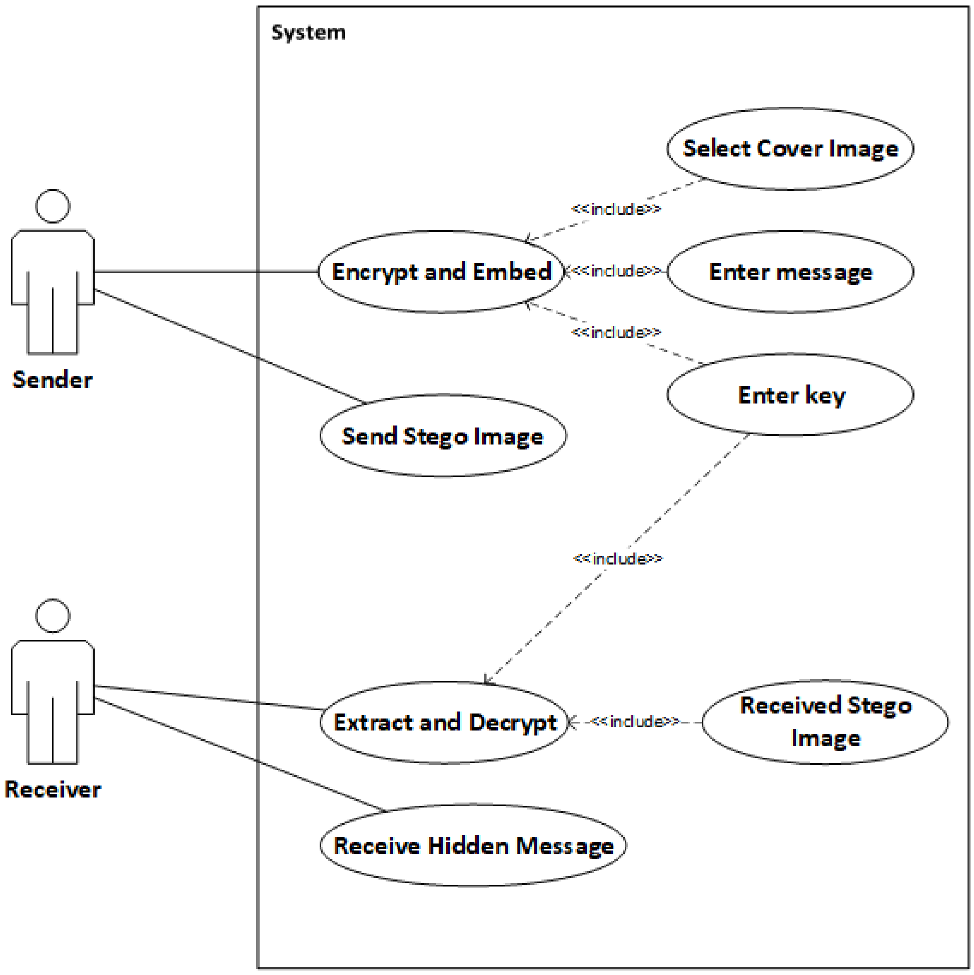


Figure 3.1 Use-Case Diagram for Image Steganography

Figure 3.1 shows the use-case diagram of the system. Sender and receiver are the primary actors. They provide required information and data to the system and receive the required output. Sender's inputs include cover image file, text message to be hidden and a secret key for message encryption. Upon receiving all valid inputs, the system generates a stego-image which consist of the text message embedded inside it. The sender can then either email the stego-image to the receiver or save the stego image to transfer it over a different media. On receiver's side, the inputs include the received stego image and associated secret key. The system processes the provided inputs and generates the hidden message in readable text. If needed, the receiver can save the generated message as text file.

In client/Server model of the system, both server and client can act as sender or receiver. They can send stego-images to one another and save or extract hidden messages upon receiving them.

3.1.2 Non-Functional Requirements

Reliability: The system program is reliable as whenever provided with correct information/input, it always gives the corresponding image (on sender’s side) or the hidden message (on receiver’s side).

Performance: The system performs well as when provided with the correct inputs, it efficiently provides correct output within seconds.

Usability: The system is quite easy to use as anyone with simple skills regarding using a windows computer can use it to create a stego image or extract the messaged hidden on a received stego image.

Maintenance: The system doesn’t really need to be maintained regularly but is open to any future enhancement, whenever required.

3.2 Feasibility Study

Feasibility Study is a test of the system according to its workability, impact of the organization, ability to meet user needs and effective use of the resources. A Feasibility Study is generic in nature and can be applied to any type of project, be it for systems and software development, making an acquisition, or any other project. We can test the system by different type of the feasibilities.

3.2.1 Economic Feasibility

The system is economically feasible as all the tools and resources required are either cheap or free.

3.2.2 Technical Feasibility

The technical requirements for the project are easily available and the system can be operated by users with simple knowledge regarding the required technologies.

3.2.3 Operational Feasibility

The system is user friendly as it is easy to use and operating the system doesn’t require too complex skills.

3.3 Process Model

3.3.1 DFD Diagram

Level 0 DFD

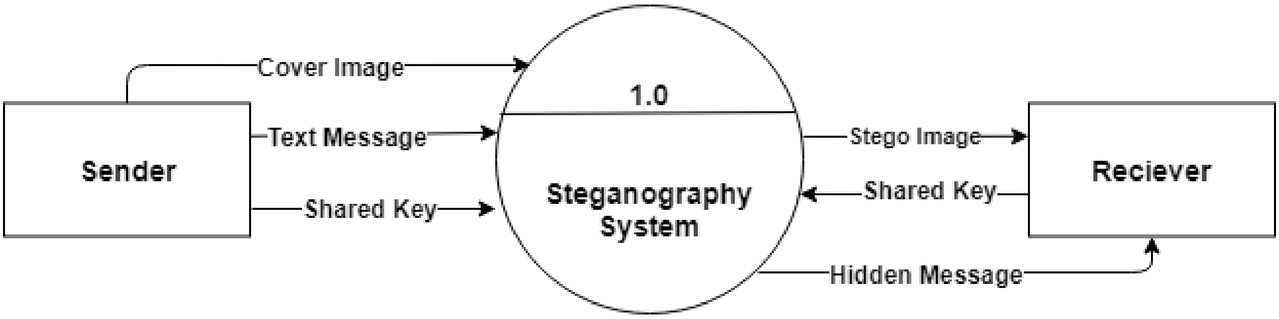


Figure 3.2 Level 0 DFD for Image Steganography

Figure 3.2 explains the Level 0 DFD of the system. The system takes the input as cover image (original image) and a secret text message from the sender. This application also uses a secret shared key for encryption. With these inputs provided by the sender, the application generates a stego image which is sent to the receiver. The system then takes the associated key from the receiver and displays the hidden message to the receiver.

Level 1 DFD

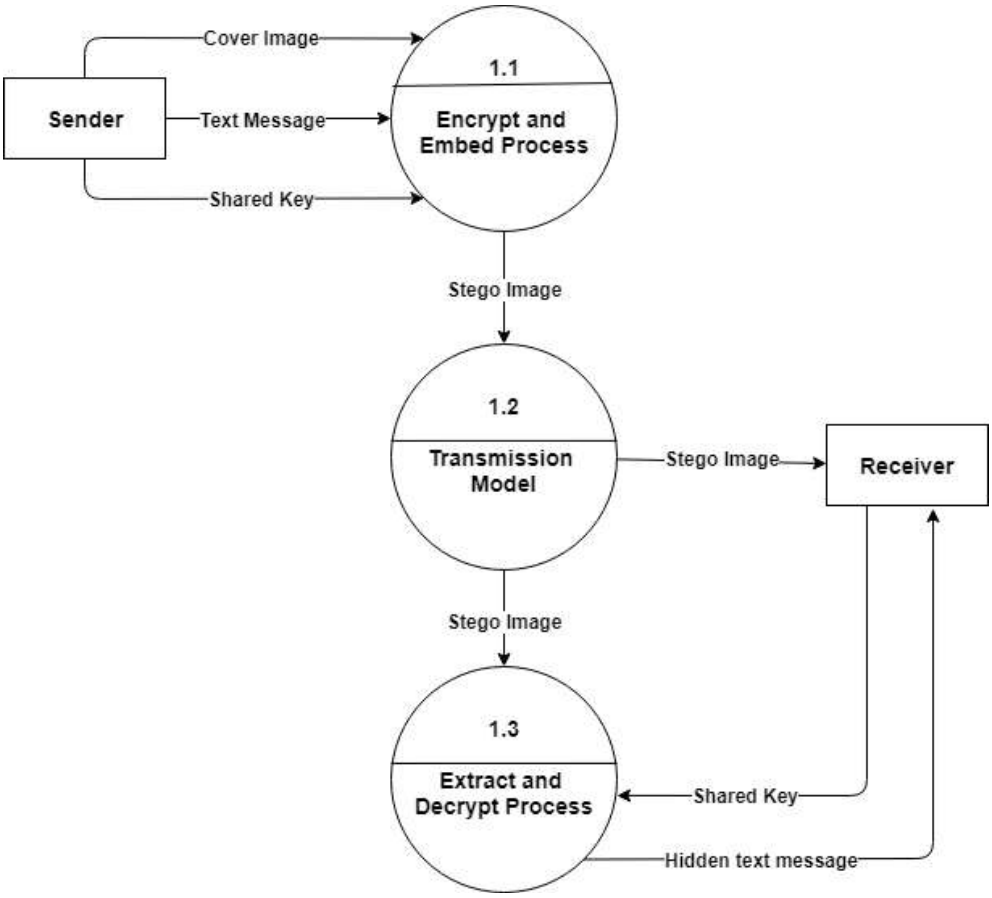


Figure 3.3 Level 1 DFD for Image Steganography

Figure 3.3 represents the Level 1 DFD of the system. In this level, more detailed process of this system is explained. The main task preformed in this level is that the text message is encrypted using AES encryption and that encrypted message is embedded to the image by replacing least significant bits of the image using LSB technique. Then the generated stego image is sent to receiver through a data transmission channel. This system then takes the stego image and shared key from the receiver and first extracts the cipher text message (encrypted hidden message) from the image and with the provided key it decrypts the cipher message and displays the original hidden message to the receiver.

Chapter 4. System Design

4.1 System Architecture

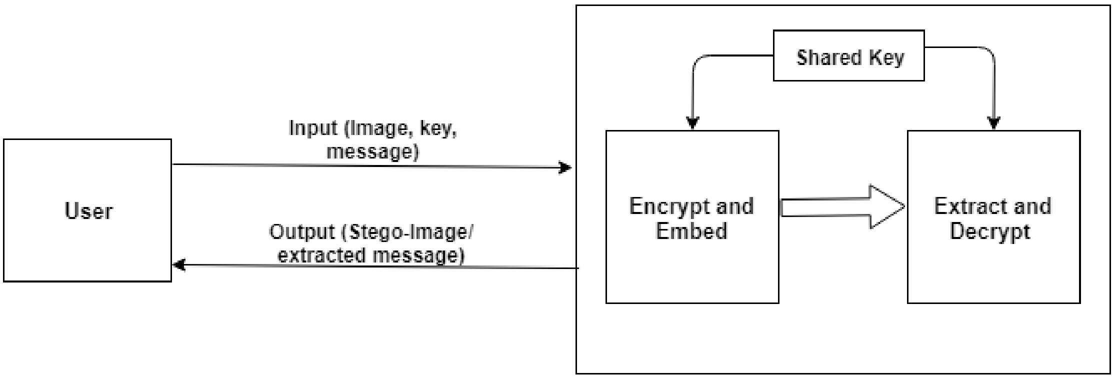

Shared Key
Input (Image, key, message)
User
Encrypt and Embed
Extract and Decrypt
Output (Stego-Image/ extracted message)

Figure 4.1 System Architecture for Image Steganography

Figure 4.1 shows the basic architecture of the system. User can interact with the system to receive expected outputs.

The system is divided into two parts: normal mode and client/server mode. When starting the program normally, the user needs to provided necessary input to get necessary output which is either stego-image (when embedding a cover image) or extracted message (when extracting a stego-image for hidden message).

When starting the program in client/server model, firstly a TCP server socket needs to be started. The user provides port number to be associated to the server. When a client connects to the server socket on particular port, the client-server connection starts. Then the client and server can communicate with each other through associated port and IP as the communication medium. For communicating sensitive information, stego-image can also be used. The sender side can generate a stego-image containing a hidden message and a key shared by both ends. Then the receiver can use the shared key to extract the hidden message from the received stego-image.

4.2 Class Diagram

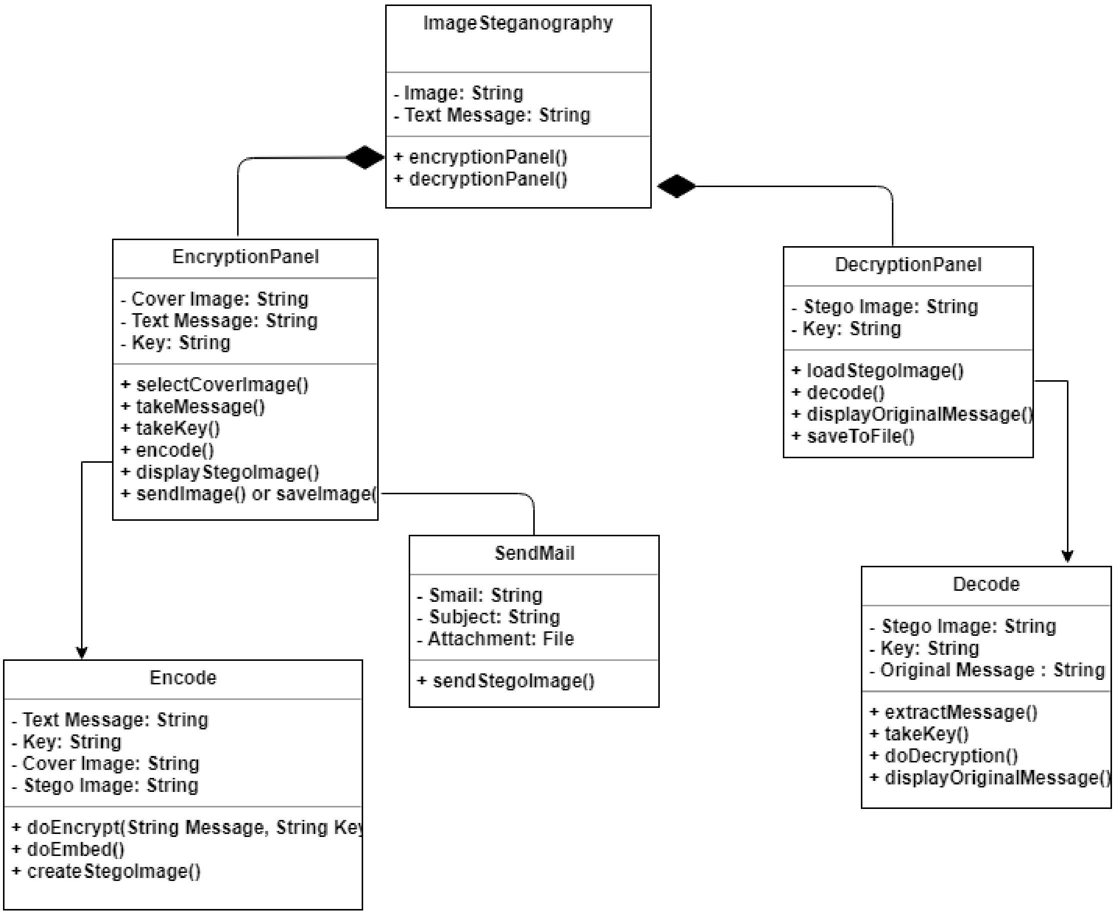


Figure 4.2 Class Diagram for Image Steganography

Figure 4.2 shows the class diagram for the system. Different classes are involved to obtain the preferred result the system. Different classes' instances are created and used whenever necessary. Some major classes involved in the system process are:

ImageSteganography: This is the main class of the system which is accessed first when starting the system. Then, preferred mode of the program can be chosen.

EncryptionPanel: This is the class that contains the panel for encrypting and embedding the cover image file. It takes cover image file, secret message and shared key as inputs, then calls necessary functions from different classes to reach the goal.

Encode: This class contains the necessary functions for encrypting the given input string and embedding the encrypted string onto provided cover image. It can be accessed from the EncryptionPanel class, which may create and use instance of this class whenever stego-image is to be created.

DecryptionPanel: This is the class that contains the panel for extracting and decrypting the hidden message from the stego image file. It takes the stego image file and shared secret key as inputs, then calls necessary functions from different classes to reach the goal.

Decode: This class contains the necessary functions for extracting the encrypted text from the stego image as well as decrypting the encrypted text to receive the original message. It can be accessed from the DecryptionPanel class, which may create and use instance of this class whenever hidden message is to be extracted.

SendMail: This class is responsible for sending the stego-image given by the Encode class, via email. It contains the frame consisting of the user interface.

4.3 Activity Diagram

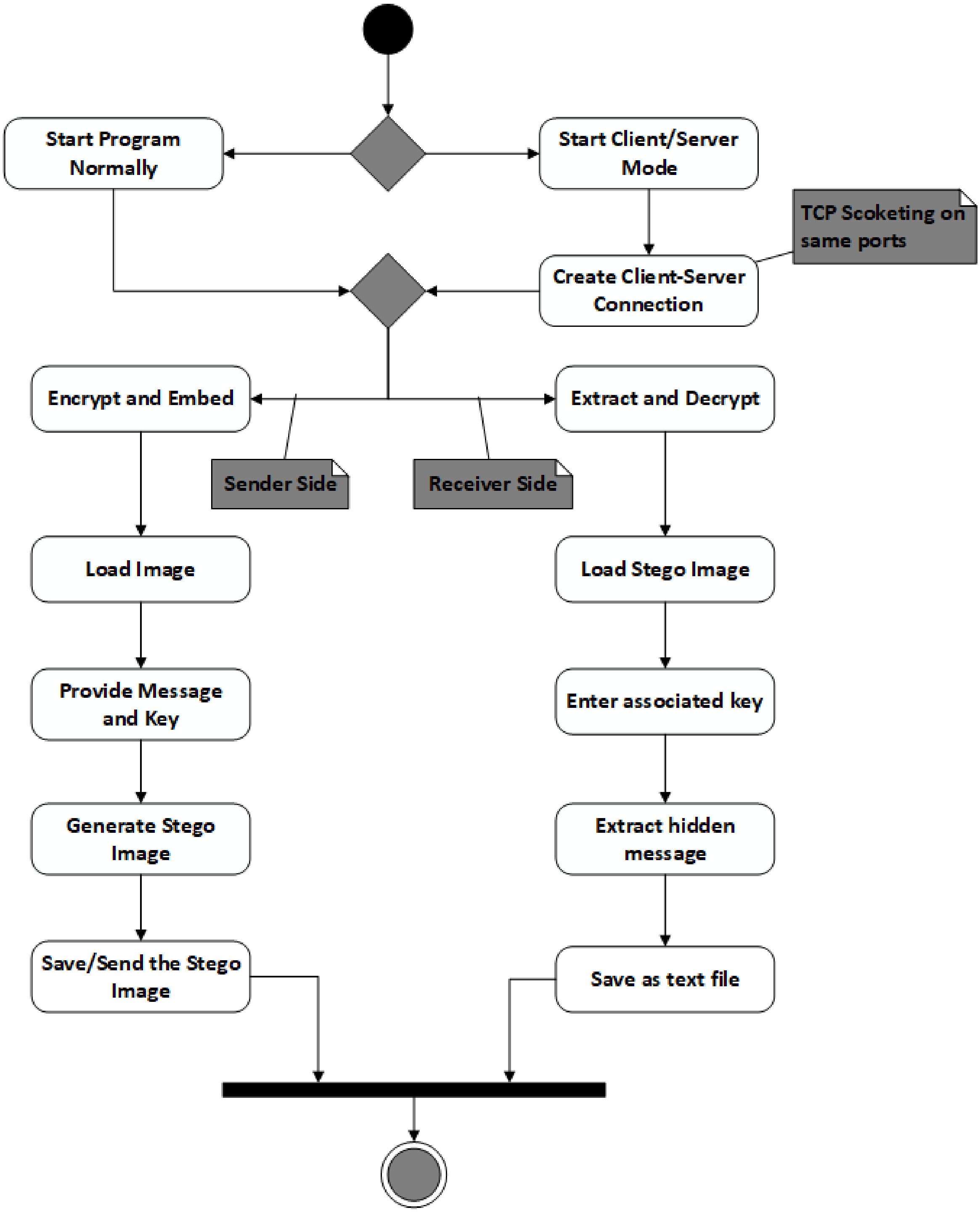


Figure 4.3 Activity Diagram for Image Steganography

Figure 4.3 explains basic activity of the system. The program can either be started in regular mode or client server mode. Either of the modes will support image steganography procedures.

The ‘encrypt and embed’ option takes a cover image, secret key and the message to be hidden as input. Then provides a stego-image as output. Similarly, the ‘extract and decrypt’ option takes the generated stego image and associated key the extracts the hidden message. Then message can be saved as a file.

In client server mode, firstly a TCP server socket needs to be started. The user provides port number to be associated to the server. When a client connects to the server socket on particular port, the client-server connection starts. Then the client and server can communicate with each other through associated port and IP as the communication medium. Sender loads the image and provide message and key. Then it generates the stego image. The stego image can be saved or send to the receiver. Receiver loads the stego image sent by the sender then enters associate key to extract the hidden message. Finally, the hidden message is extracted from stego image and can be saved as a file.

4.4 Algorithm

Least Significant Bit (LSB) Replacement Method

LSB based technique is simple approach in which message bits are embedded in the least significant bits of cover image [6]. In this technique, the least significant bit of cover image is used to hide the secret message The least significant bit of each pixel of an image is altered to a bit of a message that is to be hidden [7].

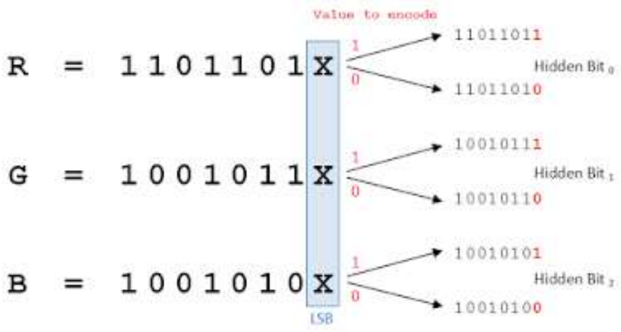


Figure 4.4 LSB Replacement

Message embedding procedure:

Input: Cover image, secret key and message.

Output: Stego image.

1. Read the cover image and secret text information which is to be embedded into the image.
2. Convert the secret information into encrypted text by using AES algorithm and secret key shared by receiver and sender.
3. Add the value of encrypted text's length at the beginning of the text along with a '/' character.
4. Convert encrypted text message into binary form – which will give the text message's characters' bits.
5. Find LSBs of each RGB pixels of the cover image.
6. Embed the bits obtained on step 4 into LSBs of RGB pixels of step 4.
7. Continue the procedure until the secret information is fully hidden in cover image file.

Message extraction procedure:

Input: Stego Image and associated secret key.

Output: Hidden message.

1. Read the stego image and secret key.
2. Retrieve LSBs of each RGB pixels of the stego image.
3. Convert binary strings formed by every 8 RGB pixels of step 2 to character and append the characters to a string builder.
4. Upon finding the first '/' character from string builder of step 3, save its previous characters as text length and discard all characters till that index.
5. Continue the process for more (text length obtained at step 4 * 8) times to fully extract the hidden encrypted text.
6. Using the secret key and AES decryption, decrypt secret information obtained on step 4 to get original information.

Chapter 5. Implementation and Testing

5.1 Tools Used

While many additional software tools were used when developing the project, there was no need for any special hardware requirement. The system was built on a 64-bit computer running with Windows 10. Various Software tools used include:

- IDE: NetBeans 8.0.2.
- Programming Language: Java programming language.
- Swing- GUI widget toolkit for java: Used for creating GUI.
- Java AWT packages: For event handling and working with files and images.
- Java Mail bean: Used to send stego image via Email.
- Java AES encryption/decryption API: Used to generate cipher text.
- Additionally, various java packages and libraries were also used.

5.2 Testing

5.2.1 Unit Testing

The modules used in this system are:

- Message encryption and embedding module:

Table 5-1 Unit Testing: Message encryption and embedding

Test Cases	Input	Expected Output	Output	Remarks
String encryption	Message string, key string	Encrypted string	Encrypted string	Encrypted successfully
Image's LSB replacement	Image, any string	Image with changed LSBs	Image with changed LSBs	String embedded to image successfully

- Emailing process module:

Table 5-2 Unit Testing: Emailing process

Test Cases	Input	Expected Output	Output	Remarks
Email Send	An image file, email attributes	Image file sent	Authentication error	Error because of gmail's security issue
Email Send	An image file, correct email attributes	Image file sent	Image file sent	Image sent successfully via email

- Message extraction and decryption module:

Table 5-3 Unit Testing: Message extraction and decryption

Test Cases	Input	Expected Output	Output	Remarks
String Extraction from Image	Random image file	Encrypted string	Input error	Image not a valid stego file
String Extraction from Image	Stego image file	Encrypted string	Encrypted string	Encrypted string extracted successfully
Decryption of encrypted string	Encrypted string, associated key string	Decrypted string	Decrypted string	String decrypted successfully

5.2.2 Integration Testing

The test cases referenced in Table 5.1 were integrated together and tested for output. For the encryption and LSB replacement process an image file, a message and a key were provided inputs, then a stego image was generated as expected, and this stego image was saved and sent concurrently for testing. The image was sent via email and thus the test case of Table 5.2 was successfully integrated.

Also, the message extraction and decryption process, as referenced on Table 5.3, were tested as a whole. At first, a random image and a key were provided as inputs, and this produced an error. Later the generated stego image and associated key was provided as inputs, and the hidden message was extracted from the image as excepted.

Table 5-4 Integration testing for encryption and embedding module

Test Cases	Input	Expected Output	Output	Remarks
Encryption and LSB replacement process integration	Message String, Key String, image file	Stego image file	Stego image file	Message successfully hidden in image

Table 5-5 Integration testing for message extraction and decryption module

Test Cases	Input	Expected Output	Output	Remarks
Message Extraction and Decryption integration	Random image, key string	Hidden message in decrypted form	Input error	Invalid image or key
Message Extraction and Decryption integration	Stego image, associated key	Hidden message in decrypted form	Hidden message in decrypted form	Hidden message extracted successfully

5.2.3 System Testing

Firstly, to test the message hiding process, the encrypt and embed option was chosen. A cover image file, a key and a message to be hidden were provided as inputs. After being provided with valid inputs, the system was successfully able to generate a stego image. After receiving the stego image file, it was saved to the hard drive. Sending the generated stego image file over email was also successfully tested. Sending the image over the internet required inputs such as sender's email and password as well as the receiver's email and password. The provided email attributes need to be correct and the sender's id should have permission to send emails via windows applications.

After the successful generation of the stego image, it was tested for message extraction. To test the image sent over the email, the image was firstly downloaded using the option provided by Gmail. After choosing the extract and decrypt option in the program, the stego image and associated key were provided as inputs. The program was able to extract the hidden message from the stego image in decrypted form. The hidden message was displayed on the screen. After receiving the hidden message, it was saved to the hard drive as text file.

Chapter 6. Conclusion and Future Enhancement

6.1 Conclusion

The final product of the project is a system that can take a cover image file, hidden message and a secret key as input and provide a stego-image as output as well as can extract the secret message hidden in the stego-image when provided with a key and the corresponding stego image. On sender’s side, the hidden message is first encrypted with AES technique with the help of provided secret key and the encrypted text is then embedded onto the cover image file to produce stego-medium. The embedding process follows image’s LSB replacement algorithm. On receiver’s side the process includes extraction of the encrypted message then its decryption with original key for obtaining the original message. During extraction of hidden cipher text, the bits at LSB are extracted and finally through a series of steps, converted to the encrypted text.

6.2 Future Enhancement

If needed, the system will be open for future enhancements. The future enhancements might include audio/video steganography with improved algorithms.

References

[1] S. Singh, Literature Review On Digital Image Steganography and Cryptography Algorithms, 2015/07/01.

[2] A. Soni, J. Jain and R. Roshan, "Image steganography using discrete fractional Fourier transform," *Intelligent Systems and Signal Processing (ISSP),* 2013.

[3] N. Akhtar, P. Johri and S. Khan, "Enhancing the Security and Quality of LSB Based Image Steganography," *Computational Intelligence and Communication Networks (CICN),* 2013.

[4] I. V. S. Mano, "A New LSB-S Image Steganography Method Blend with Cryptography for Secret Communication," in *Third International Conference on Image Infonnation Processing*, 2015.

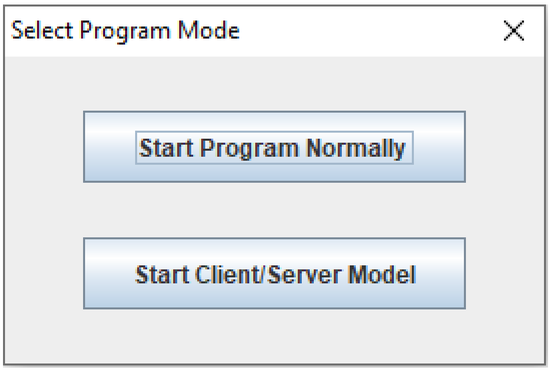
[5] S. Sugathan, "An Improved LSB Embedding Technique for Image Steganography," in *2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, 2016.

[6] M. R. Garg, "Comparison Of Lsb & Msb Based Steganography In Gray-Scale Images," *International Journal of Engineering Research and Technology (IJERT),* vol. 1, no. 8, 2012.

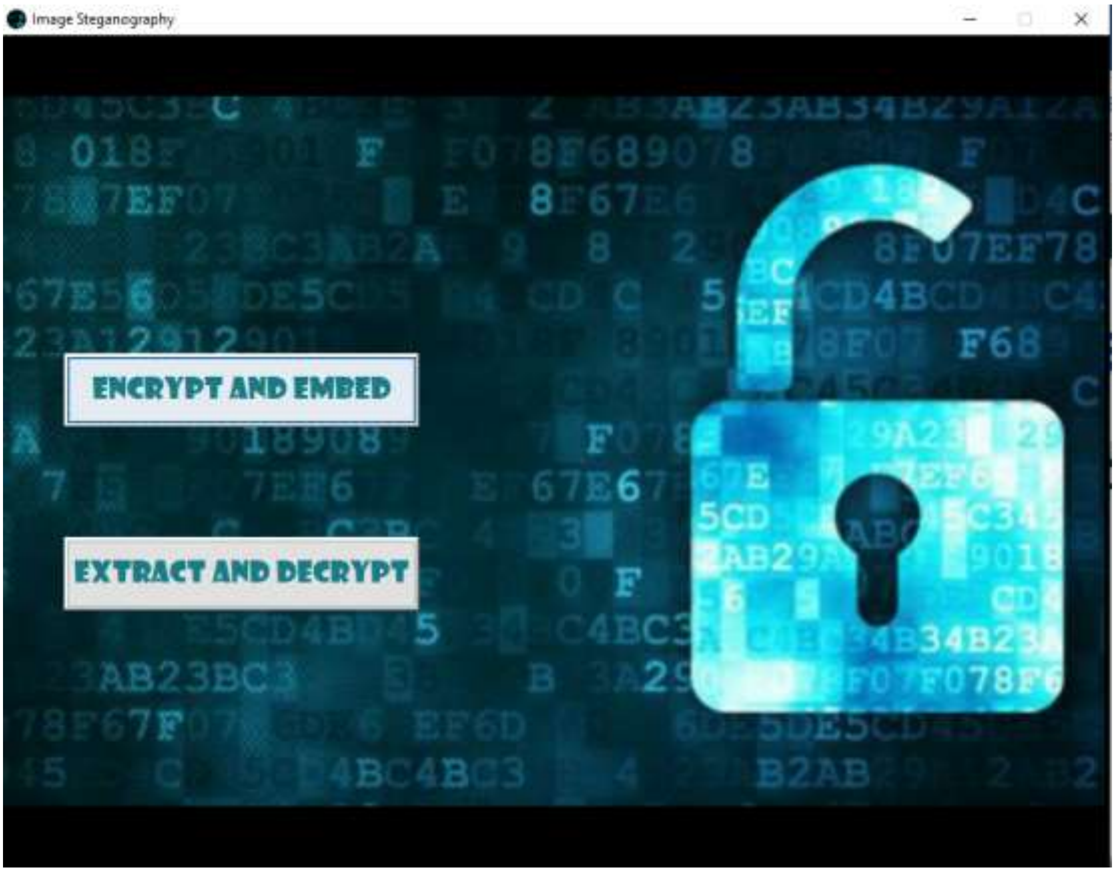
[7] P. J. N. Dr. Ekta Walia, "“An Analysis of LSB & DCT based Steganography," *Global Journal of Computer Science and Technology,* vol. 10, no. 1, 2010.

Appendix

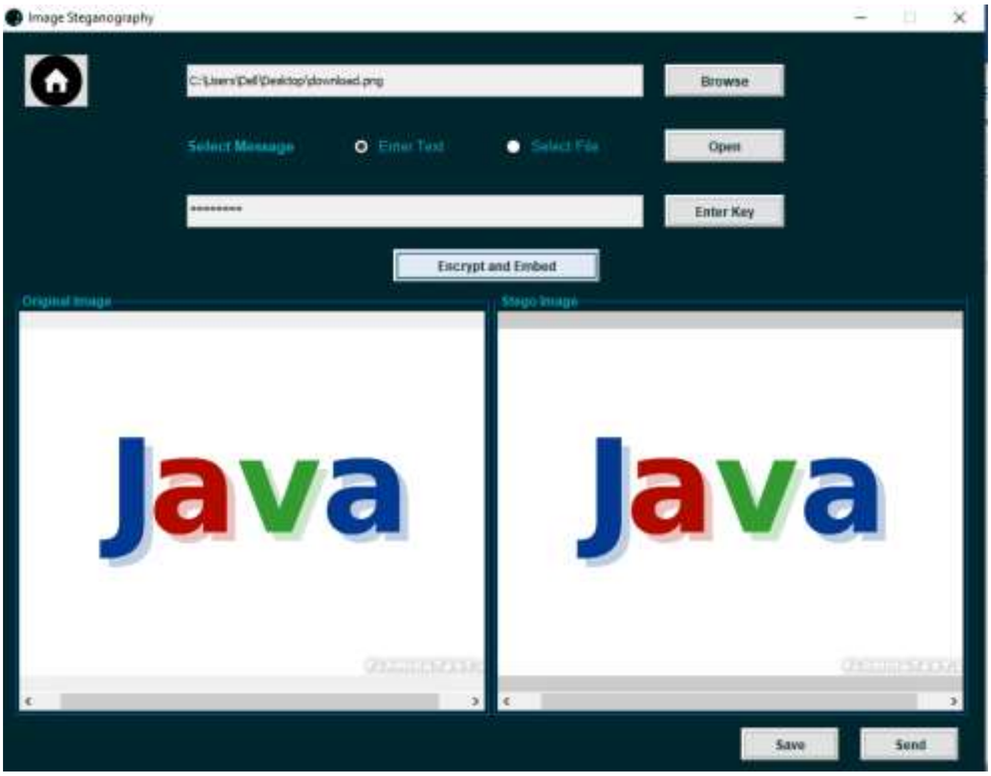
Screenshots



Screenshot 1: Program mode selection



Screenshot 2: Normal mode user Interface

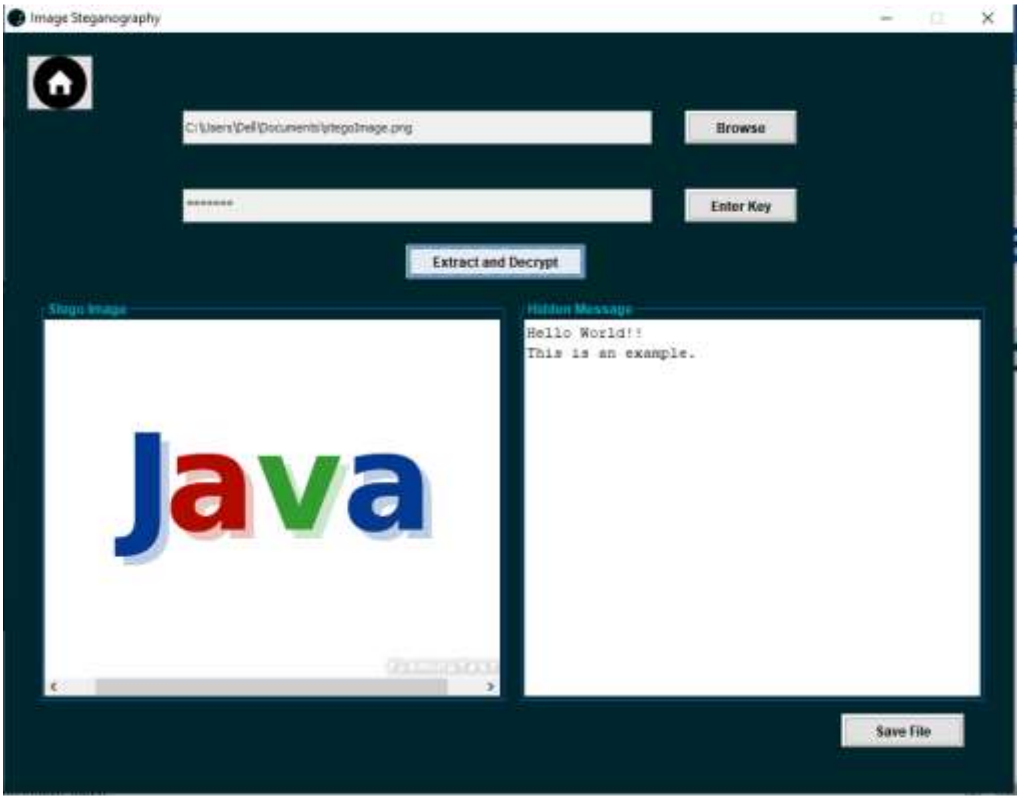


C:\Users\Dell\Desktop\download.png

Select Message ○ Enter Text ● Select File

****

Screenshot 3: "Encrypt and Embed" screen

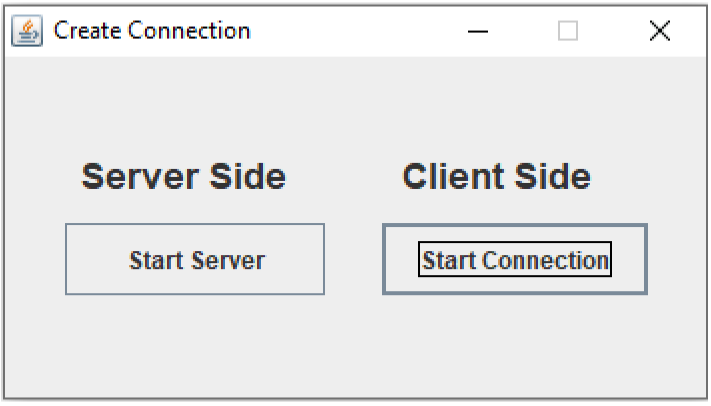


C:\Users\Dell\Documents\stegoImage.png

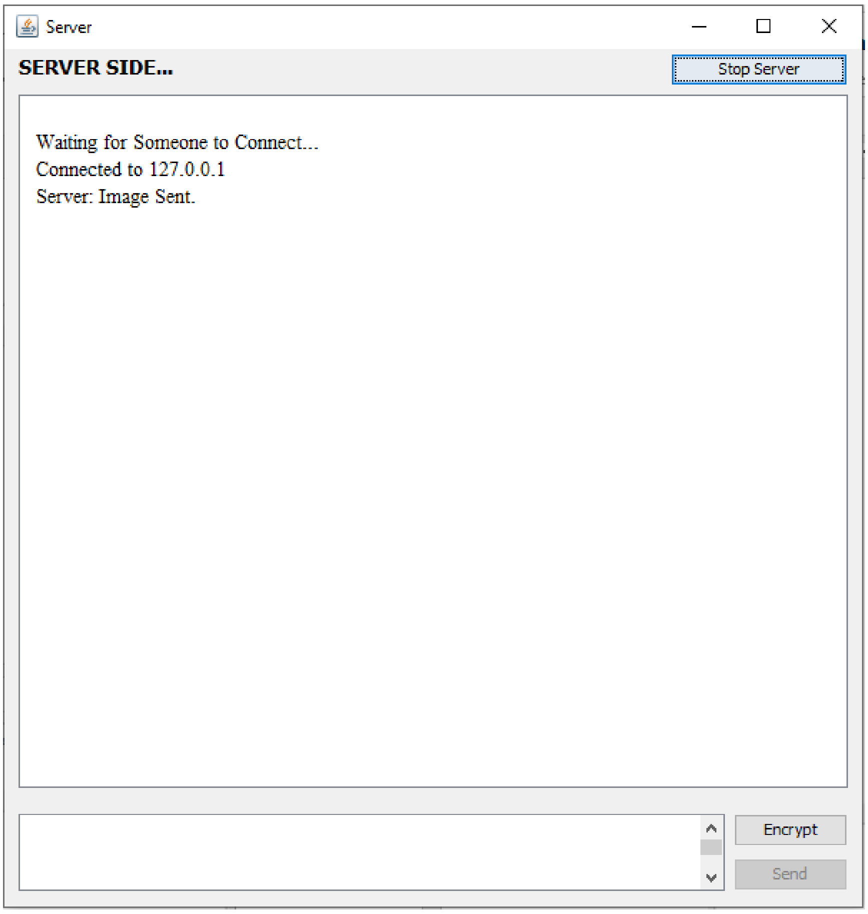
****

```
Hello World!!
This is an example.
```

Screenshot 4: "Extract and Decrypt" screen

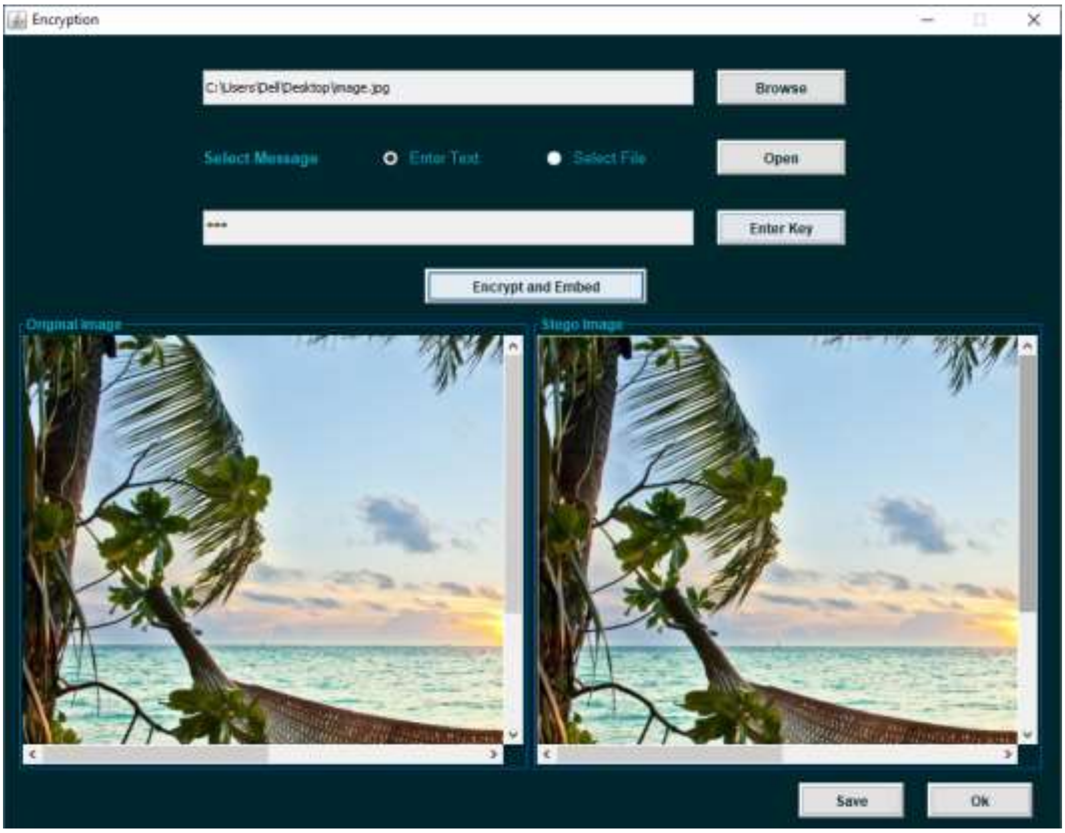


Screenshot 5: Client/Server mode: User Interface



```
Waiting for Someone to Connect...
Connected to 127.0.0.1
Server: Image Sent.
```

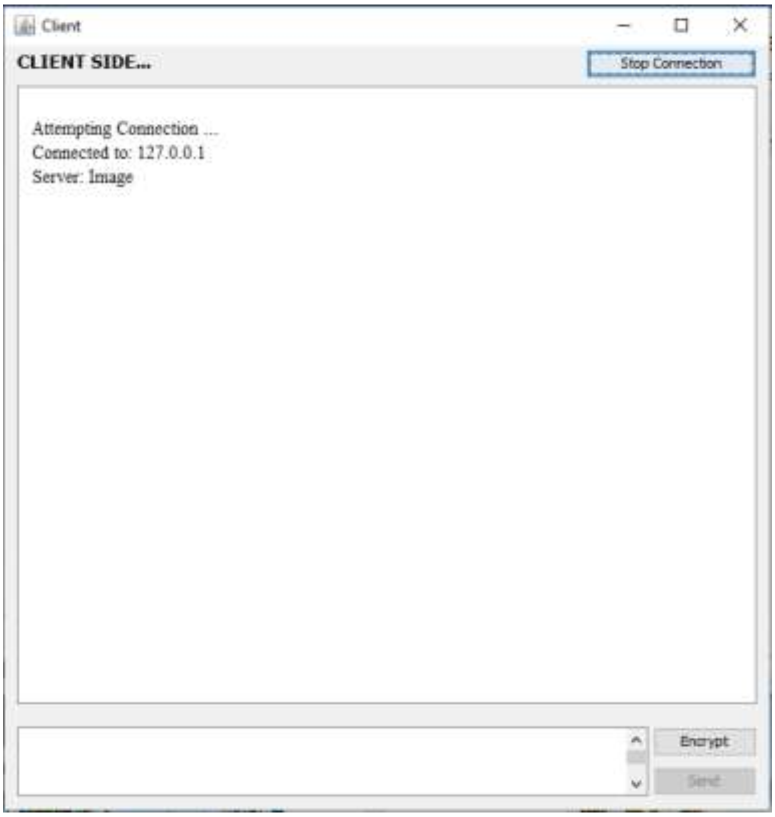
Screenshot 6: Server side



C:\Users\Dell\Desktop\image.jpg

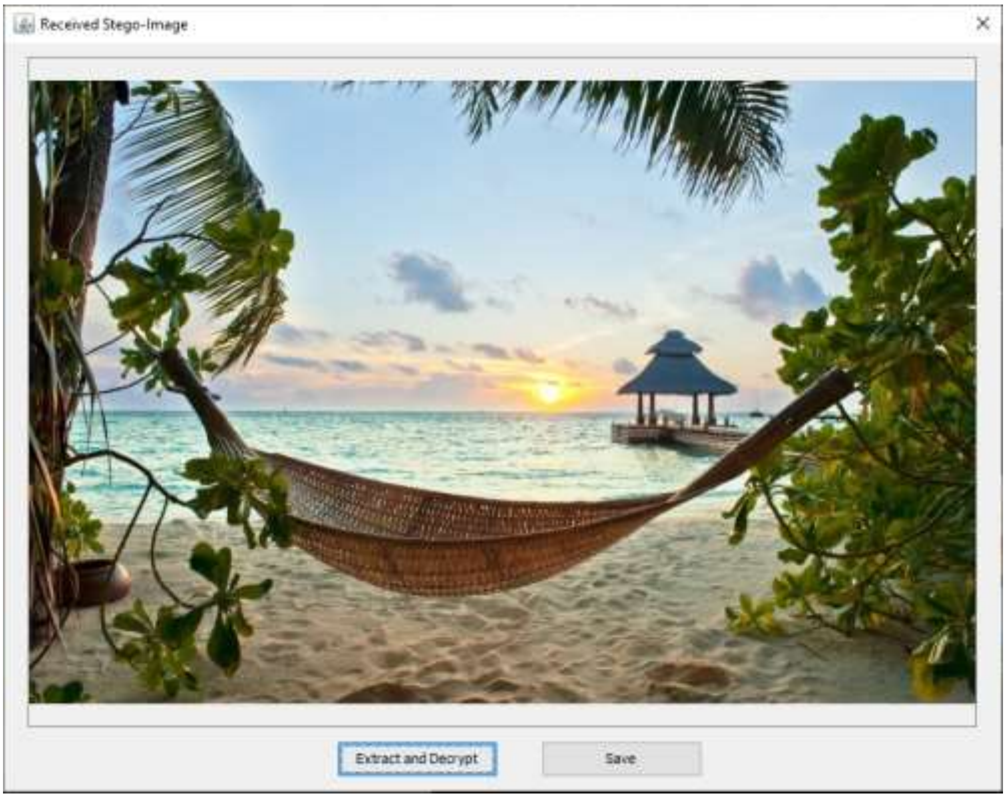
***

Screenshot 7: Encryption window at Sender (server) side

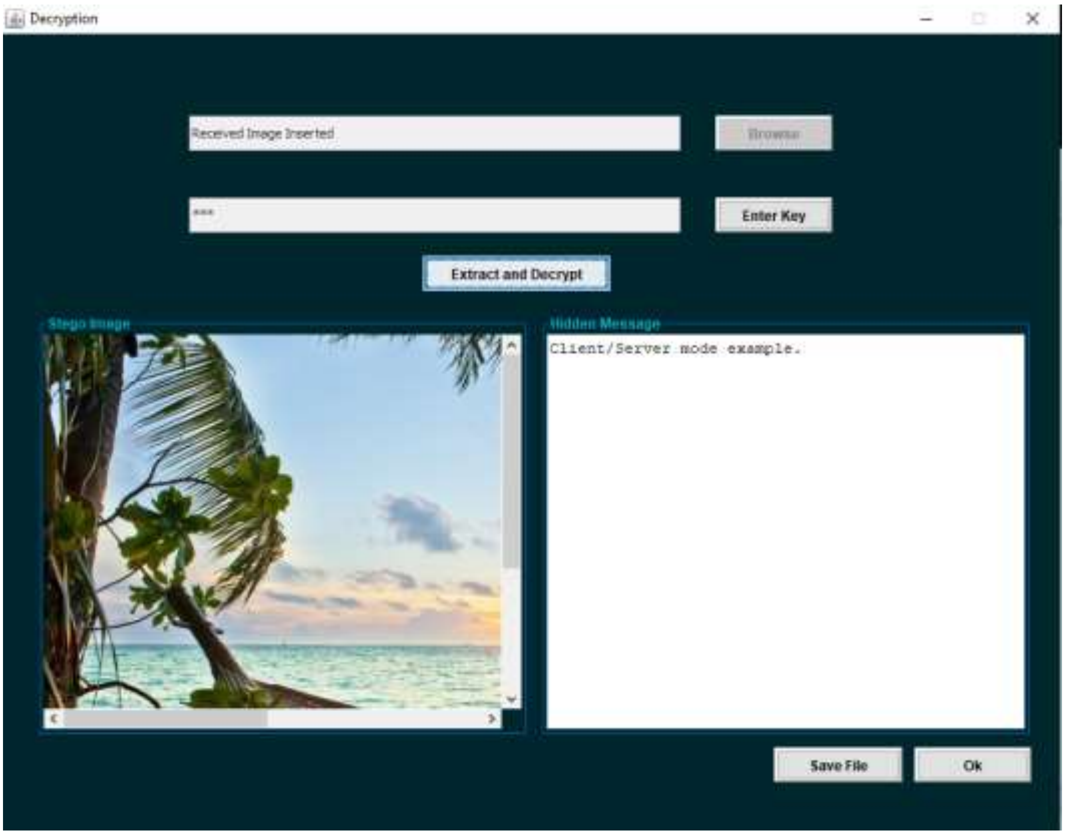


```
Attempting Connection ...
Connected to: 127.0.0.1
Server: Image
```

Screenshot 8: Client side



Screenshot 9: Automatic image viewer at receiver (client) side



Received Image Inserted

***

```
Client/Server mode example.
```

Screenshot 10: Hidden message extraction at receiver side