# 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

[ACKNOWLEDGEMENT I](#_Toc12489369)

[ABSTRACT II](#_Toc12489370)

[Table of Contents III](#_Toc12489371)

[List of Figures V](#_Toc12489372)

[List of Tables V](#_Toc12489373)

[Abbreviations VI](#_Toc12489374)

[Chapter 1. Introduction VII](#_Toc12489375)

[1.1 Introduction to Image Steganography VII](#_Toc12489376)

[1.2 Problem Statement VIII](#_Toc12489377)

[1.3 Objective VIII](#_Toc12489378)

[1.4 Scope and Limitations IX](#_Toc12489379)

[1.5 Report Organization IX](#_Toc12489380)

[Chapter 2. Literature Review X](#_Toc12489381)

[Chapter 3. System Analysis and Feasibility Study XI](#_Toc12489382)

[3.1 Requirement Analysis XI](#_Toc12489383)

[3.1.1 Functional Requirements XI](#_Toc12489384)

[3.1.2 Non-Functional Requirements XII](#_Toc12489385)

[3.2 Feasibility Study XIII](#_Toc12489386)

[3.2.1 Economic Feasibility XIII](#_Toc12489387)

[3.2.2 Technical Feasibility XIII](#_Toc12489388)

[3.2.3 Operational Feasibility XIII](#_Toc12489389)

[3.3 Process Model XIV](#_Toc12489390)

[3.3.1 DFD Diagram XIV](#_Toc12489391)

[Chapter 4. System Design XVI](#_Toc12489392)

[4.1 System Architecture XVI](#_Toc12489393)

[4.2 Class Diagram XVII](#_Toc12489394)

[4.3 Activity Diagram XIX](#_Toc12489395)

[4.4 Algorithm used XX](#_Toc12489396)

[Least Significant Bit (LSB) Replacement Method XX](#_Toc12489397)

[Message embedding procedure: XXI](#_Toc12489398)

[Message extraction procedure: XXI](#_Toc12489399)

[Chapter 5. Implementation and Testing XXII](#_Toc12489400)

[5.1 Tools Used XXII](#_Toc12489401)

[5.2 Testing XXII](#_Toc12489402)

[5.2.1 Unit Testing XXII](#_Toc12489403)

[5.2.2 Integration Testing XXIII](#_Toc12489404)

[5.2.3 System Testing XXIII](#_Toc12489405)

[Chapter 6. Conclusion and Future Enhancement XXIV](#_Toc12489406)

[6.1 Conclusion XXIV](#_Toc12489407)

[6.2 Future Enhancement XXIV](#_Toc12489408)

[References XXV](#_Toc12489409)

# List of Figures

[Figure 3.1.1: Use Case Diagram for Image Steganography 11](file:///E:\Project\Documentation\Final.docx#_Toc12487139)

[Figure 3.3.1: Level 0 DFD 14](file:///E:\Project\Documentation\Final.docx#_Toc12487140)

[Figure 3.3.2: Level 1 DFD 14](file:///E:\Project\Documentation\Final.docx#_Toc12487141)

[Figure 4.1.1: System Architecture 16](file:///E:\Project\Documentation\Final.docx#_Toc12487142)

[Figure 4.3.1: Activity Diagram for Image Steganography 19](file:///E:\Project\Documentation\Final.docx#_Toc12487143)

[Figure 4.4.1 LSB replacement 20](file:///E:\Project\Documentation\Final.docx#_Toc12487144)

# List of Tables

Table 5.2‑1: Unit Testing 23

Table 5.2‑2: Integrated Testing 23

# Abbreviations

AES: Advanced Encryption Standard 2, 9

API: Application Program Interface 22

DFD: Data Flow Diagram 9

ICT: Information and Communication Technology 7

IDE: Integrated Development Environment 22

IP: Internet Protocol 16, 20

LSB: Least Significant Bit 9

RGB: Red Green Blue 21

TCP: Transmission Control Protocol 16, 20

UI: User Interface 24

# Introduction

## 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.

## Problem Statement

No matter how large or small a company is, there is a need to have a plan to ensure the security of the information assets. 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.

But, as mentioned earlier, with the increasing use of communication of information over digital medium, security for digital methods are to be developed. Furthermore, the data hidden inside the stego medium needs to be encrypted rather than in its original form. This ensures even more data security.

## 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.

## 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.

## 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.

# 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.

1. System Analysis and Feasibility Study

## Requirement Analysis

### Functional Requirements

**Figure 3.1.1: Use Case Diagram for Image Steganography**

Figure 1 shows the use-case diagram of the system. Sender and receiver are the primary actor. 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 generate the hidden message in text form. 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. Along with text messages, they can also send stego-images to one another and save or extract messages upon receiving them.

### 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.

**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, if needed.

## 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.

### Economic Feasibility

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

### 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.

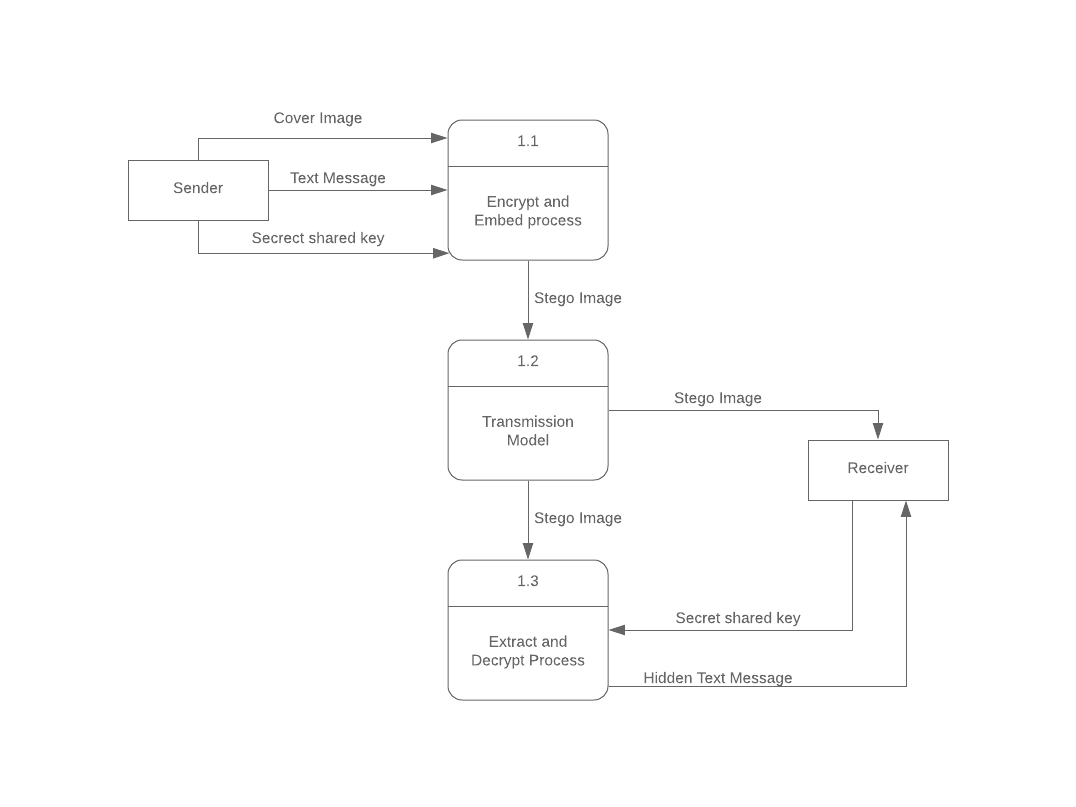
### Operational Feasibility

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

## Process Model

### DFD Diagram

**Figure 3.3.1: Level 0 DFD**



**Figure 3.3.2: Level 1 DFD**

Figure 3.3.1 explains the Level 0 DFD of the system. The proposed model 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.

Figure 3.3.2 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.

1. System Design

## System Architecture

**Figure 4.1.1: System Architecture**

Figure 3.3.1 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.

## Class Diagram

**Figure 4.2.1: Class Diagram for Image Strganography**

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.

## Activity Diagram

**Figure 4.3.1: Activity Diagram for Image Steganography**

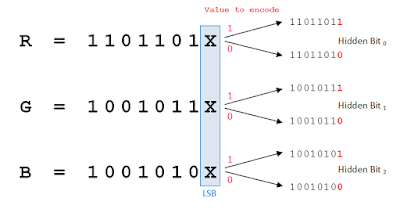
Figure 4.3 explains basic activity of the system. The program can either be started in regular mode or client server mode. Either of the modes will support image steganography procedures.

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.

## Algorithm used

### 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 4.4.1 LSB replacement**

### Message embedding procedure:

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:

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.

# Implementation and Testing

## Tools Used

While many additional softwares 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.

## Testing

### Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Cases** | **Input** | **Expected Output** | **Output** | **Remarks** |
| String encryption | Message string, key string | Encrypted string | Encrypted string | Encrypted successfully |
| Image LSB replacement | Image, any string | Image with changed LSBs | Image with changed LSBs | String embedded to image successfully |
| String Extraction from Image | Random image file | Encrypted string | Input error | Image not a stego Image |
| 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 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Cases** | **Input** | **Expected Output** | **Output** | **Remarks** |
| 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 |

**Table 5.2‑1: Unit Testing‑**

### Integration Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |
| 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 |

**Table 5.2‑2: Integrated Testing‑**

### System Testing

The system is successfully able to generate a stego image file when provided with a cover image, a message string and a key string. Sending the generated stego image file over email is also possible. But the provided email attributes need to be correct and have permission to send emails via windows applications. The system is also able to extract the hidden message from generated stego image in decrypted form, when provided with a stego-image and associated key.

1. Conclusion and Future Enhancement

## 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.

## 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. |