A Minor Project Mid-Term Report on

Data Security through Steganography

by using LSB Replacement Algorithm

Submitted in Partial Fulfillment of the Requirements for

Bachelor of Engineering in **Software Engineering**

under **Pokhara University**

Submitted by:

Sanjaya Adhikari, 14731

Mithun Adhikari, 14748

Date: 21-01-2018

Department of Software Engineering

NEPAL COLLEGE OF

INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal

A Minor Project Mid-Term Report on

Data Security through Steganography

by using LSB Replacement Algorithm

Submitted in Partial Fulfillment of the Requirements for

Bachelor of Engineering in **Software Engineering**

under **Pokhara University**

Submitted by:

Sanjaya Adhikari, 14731

Mithun Adhikari, 14748

Date: 21-01-2018

Department of Software Engineering

NEPAL COLLEGE OF

INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal

ACKNOWLEDGEMENT

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us during this project. We owe special thanks to a number of people who has devoted much of their time and expertise without which it would have been very difficult for us to complete our project entitled "**Data Security through Steganography**".

We would like to take the opportunity to express our humble gratitude to our supervisor **Mr. Roshan Kumar Shah**, Assistant Professor, NCIT, under whom we undertook our project. His constant guidance and willingness to share his vast knowledge made us enhance our knowledge and helped us to complete the assigned tasks to perfection. Without his effort and full support & an astonishing testing ability this project may not have succeeded.

Moreover, we would like to extend our special thanks to **Dr. Roshan Chitrakar** for providing technical support with suggestions which helped our project to foster. It would be our privilege to thank **Er. Birendra Bista**, Head of Department of Software Engineering for his support. We would also like to thank **Er. Tulsi Dahal** for his valuable suggestion. We are thankful to our college, **NCIT** that provided us an opportunity to apply our theoretical knowledge through the project. We feel obliged in submitting this project as part of our curriculum

Although, there may be still many who are unacknowledged in this humble vote of thanks, there are none who remain unappreciated.

Sanjaya Adhikari

Mithun Adhikari

Nepal College of Information Technology,

Balkumari, Lalitpur, Nepal

ABSTRACT

The rapid development of data transfer through internet made it easier to send the data accurate and faster to the destination. There are many transmission media to transfer the data to destination like e-mails; at the same time, it is may be easier to modify and misuse the valuable information through hacking. So, in order to transfer the data securely to the destination without any modifications, there are many approaches like cryptography and steganography.

Steganography is the process of hiding one file inside another such that others can neither identify the meaning of the embedded object, nor even recognize its existence. Current trends favor using digital image files as the cover file to hide another digital file that contains the secret message or information.

This paper presents to provide the transfer of secret data embedded into master file to obtain new image, which is practically indistinguishable from the original image, so that other than the indeed user, cannot detect the presence of the secrete data sent. Here we use the Least Significant Bit (LSB) algorithm.

Keywords: Steganography, Steganalysis, Information Security.

LIST OF ABBREVIATIONS

NPM	Node Package Manager
GUI	Graphical User Interface
DFD	Data Flow Diagram
SD	Sequence Diagram
DCD	Design Class Diagram
UML	Unified Modeling Language
BMP	Bitmap
PNG	Portable Network Graphics
FDD	Features Driven Development

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1 INTRODUCTION

Our project ‘Data Security using Steganography’ is cross platform native mobile application. The term ‘Data’ here can be further decompose into either images or message itself. By using steganography, we can secure the message by hiding it into the career image or can secure the image by hiding the label to identify the image if the image gone leaked. We focused on the Image security in this project.

The word steganography is derived from the Greek words *stego* meaning cover and *grafia* meaning writing defining it as covered writing [1]. 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 original files can be referred to as cover text, cover image, or cover audio. After inserting the secret message, it is referred to as stego-media. A stego-key is used for hiding/encoding process to restrict detection or extraction of the embedded data [2].

1.1 PROBLEM STATEMENT

This project attempts to make use of the standard steganographic tool and a modified version of standard encryption algorithm to perform the task of data hiding for the purpose of observing privacy. The user should be able to conceal a text message in an image file without any visible alterations to the image as such i.e. there should not be any noticeable changes to the coloring or the position of the various objects in the image.

As we are hiding the data into the image for data security, the main problem is to selecting the best suitable algorithm from different available algorithms. The other problem in the hiding information or steganography is the size of data that user want to embed inside the image file. The most common method for hiding information in the image is LSB, LSB is efficient yet hard to analysis, however, it is not effective in term of the data hidden quantity, all researchers agreed the fact that the size of data hidden is a problem in that particular area, the other problem that faced there, in fact if we try to increase the quantity of data in the image there will be a suspect changes which become clear to human eyes, for instance, this research will face a challenge that high rate data hidden without affecting the images quality.

In short, the main problems in the steganography are as follow:

- Adjusting the quality of image from Human Visual System
- Successfully encrypting the data with symmetric key
- The size of data hidden
- Level of data protection
- The level of suspicion

1.2 PROJECT OVERVIEW

One of the reason that intruders can be successful is that most of the information they acquire from a system is in a form that they can read and comprehend. Intruders may reveal the information to others, modify it to misrepresent an individual or organization, or use it to launch an attack. One solution to this problem is, through the use of steganography. Steganography is a technique of hiding information in digital media. 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 is the art of hiding the fact that communication is taking place, by hiding information in other information. By using it, data can be hidden into the career data. We have used steganography to secure the data[3].

1.3 PROJECT OBJECTIVES

In this project, we primarily concentrated on the data security issues when sending the data over the network using steganographic techniques. The main objectives of the project are:

1. To create a tool that can be used to hide data inside a 24-bit color image.
2. The tool should be able to encrypt the message with symmetric key and produce cipher text before embedding it.
3. To display the hidden message successfully from the cover image if and only if private key is matched.
4. The tool should be easy to use, and should use a graphical user interface.

5. The tool should work cross-platform.

1.4 PROJECT SCOPE AND LIMITATIONS

The scope of the project is to limit unauthorized access and provide better security during message transmission. To meet the requirements, we use the simple and basic approach of steganography. In this project, the proposed approach finds the suitable algorithm for embedding the data in an image using steganography which provides the better security for sending messages through a network.

The scope of the study encompasses:

1. This research will focus on hiding data in image
2. This research will focus on extracting the secret message from stego-image.

The limitations of the project are listed below:

1. Only bitmap file format is allow as input/output image.
2. Only 24-bit color depth is allow as input image.

1.5 SIGNIFICANCE OF STUDY

The study investigates the need of system i.e. android and iOS –based steganography tool that provide the users with a platform to hide the text into the image. The current project aims to use steganography for an image with text using spatial domain technique. This hidden information can be retrieved only through proper decoding technique.

We have studied the research paper found on the internet which are listed in the bibliography entitle.

To overcome the problems listed above, we use following techniques:

We used steganography to maintain the quality of the image from

2 LITERATURE REVIEW

2.1 REVIEW

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 because of their frequency on the Internet. For hiding secret information in images, there exists a large variety of steganographic techniques some are more complex than others and all of them have respective strong and weak points. Different applications have different requirements of the steganography technique used. For example, some applications may require absolute invisibility of the secret information, while others require a larger secret message to be hidden. This project intends to give an overview of image steganography, its uses and techniques. It also attempts to identify the requirements of a good steganographic algorithm and briefly reflects on which steganographic techniques are more suitable for which applications

2.2 BASIC STRUCTURE OF STEGANOGRAPHY

Steganography is the practice of hiding private or sensitive information within something that appears to be nothing out to the usual. The basic structure of Steganography is made up of three components:

1. **Carrier -** The carrier can be a painting, a digital image, an mp3, even a TCP/IP packet among other things. It is the object that will ‘carry’ the hidden message.
2. **Message -** The message (hidden) is being carried by the object (carrier).
3. **Key -** A key is used to decode/decipher/discover the hidden message. This can be anything from a password, a pattern, a black-light, or even lemon juice.

Basically, the model for steganography is shown in Figure 1. Message is the data that the sender wishes to remain it confidential. It can be plain text, cipher text, other image, or anything that can be embedded in a bit stream such as a copyright mark, a covert communication, or a serial number. Password is known as *stego-key*, which ensures that only recipient who knows the corresponding decoding key will be able to extract the message from a *cover-object*. The *cover-object* with the secretly embedded message is then called the *stego-object.*

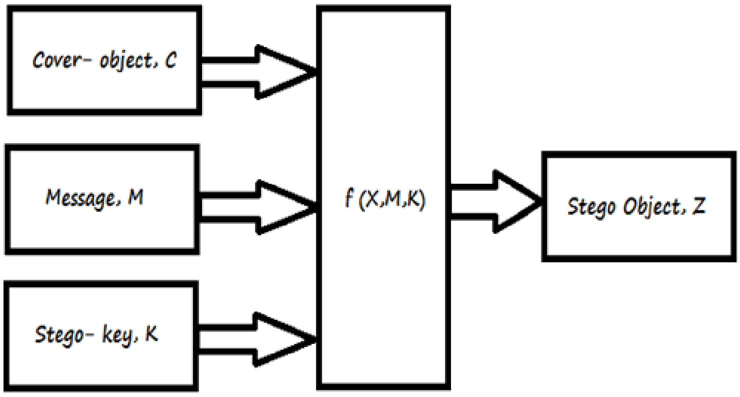


Figure 1: Basic Steganography Model

Recovering message from a *stego-object* requires the *cover-object* itself and a corresponding decoding key if a *stego-key* was used during the encoding process. The original image may or may not be required in most applications to extract the message.[3]

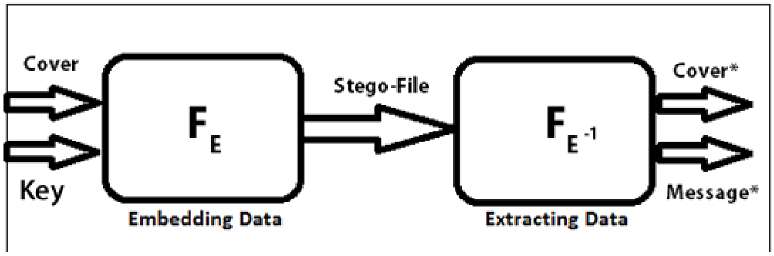


Figure 2: Structure of Steganography system

2.3 STEGANOGRAPHY vs CRYPTOGRAPHY

Basically, the purpose of cryptography and steganography is to provide secret communication. However, steganography is not the same as cryptography. Cryptography hides the contents of a secret message from a malicious people, whereas steganography even conceals the existence of the message. Steganography must not be confused with cryptography, where we transform the message so as to make its meaning obscure to malicious people who intercept it. Therefore, the definition of breaking the system is different. In cryptography, the system is broken when the attacker can read the secret message. Breaking a steganographic system need the attacker to detect that steganography has been used and he is able to read the embedded message. [4]

In cryptography, the structure of a message is scrambled to make it meaningless and unintelligible unless the decryption key is available. It makes no attempt to disguise or hide the encoded message. Basically, cryptography offers the ability of transmitting information between persons in a way that prevents a third party from reading it. Cryptography can also provide authentication for verifying the identity of someone or something.

In contrast, steganography does not alter the structure of the secret message, but hides it inside a *cover-image* so it cannot be seen. A message in cipher text, for instance, might arouse suspicion on the part of the recipient while an “invisible” message created with steganographic methods will not. In other word, steganography prevents an unintended recipient from suspecting that the data exists. In addition, the security of classical steganography system relies on secrecy of the data encoding system. Once the encoding system is known, the steganography system is defeated.

It is possible to combine the techniques by encrypting message using cryptography and then hiding the encrypted message using steganography. The resulting *stego-image* can be transmitted without revealing that secret information is being exchanged. Furthermore, even if an attacker were to defeat the steganographic technique and detect the message from the *stego-object*, he would still require the cryptographic decoding key to decipher the encrypted message. Table 1 shows that both technologies have counter advantages and disadvantages.

Steganography	**Crypyography**
➢ Hide, Without altering ➢ Obfuscates the fact of communication, not the data ➢ Preventative – deter attacks	➢ Alter, without hiding ➢ Obfuscates the data, not fact of the communication ➢ Curative - defends attacks

Table 1:Comparision of steganography and cryptography

2.4 HISTORY OF STEGANOGRAPHY

The word “Steganography” technically means “covered or hidden writing”. Its ancient origins can be traced back to 440 BC. Although the term steganography was only coined at the end of the 15th century, the use of steganography dates back several millennia.[5].

Some examples of use of Steganography is past times are:

1. In ancient Greece, messages were hidden on the back of wax writing tables where someone would peel off the wax that was or written on the stomachs of rabbits.
2. In Ancient Greece they used to select messengers and shave their head, they would then write a message on their head. Once the message had been written the hair was allowed to grow back. After the hair grew back the messenger was sent to deliver the message, the recipient would shave off the messenger's hair to see the secrete message.
3. During World War 2 invisible ink was used to write information on pieces of paper so that the paper appeared to the average person as just being blank pieces of paper. Liquids such as milk, vinegar and fruit juices were used, because when each one of these substances were heated they darken and become visible to the human eye. Invisible ink has been in use for centuries—for fun by children and students and for serious espionage by spies and terrorists.
4. Cryptography became very common place in the middle ages. Secret writing was employed by the Catholic Church in its various struggles down the ages and by the major governments of the time. Steganography was normally used in conjunction with cryptography to further hide secret information

2.5 STEGANOGRAPHY TYPES

Almost all digital file formats can be used for steganography, but the formats that are more suitable are those with a high degree of redundancy. Redundancy can be defined as the bits of an object that provide accuracy far greater than necessary for the object's use and display. The redundant bits of an object are those bits that can be altered without the alteration being detected easily Image and audio files especially comply with this requirement, while research has also uncovered other file formats that can be used for information hiding.[6]

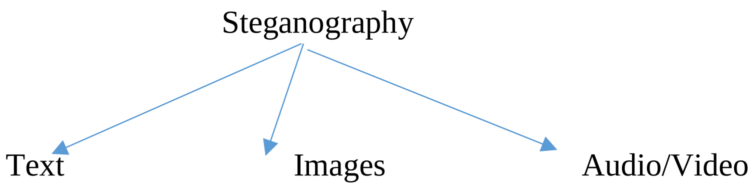


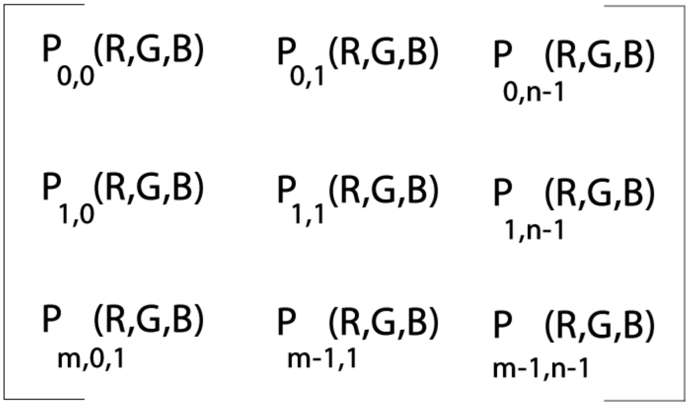
Figure 3: Types of steganography

2.6 DIGITAL IMAGE STEGANOGRAPHY

Steganography can also be classified a on the basis of carrier media. The most commonly used media are text, image, audio and video. So here Digital Images are used as the carrier media.

2.6.1 DIGITAL IMAGE

A digital image is defined for the purposes of this document as a raster based, 2-dimensional, rectangular array of static data elements called pixels, intended for display on a computer monitor or for transformation into another format, such as a printed page. To a computer, an image is an array of numbers that represent light intensities at various points, or pixels. These pixels make up the image's raster data. Digital images are typically stored in 32-, 24- or 8-bit per pixel files. In 8-bit color images, (such as GIF files), each pixel is represented as a single byte[7]. A typical 32-bit picture of width=n pixels and height = m pixels can be represented by an m x n matrix of pixels. We have used 24-bit bitmap image file format here.



$$\begin{bmatrix} P_{0,0}(R,G,B) & P_{0,1}(R,G,B) & P_{0,n-1}(R,G,B) \\ P_{1,0}(R,G,B) & P_{1,1}(R,G,B) & P_{1,n-1}(R,G,B) \\ P_{m,0,1}(R,G,B) & P_{m-1,1}(R,G,B) & P_{m-1,n-1}(R,G,B) \end{bmatrix}$$

Where,

$0 \leq R \leq (2^8\text{-}1)_2$

$0 \leq G \leq (2^8\text{-}1)_2$

$0 \leq B \leq (2^8\text{-}1)_2$

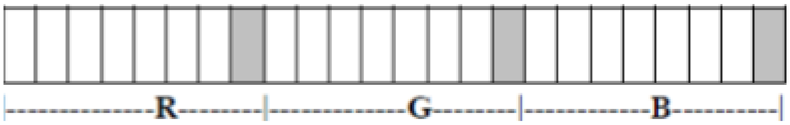


Figure 4: Matrix and Bits Representation of Image

The three 8 bit parts - red-R, blue-B and green-G - constitute 24 bits which means that a pixel should have 24 bits. We have used LSB substitution algorithm to hide the encrypted message into the image. We can hide 3 bit in each pixel of bitmap image.

2.6.2 IMAGE FORMATS

There are several image formats in use nowadays. Since raw image files are quite large, some suitable compression technique is applied to reduce the size. Based on the kind of compression employed a given image format can be classified as lossy or lossless. Lossy compression is used mostly with JPEG files and may not maintain the original image's integrity despite providing high compression. Obviously it would infect any data embedded in the image. Lossless compression does maintain the original image data exactly but does not offer such high compression rates as lossy compression.

BMP:

Since BMP is a fairly simple file format, its structure is pretty straightforward. Each bitmap file contains:

- a bitmap-file header: this contains information about the type, size, and layout of a device-independent bitmap file.
- a bitmap-information header which specifies the dimensions, compression type, and color format for the bitmap.
- a color table, defined as an array of RGBQUAD structures, contains as many elements as there are colors in the bitmap. The color table is not present for bitmaps

with 24 color bits because each pixel is represented by 24-bit red-green-blue (RGB) values in the actual bitmap data area.

- an array of bytes that defines the bitmap bits. These are the actual image data, represented by consecutive rows or 'scan lines' of the bitmap. Each scan line consists of consecutive bytes representing the pixels in the scan line, in left-to-right order.[8]

BMP files always contain RGB data. The file can be:

- 1-bit: 2 colors (monochrome)
- 4-bit: 16 colors
- 8-bit: 256 colors.
- 24-bit: 16777216 colors, mixes 256 tints of Red with 256 tints of Green and Blue

We have chosen 24-bit Bitmap image format as our carrier media because of the following advantages:

- minimum compression loss. The image quality is not changed by any compression ratio;
- format is suitable for storage of intermediate versions of the image. When you re-save image, quality is not lost;
- PNG supports a large number of colors. PNG-8 (256 colors) and PNG-24 (about 16.7 million. Colors);
- it supports multi-level of transparency. Image has the 256 levels of opacity from fully opaque to fully transparent;
- it's possible to work with layers;
- small sizes files.
- Minimum compression loss. The image quality is not changed.

2.7 STEGANOGRAPHY TECHNIQUES

Digital data can be embedded in many ways into the images, e.g. sequential, random, non-random (looking for .noisy. areas of the image, that will attract less attention), redundant etc. Each one of these has its own merits and demerits. The most common techniques of data hiding in images are:

1. Appending data bytes at the end of carrier:

   The secret data bytes are appended at the end of the carrier media such as image and the carrier media is then compressed to its original size to reduce the suspects of having secret data. Advantage is that it is very easy to implement. Disadvantage is it is very easy to detect and get the message.
2. Transform domain based embedding: Transform Embedding Techniques embed the data by modulating coefficients in a transform domain, such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) (used in JPEG compression), or Discrete Wavelet Transform (DWT). Modifying the transform coefficients provides more robustness to the compression (especially to lossy), cropping, or some image processing, than LSB techniques. The Spread-Spectrum Image Steganography (SSIS) hides the data within noise which is then added to the cover. The noise is of the type usually incurred during the image acquisition process. Such a noise is imperceptible to humans if kept to limited extent. The decoding process involves image restoration techniques and error control coding.
3. Masking and filtering techniques:

   This technique embeds information to perceptually significant areas of the image. The use of significant parts makes these techniques very robust. Masking refers to the phenomenon were a signal can be imperceptible to an observer in the presence of another signal - referred to as the masker. The phenomenon of camouflage is manifestation of this human weakness. The image must be analyzed in advance for the information to determine appropriate regions to place the message data so that it is camouflaged in the environment.
4. Least significant bit (LSB) insertion:

   LSB techniques embed the message bits directly into the least significant bit plane of the cover image in a deterministic sequence. This results in a change with too low an amplitude to be human-perceptible. LSB embedding is simple, popular and many techniques use these methods. The problem is its vulnerability to image manipulation.

We have used LSB algorithms for hiding the text into the carrier image.

2.8 LEAST SIGNIFICANT BIT (LSB) ALGORITHM

LSB (Least Significant Bit) substitution is the process of adjusting the least significant bit pixels of the carrier image. It is a simple approach for embedding message into the image. The Least Significant Bit insertion varies according to number of bits in an image. For an 8-bit image, the least significant bit i.e., the 8th bit of each byte of the image is changed to the bit of secret message. For 24-bit image, the colors of each component like RGB (red, green and blue) are changed. LSB is effective in using BMP images since the compression in BMP is lossless. But for hiding the secret message inside an image of BMP file using LSB algorithm it requires a large image which is used as a cover. LSB substitution is also possible for GIF formats, but the problem with the GIF image is whenever the least significant bit is changed the whole color palette will be changed. The problem can be avoided by only using the gray scale GIF images since the gray scale image contains 256 shades and the changes will be done gradually so that it will be very hard to detect. For JPEG, the direct substitution of steganographic techniques is not possible since it will use lossy compression.

2.8.1 LSB INSERTION

Other question which may arise in mind may be “Why we have used least significant bit approach for steganography and bmp format of the image?”

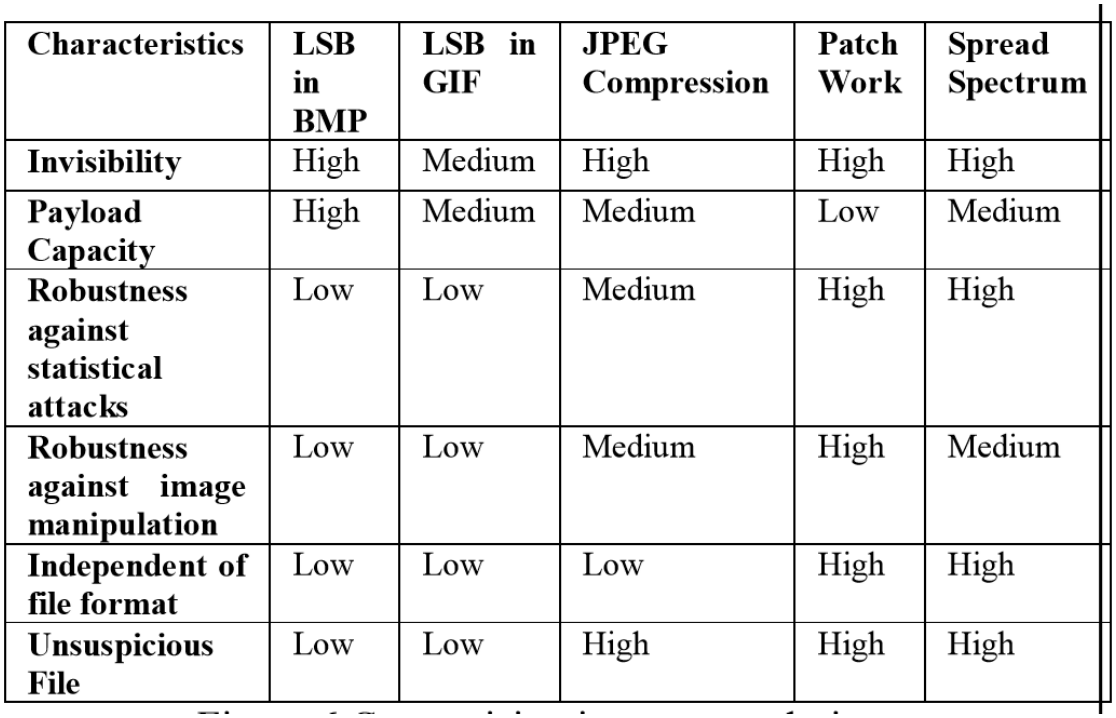
Before answering this question want to share some important things. Image Steganography technique can be divided into two groups: Image Domain and Transform Domain. Image Domain technique embed message in the intensity of the pixel directly whereas in transform domain technique, images are first transformed and then the message is embedded in image. Image Domain technique encompasses bit wise methods that apply bit insertion and noise manipulation and are sometimes characterized as” simple system” but the transform domain involves the manipulation of algorithms and image transforms. The LSB method is used to embed the image. Least Significant Bit(LSB) insertion is a common, simple approach to embedding information in a cover file which is image. The algorithm of LSB method used in the system is described here with the help of example. Example: The letter A can be hidden in three pixels (assuming no compression). The original raster data for 3 pixels (9 bytes) may be:

(00100111 11101001 11001000) (00100111 11001000 11101001) (11001000 00100111 11101001)

The binary value for A is 10000011. Inserting the binary value for A in the three pixels would result

(00100111 11101000 11001000) (00100110 11001000 11101000) (11001000 00100111 11101001)

We have used BMP Image for our proposed work because by comparing between stego techniques as shown in figure below. we find that LSB coding in BMP image is the simplest one with great invisibility and payload capacity



Characteristics	LSB in BMP	LSB in GIF	JPEG Compression	Patch Work	Spread Spectrum
Invisibility	High	Medium	High	High	High
Payload Capacity	High	Medium	Medium	Low	Medium
Robustness against statistical attacks	Low	Low	Medium	High	High
Robustness against image manipulation	Low	Low	Medium	High	Medium
Independent of file format	Low	Low	Low	High	High
Unsuspicious File	Low	Low	High	High	High

Table 2: Comparison in stegno techniques

The Human visual system (HVS) cannot detect changes in the color or intensity of a pixel when the LSB bit is modified. This is psycho-visual redundancy since this can be used as an advantage to store information in these bits and yet notice no major difference in the image. [9]

2.8.2 ADVANTAGE OF LSB ALGORITHM

The advantages of LSB are its simplicity to embed the bits of the message directly into the LSB plane of cover-image and many techniques use these methods [11]. Modulating the LSB does not result in a human-perceptible difference because the amplitude of the change is small. Therefore, to the human eye, the resulting stego-image will look identical to the cover-image. [10]

2.8.3 DISADVANTAGE OF LSB ALGORITHM

- We noticed that in the approach discussed above, the time taken for generating the random numbers depends on the size of the key. In our approach it means that it also depends on the cover-image size.
- Although the LSB embedding methods hide data in such a way that the humans do not perceive it, such schemes can be easily destroyed by an opponent such as using lossy compression algorithms or a filtering process.
- Any process that modifies the values of some pixels, either directly or indirectly, may result in degrading of the quality of the original object.[11]

2.9 BLOWFISH ALGORITHM

Blowfish has a 64-bit block size as shown in figure.4 and a variable key length from 32 bits up to 448 bits. It is a 16-round Feistel cipher and uses large key-dependent S-boxes. In blowfish algorithm a 64-bit plaintext message is first divided into 32 bits. Each line represents 32 bits. The algorithm keeps two sub key arrays: the 18-entry P-array and four 256-entry S-boxes. The S-boxes accept 8-bit input and produce 32-bit output. One entry of the P-array is used every round, and after the final round, each half of the data block is XORed with one of the two remaining unused P-entries.

The F-function as we can see in figure 5 splits the 32-bit input into four eight-bit quarters, and uses the quarters as input to the S-boxes. The outputs are added modulo 232 and XORed to produce the final 32-bit output. Decryption is exactly the same as encryption, except that P1, P2,..., P18 are used in the reverse order. This is not so obvious because xor is commutative and associative.[12]

We have used pre-built library for blowfish algorithm to encrypt and decrypt text.

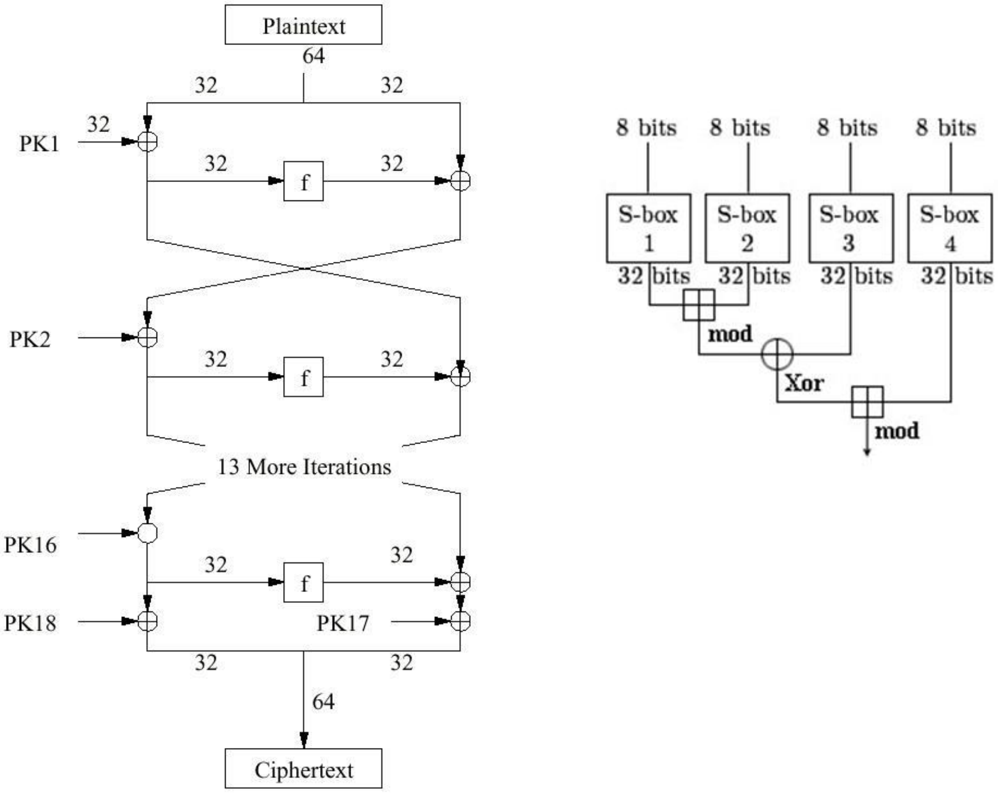


Figure 5: Implementation of Blowfish Encryption Algorithm

3 METHODOLOGY

We have planned to work following these methodologies for the application of knowledge, skills, tools and techniques to a broad range of activities in order to meet the requirements of our project, data security through steganography using LSB. This section presents a detailed information about the software development process, project approach and the tool that we used for our project

3.1 SOFTWARE DEVELOPMENT LIFE CYCLE: AGILE

The framework we will be using for developing this project is Agile model. This is an umbrella term for several iterative and incremental software development methodologies. While each of the agile methodologies is unique in its specific approach, they all share a common vision and core values. They all fundamentally incorporate iteration and the continuous feedback that it provides to successively refine and deliver a software system. They all involve continuous planning, continuous testing, continuous integration, and other forms of continuous evolution of both the project and the software. They are all lightweight, especially compared to traditional waterfall-style processes, and inherently adaptable. What is more important about agile methods is that they all focus on empowering people to collaborate and make decisions together quickly and effectively

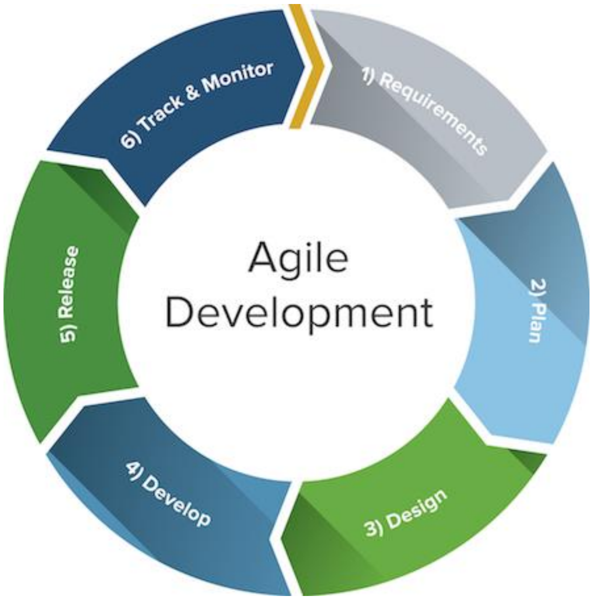


Figure 6 Graphical illustration of the Agile Model

As you see in Figure below, there are five main activities in FDD that are performed iteratively.

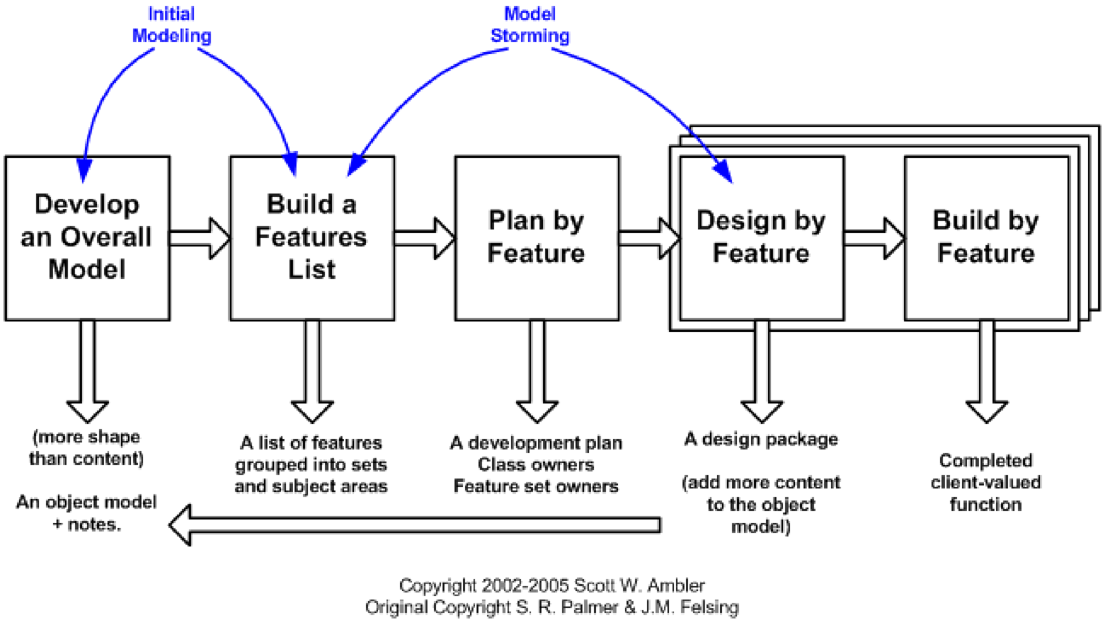

Initial Modeling
Model Storming
Develop an Overall Model
Build a Features List
Plan by Feature
Design by Feature
Build by Feature
(more shape than content)
An object model + notes.
A list of features grouped into sets and subject areas
A development plan Class owners Feature set owners
A design package
(add more content to the object model)
Completed client-valued function
Copyright 2002-2005 Scott W. Ambler
Original Copyright S. R. Palmer & J.M. Felsing

Figure 7: Agile- Feature Driven Development life cycle

3.1.1 Develop an overall model

The initial result being a high-level object model and notes. At the start of a project your goal is to identify and understand the fundamentals of the domain that your system is addressing, and throughout the project you will flesh this model out to reflect what you're building.

3.1.2 Build a features list

The second step is grouping them into related sets and subject areas. These first two steps map to the initial envisioning effort of AMDD. We collect the list of all the features found in our project and specialized them accordingly.

The features are listed below:

1. Image Picker
2. Encryption
3. Decryption
4. Steganography (hide message into image)

5. Steganalysis (Extract message from image)

3.1.3 Analysis by feature

The end result being a development, the identification of class owners, and the identification of feature set owners.

3.1.4 Design by feature

The majority of the effort on an FDD project, roughly 75%, is comprised of the fourth and fifth steps. In this step, we continuously model the feature in detail and build the feature iteratively until all the features are implemented.

3.1.5 Build by features

In this step, the detailed design gotten from previous step are programmed, tested and packaged iteratively.

3.2 PROS OF FDD

1. Lads to move to large projects and obtain repeatable success.

2. Practicing the five processes helps to bring new staff in with a shorter ramp-up time.

3. Feature-Driven Development is built around a core set of industry-recognized best practices

4. Risk Reduction via iteration of design & build in small chunks. FDD helps in reducing risks using shorter iterations of designing, understanding of the requirements and the system- in a clear and distinct way, thereby leading to a state where there are no ambiguities, as the needs and expectations are already understood very well.

5. Clarity of requirements and better understanding of system to be built is gained through the develop overall model process. This process includes high-level walkthrough of the scope of the system and its context. Next, detailed domain walkthroughs are held for each modeling area.

6. Costing the project by feature leads to greater accuracy.

3.3 WHY DID WE CHOOSE FDD

The first process, developing the overall model makes us have a deep understanding of the scope and the context of the project.

The fact that we have a deeper understanding of the requirements and the expectations, that we do small iterations and build small parts, one by one, implies that the risk is really reduced. Less unwanted surprises!

3.4 TOOLS USED

Tools	Purpose
Android emulator	Application testing
Web storm	IDE for react-native app development
GitHub	Manage source code
Star UML	Design
Adobe fireworks	Designing UI/UX
Google chrome	Debugging the code

Table 3: tools used

TECHNOLOGIES USED

1) React native for overall development of the system.
2) Node package manager for running the react native.

4 REQUIREMENT ANALYSIS

Requirement analysis**,** in software engineering encompasses those tasks that go into determining the need and conditions to meet for a new or altered product, taking account of possibly conflicting requirements of the various stakeholders, such as users. It is the early stage activity of requirement engineering which encompasses all activities concerned with eliciting, analyzing, documenting, validating and managing system requirements.

4.1 SYSTEM REQUIREMENT SPECIFICATIONS

4.1.1 Functional Requirements

Provide users to select the image from camera roll as well as the gallery.

Provide users with the interface to insert the secret message and key.

Embed the generated cipher text into the selected image. The cipher text is generated from the message and the key provided.

Extract the secret message from the stegano image if the key matches.

4.1.2 Data entry model

The following input formats were applied:

The secret message and the key are taken from the keyboard entry.

The image to embed message and text is taken from either gallery or camera roll

The image to extract information from is taken from gallery.

4.1.3 Outputs to the user

A stegano image with the embedded message and the key.

The secret message from the stegano image.

4.1.4 Interface Required

Users are provided with an ios/android application interface .

5 SYSTEM DESIGN AND UML MODELS

5.1 USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. The actors for our system are: Sender and Receiver. The simplified and graphical representation of what our system must actually do is represented below:

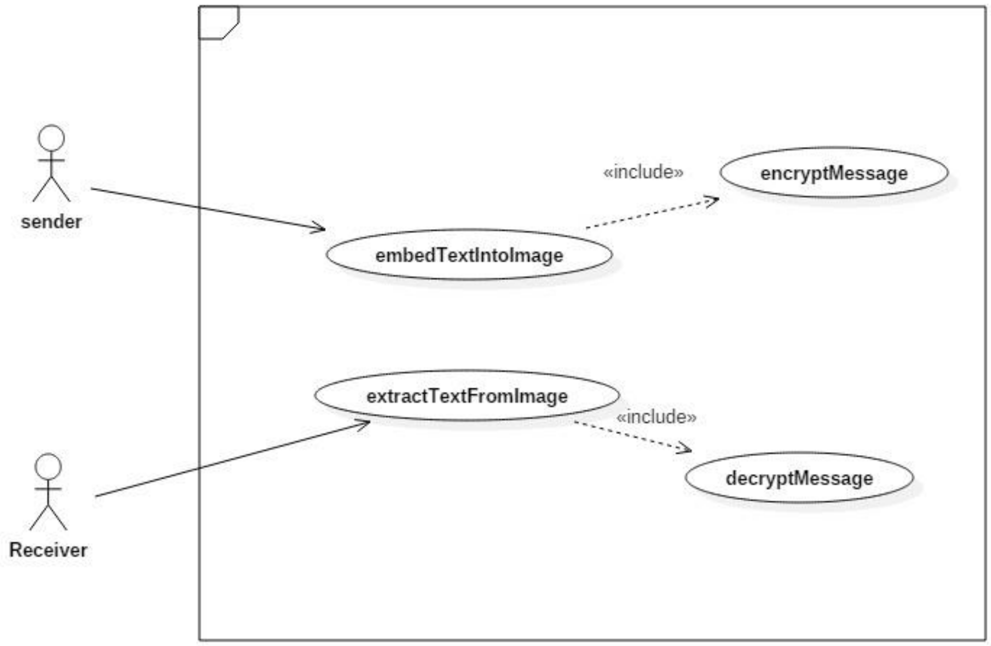


Figure 8: Use case diagram

5.2 OPERATION CONTRACT

Use case UC1: EmbedTextIntoImage:

Primary Actor: Sender

Cross References: Steganography

Secondary Actor: none

Stakeholders:

Sender: Select the image, insert message and key

Preconditions:

image must be .bmp format

24 bit color image

Post conditions:

CryptText instance creation

image source attribute modification

Relation established between EncodeScreen and CryptText

StegoManager instance creation

cipherText attribute modification

Relation established between EncodeScreen and StegoManager

Use case UC1: EmbedTextIntoImage:

Primary Actor: Receiver

Cross References: Steganalysis

Secondary Actor: none

Stakeholders:

receiver: Select the image, insert key

Preconditions:

image must be .bmp format

24-bit color image

Post conditions:

CryptText instance creation

image source attribute modification

Relation established between DecodeScreen and CryptText

StegoManager instance creation

cipherText attribute modification

Relation established between DecideScreen and StegoManager

5.3 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. We used DFD as a preliminary step to create an overview of the system, which can later be elaborated also be used for the visualization of data processing (structured design)

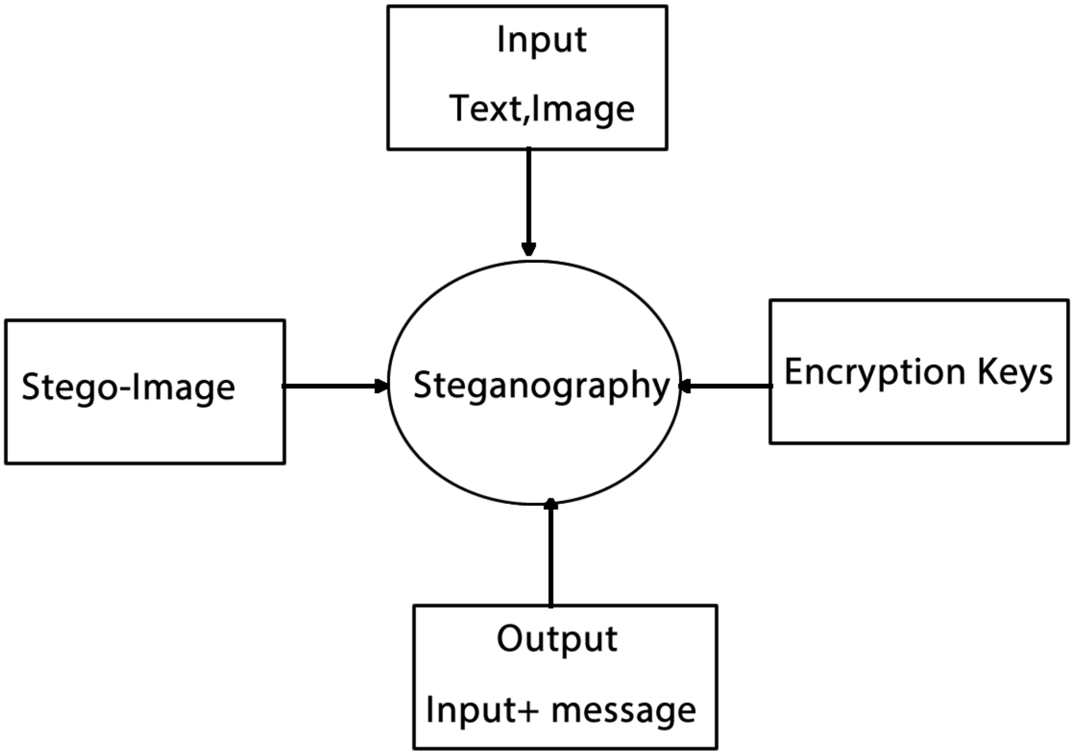


Figure 10: Context Diagram

5.4 DESIGN CLASS DIAGRAM

Design class diagrams are the mainstay of object-oriented analysis and design. Class diagrams show the classes of the system, their interrelationships (including inheritance, aggregation, and association), and the operations and attributes of the classes. Class diagrams are used for a wide variety of purposes, including both conceptual/domain modeling and detailed design modeling

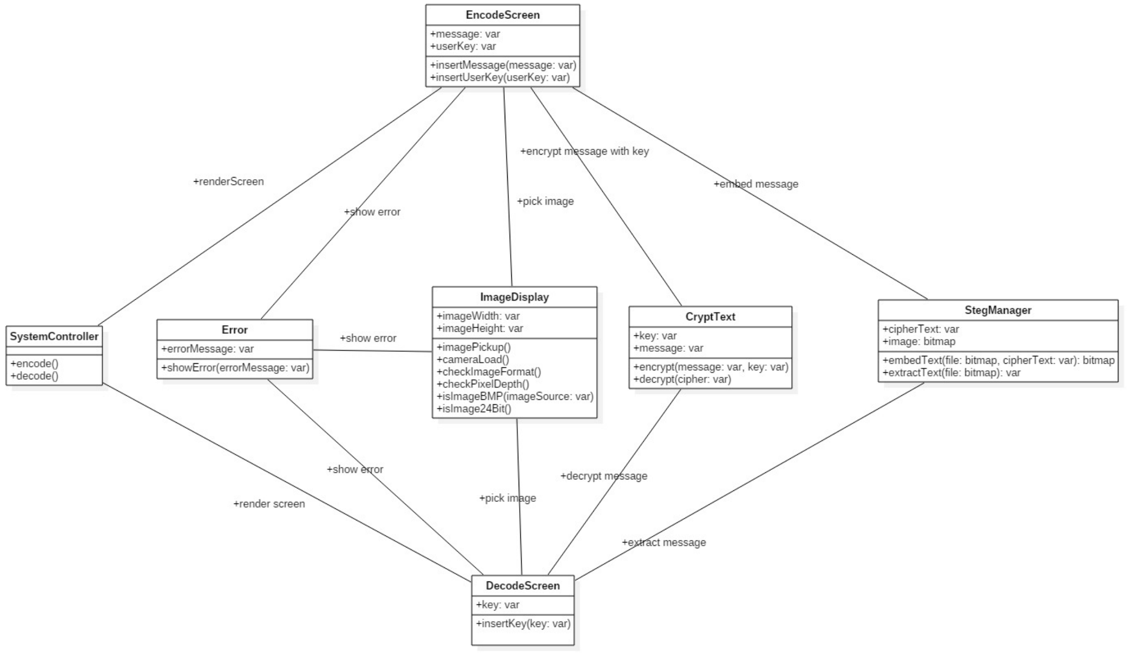


Figure 11: class diagram

5.5 SEQUENCE DIAGRAM

Sequence Diagram is an interaction diagram. It shows how the events occur and in what order. For our system we have designed sequence diagrams for most critical and influential activities which are shown below:

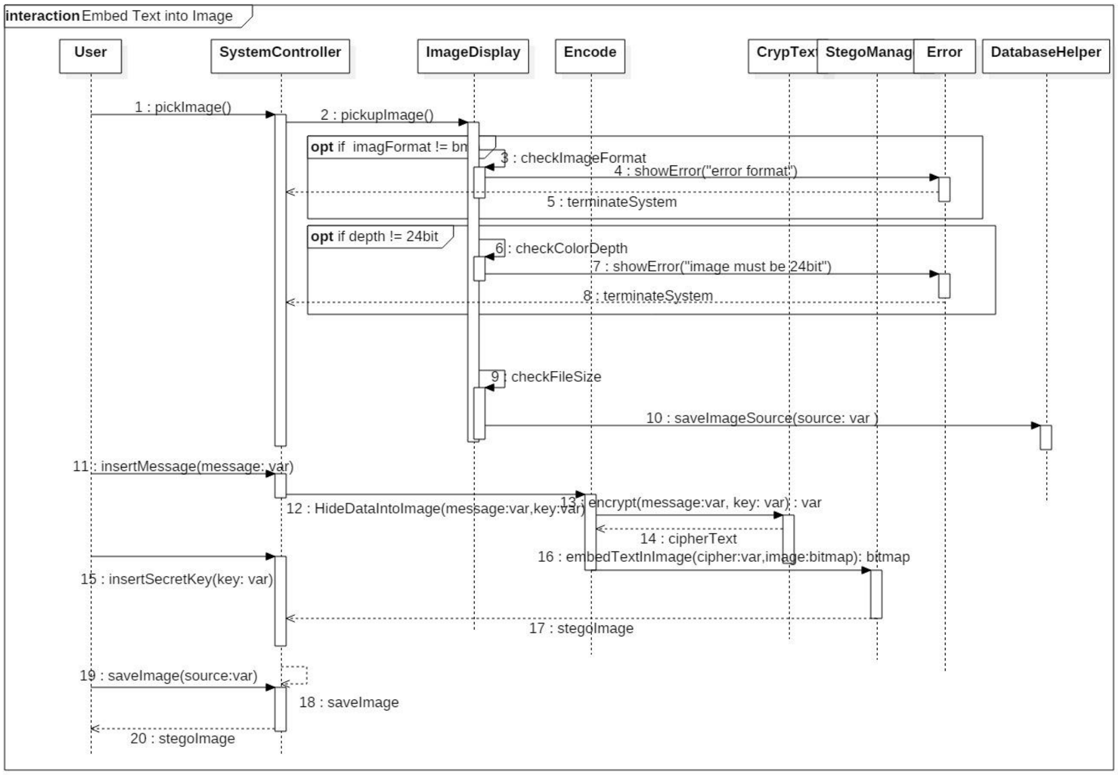

interaction Embed Text into Image
User
SystemController
ImageDisplay
Encode
CrypText
StegoManag
Error
DatabaseHelper
1 : pickImage()
2 : pickupImage()
opt if imagFormat != bm
3 : checkImageFormat
4 : showError("error format")
5 : terminateSystem
opt if depth != 24bit
6 : checkColorDepth
7 : showError("image must be 24bit")
8 : terminateSystem
9 : checkFileSize
10 : saveImageSource(source: var)
11 : insertMessage(message: var)
12 : HideDataIntoImage(message:var,key:var)
13 : encrypt(message:var, key: var) : var
14 : cipherText
16 : embedTextInImage(cipher:var,image:bitmap): bitmap
15 : insertSecretKey(key: var)
17 : stegoImage
19 : saveImage(source:var)
18 : saveImage
20 : stegoImage

Figure 12: Sequence Diagram of Embed Text Into Image

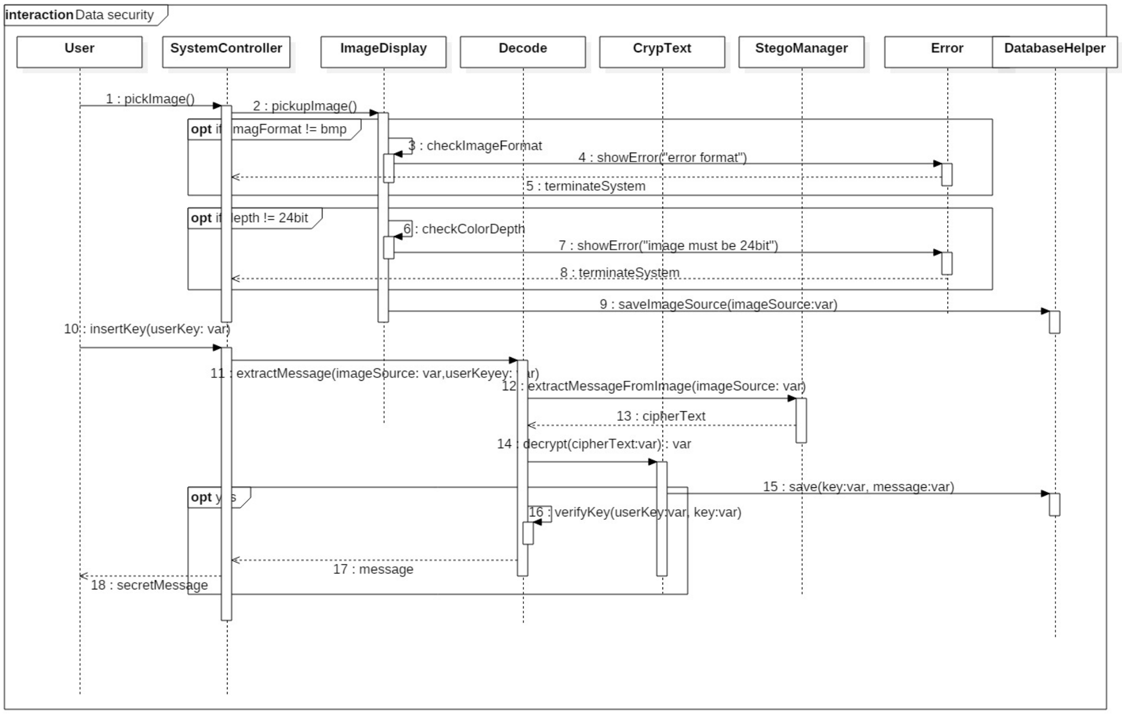

interaction Data security
User
SystemController
ImageDisplay
Decode
CrypText
StegoManager
Error
DatabaseHelper
1 : pickImage()
2 : pickupImage()
opt if imagFormat != bmp
3 : checkImageFormat
4 : showError("error format")
5 : terminateSystem
opt if depth != 24bit
6 : checkColorDepth
7 : showError("image must be 24bit")
8 : terminateSystem
9 : saveImageSource(imageSource:var)
10 : insertKey(userKey: var)
11 : extractMessage(imageSource: var,userKeyey: var)
12 : extractMessageFromImage(imageSource: var)
13 : cipherText
14 : decrypt(cipherText:var) : var
15 : save(key:var, message:var)
opt yes
16 : verifyKey(userKey:var, key:var)
17 : message
18 : secretMessage

Figure 13: Sequence Diagram of Extract Text

6 TESTING

We wanted to make sure that all the component of the developed worked functioned properly. For this, we created a test cases for our work, in which elements such as validation, reliability and user acceptance will be tested. The system will be tested for normal condition, primarily.

6.1 TESTING TABLE

Test no	Component	Test	Expected result	Outcome	Evidence
1	Image picker	Pick image	The image is successfully picked	Successful	Test 1.1
2	Encrypt	Generate cipher text from message and key	A cipher text is successfully generated from the message and key	Successful	Test 1.2
	Decrypt	Extract cipher text from image	A cipher text is successfully extracted from the stegano image	Successful	Test 1.3
	Embed	Insert cipher text into the image	Generated cipher text is embedded into the selected image	Successful	Test 1.4
	Extract	Extract the message from cipher text extracted from stegano image	The message is extracted from the generated cipher text extracted from stegano image	Successful	Test 1.5

Table 4: Test Cases

6.2 TEST EVIDENCES

6.2.1 Test 1.1

Component: Image picker

Purpose: Pick Image

Expected output: The image is successfully picked

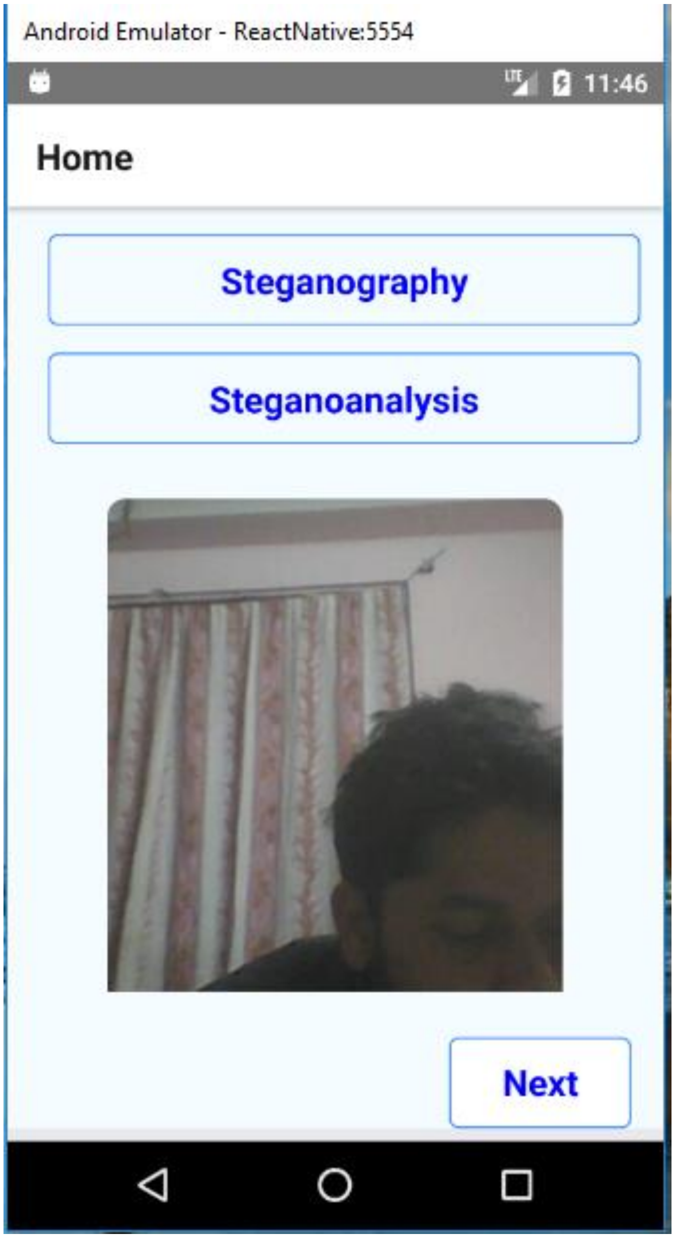


Figure 14: Image picker test case 1.1

6.2.2 Test 1.2

Component: Encrypt

Purpose: Generate cipher text from message and key

Expected output: A cipher text is successfully generated from the message and text

6.2.3 Test 1.3

Component: Decrypt

Purpose: Extract the cipher text from the stegano image

Expected output: A cipher text is successfully generated from the stegano image

6.2.4 Test 1.4

Component: Embed

Purpose: Insert cipher text into the selected image

Expected output: The generated cipher text from the message and key is embedded into the selected image.

6.2.5 Test 1.5

Component: Extract

Purpose: Extract the message from the cipher text

Expected output: The message is extracted from the cipher text generated from the stegano image.

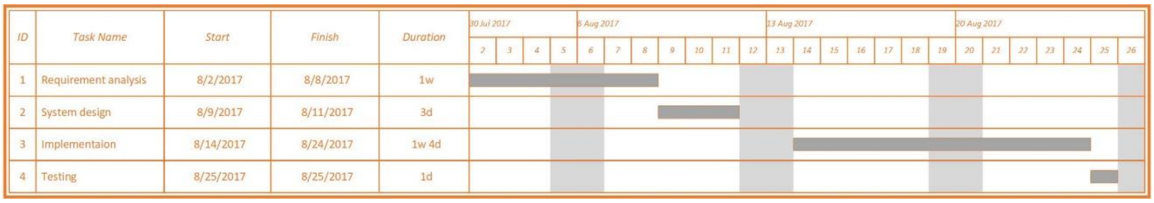
8 PROJECT TASK AND TIME SCHEDULE

The project schedule has been designed as per requirements and constraints involved. This project is scheduled to be completed in about 4-5 months. Requirement analysis have been given more emphasis. Research is to be done first and well documented. Debugging and Testing is to be done prior to the completion of the project.

Task	ImagePicker Feature	Encrypt Feature	Decrypt Feature	Steganography	Steganalysis
Requirement Analysis	3d	2d	1d	4d	1d
Analysis of System	4d	2d	1d	4d	1d
Design System	3d	3d	2d	1w 2d	2d
Implementation	1w 4d	3d	1d	1w	1d
Testing and Debugging	1d	4d	2d	1w 3d	2d
Develop Documentation	6d	1w 1d	6d	1w 2d	5d
Approx. Duration(in hour)	4w	3w 1d	1w 6d	6w 1d	1w 5d

Table 5 Project Task Schedule

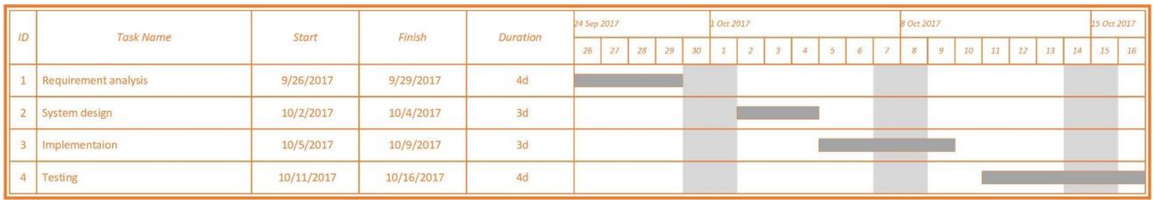
FEATURE 1: IMAGE PICKER



ID	Task Name	Start	Finish	Duration
1	Requirement analysis	8/2/2017	8/8/2017	1w
2	System design	8/9/2017	8/11/2017	3d
3	Implementaion	8/14/2017	8/24/2017	1w 4d
4	Testing	8/25/2017	8/25/2017	1d

Figure 15: Gantt Chart of Image Picker

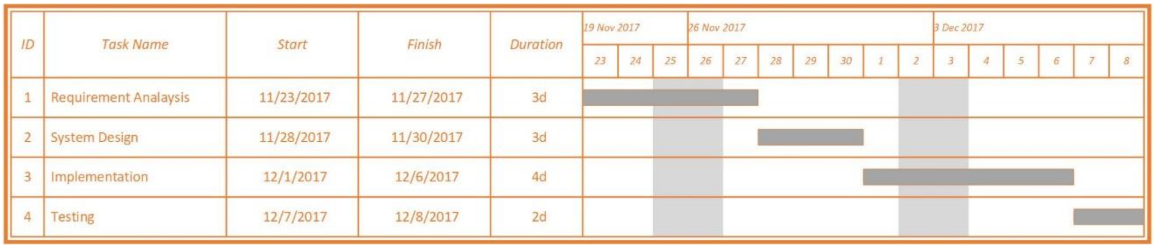
FEATURE 2: ENCRYPTION



ID	Task Name	Start	Finish	Duration
1	Requirement analysis	9/26/2017	9/29/2017	4d
2	System design	10/2/2017	10/4/2017	3d
3	Implementaion	10/5/2017	10/9/2017	3d
4	Testing	10/11/2017	10/16/2017	4d

Figure 16: : Gantt Chart of Encryption

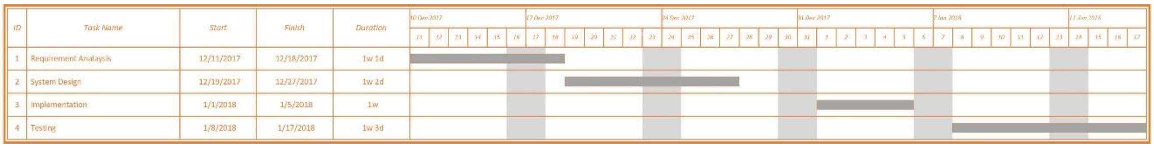
FEATURE 3: DECRYPTION



ID	Task Name	Start	Finish	Duration
1	Requirement Analaysis	11/23/2017	11/27/2017	3d
2	System Design	11/28/2017	11/30/2017	3d
3	Implementation	12/1/2017	12/6/2017	4d
4	Testing	12/7/2017	12/8/2017	2d

Figure 17: : Gantt Chart of Decryption

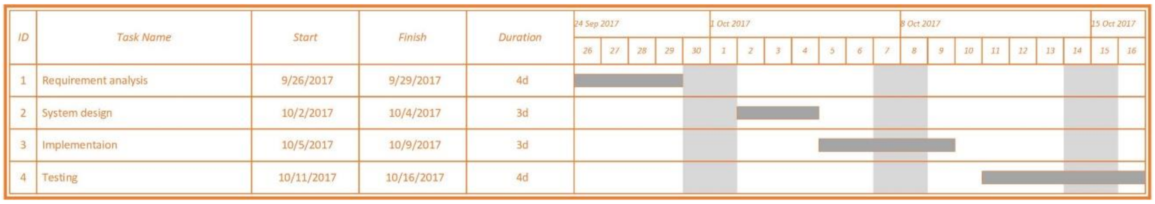
FEATURE 4: STEGANOGRAPHY



ID	Task Name	Start	Finish	Duration
1	Requirement Analaysis	12/11/2017	12/18/2017	1w 1d
2	System Design	12/19/2017	12/27/2017	1w 2d
3	Implementation	1/1/2018	1/5/2018	1w
4	Testing	1/8/2018	1/17/2018	1w 3d

Figure 18: : Gantt Chart of Steganography

FEATURE 5: STEGANALYSIS



ID	Task Name	Start	Finish	Duration
1	Requirement analysis	9/26/2017	9/29/2017	4d
2	System design	10/2/2017	10/4/2017	3d
3	Implementaion	10/5/2017	10/9/2017	3d
4	Testing	10/11/2017	10/16/2017	4d

Figure 19: : Gantt Chart of Steganalysis

9 CONCLUSION

It is observed that through LSB Substitution Steganographic method, the results obtained in data hiding are pretty impressive as it utilizes the simple fact that any image could be broken up to individual bit-planes each consisting of different levels of information. It is to be noted that as discussed earlier, this method is only effective for bitmap images as these involve lossless compression techniques. Also, in this project grey-scale images have been used for demonstration. But this process can also be extended to be used for color images where, biplane slicing is to be done individually for the top four bit-planes for each of R, G, B of the message image, which are again to be placed in the R, G, B planes of the cover image, and extraction is done similarly.

It is also important to discuss that though steganography was once undetected, with the various methods currently used, it is not only easy to detect the presence but also retrieving them is easier. For instance, without having to use a software or complex tools for detection, simple methods to observe if an image file has been manipulated are:

1. Size of the image: A Steganographic image has a huge storage size when compared to a regular image of the same dimensions. I.e. if the original image storage size would be few KBs, the Steganographic image could be several MBs in size. This again varies with the resolution and type of image used.
2. 2. Noise in image: A Steganographic image has noise when compared to a regular image. This is the reason why initially little noise is added to the cover image, so that the Steganographic image doesn't appear very noisy when compared to the original cover image.

Though this project focusses on LSB and spatial domain steganography, few details about transform domain methods have also been researched, basics of which have been discussed. So through the various articles and theory available, it is observed that transform domain methods perform better in comparison with spatial domain methods.

10 FURTHER WORKS

LSB techniques used with BITMAP format could prove to be beneficial for hiding data in BMP images in spatial domain over the network. High level security of confidential information can be achieved in spatial domain Distortion caused is less , so data could be easily retrieved and reconstructed by the intended receiver will help in its wider use. The proposed system which we are using for standalone system but the same technique could be implemented for multiple machine over the network.

11 BIBLIOGRAPHY

[1] R.Anderson and F. Petitcolas, "On the limits of steganography" IEEE Journal of Selected Areas in Communications, Vol. 16, No. 4, May 1998. [offline]

[2] Niels Provos, Peter Honeyman, "Hide and Seek: An Introduction to Steganography," IEEE computer society,2003 [offline]

[3] Animesh Shaw, Feb 24, 2015. Cryptography vs Steganography. [https://www.slideshare.net/AnimeshShawRana/cryptography-steganography]. Accessed 3rd Nov, 2017.

[4] Implementation & Analysis of Covert Data Hiding Techniques [http://shodhganga.inflibnet.ac.in/bitstream/10603/56270/12/12_chapter%202.pdf] Accessed on 11st Nov 2017

[5] David Kahn , May 30 - June 01, 1996. The History of Steganography [https://www.researchgate.net/publication/220722213_The_History_of_Steganography] Accessed on Nov 22nd 2017.

[6] Harpreet Kaur,Jyoti Rani; Sept 18, 2016.A Survey on different techniques of steganography [https://www.matecconferences.org/articles/matecconf/pdf/2016/20/matecconf_icaet2016_02003.pdf] Accessed on Dec 3rd 2017.

[7] Abbas Cheddad, Kevin Curran, Paul Mc Kevitt; Aug 2009. Digital image steganography: Survey and analysis of current methods [https://www.sciencedirect.com/science/article/pii/S0165168409003648] Accessed on Dec 8th 2017

[8] International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-5, November 2013 [offline]. Access on Dec 21st

[9] Champakamala .B.S, Padmini.K, Radhika .D. K; INTERNATIONAL JOURNAL OF ADVANCE COMPUTER TECHNOLOGY | VOLUME 3, NUMBER 4, [http://ijact.org/volume3issue4/IJ0340004.pdf] Access on Jan 5th 2018

[10] Sree Lakshmi Sree; Nov 9,2015. Image Steganography using LSB [https://www.slideshare.net/SreelekshmiSree1/image-steganography-using-

lsb]. Accessed on jan 6th 2018

[11] Kshetrimayum Jenita Devi; May 2013. A Sesure Image Steganography Using LSB Technique and Pseudo Random Encoding Technique [http://ethesis.nitrkl.ac.in/4626/1/109CS0608.pdf] Accessed on Jan 8th 2018

[12] Kevin Hackett; Nov 26, 1998. Encryption Using Blowfish Algorithm. [http://www.ece.ualberta.ca/~elliott/ee552/studentAppNotes/1998f/blowfish_encryption/] Accessed on Jan 12 2018