Project Report

On

STEGANOGRAPHY

Project work submitted in partial fulfilment

of the requirement for the award of the

degree

BACHELOR OF COMPUTER APPLICATIONS

By

CHINNI DILEEP RAJA (107217861012)

A.NITESH KUMAR (107217861028)

BHAVAN'S VIVEKANANDA COLLEGE

SAINIKPURI, SECUNDERABAD

Accredited with 'A' grade by

NAAC Autonomous College

(Affiliated to Osmania University)

BHAVAN'S VIVEKANANDA COLLEGE

OF SCIENCE, HUMANITIES AND COMMERCE

SAINIKPURI,SECUNDERABAD

Autonomous College - Affiliated to Osmania University

Accredited with 'A' grade by NAAC

PROJECT CERTIFICATE

(B.C.A)

It is to certify that this is a bonafide record of the work done in the laboratory during the year 2019 - 2020 by

Name	**: CHINNI DILEEP RAJA**	**A.NITESH KUMAR**
Roll Number	**: 107217861012**	**107217861028**
Class / Sem	**: B.C.A (VI SEMESTER)**	
Subject	**: BCA646 Main Project**	

Certified by:

Signature of the External Examiner **Signature of the Internal Examiner**

Course Co-Ordinator

STEGANOGRAPHY

(A new technique to hide information within image file)

ACKNOWLEDGEMENT

My project report **STEGANOGRAPHY** would be incomplete without expressing my thanks to all those who have helped me in completing this project.

I would like to thank our principal **Prof. Y. ASHOK** for giving me this opportunity to make a study on any well-defined problem. I always got a great support and encouragement from him.

I would like to express my sincere thanks to our **BCA** course coordinator **Mr. G. MAHESH KUMAR** of our college and has also been my guide throughout my study. During the project, he was there with me and always gave me a right guidance and it is with his support and guidance that today I have completed this project.

I would like to thank **Ms. DIVYA REKHA** who always welcomed us for any sort of problem and helped every time by giving us information and guidance. She always supported us and pushed us in completing the project. She helped us to cross every hurdle that we faced during the project, and made sure that everything is going accordingly and now here we are with the complete project. I thank mam for supporting us and leading us in completing the project successfully.

I would also like to thank **Mr. BHASKAR** who always helped us with any type of queries we had. He always supported us in completing the project. He was always there with me and gave me right guidance in completing the project.

DECLARATION:

I hereby declare that the project work entitled **"STEGANOGRAPHY"** submitted to the Bhavan's Vivekananda College, Sainikpuri, Secunderabad, is a record of an original work done by me under the guidance of Guide Name, Lecturer, Department of Computer Science, Bhavan's Vivekananda College, Sainikpuri, Secunderabad.

I further certify that the work has not been submitted to any other Institution for the award of any degree or diploma.

CHINNI DILEEP RAJA (107217861012)

A.NITESH KUMAR (107217861028)

ABOUT ORGANISATION

Bhavan's Vivekananda College of Science, Humanities and Commerce was established in 1993 under the aegis of the Bharatiya Vidya Bhavan. Ever since its inspection, the college has been providing quality education to all students in a large variety of courses. The college continues to provide all students a solid foundation for further educational opportunities and the skill for various career opportunities upon graduation.

The college offers a peaceful atmosphere which is ideal for academic pursuits and for the overall development of the students. The college seeks to integrate the student's program of study and the development of skills which includes critical thinking, problem solving, written and oral communication and encouraging research techniques in various fields. It seeks to learn, to adapt and to lead in the creation of a pool of committed and competent individuals indicated to the process of nation building.

The college has a dedicated set of staff, lecturers and support staff. The lecturers, apart from teaching, also imbibe in principles of discipline, commitment and hard work and to strive towards achieving the goals. They support the students in all possible ways and help students to solve their various problems and encourage the students to perform well in both academics and other co-curricular activities.

Overall, the college, with its holistic atmosphere aided for learning, the principles and values that are imbibed into the students, the excellent staff headed by an able and strong Principle dedicated to making the students the citizens of the future, Bhavan's Vivekananda College of Science, Humanities and Commerce is the college, where we enter in as students and pass out as moulded and well read persons, ready to take on the challenges that come by us in the future.

CONTENTS

ABSTRACT

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 steganography techniques some are more complex than others and all of them have respective strong and weak points. Different applications may require absolute invisibility of the secret information, while others require a large secret message to be hidden. This project report intends to give an overview of image steganography, its uses and techniques. It also attempts to identify the requirements of a good steganography algorithm and briefly reflects on which steganographic techniques are more suitable for which applications.

INTRODUCTION

One of the reasons that intruders can be successful is the 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 become more important as more people join the cyberspace revolution. Steganography is the art of concealing information in ways that prevents the detection of hidden messages. Steganography include an array of secret communication methods that hide the message from being seen or discovered.

Due to advances in ICT, most of information is kept electronically. Consequently, the security of information has become a fundamental issue. Besides cryptography, steganography can be employed to secure information. In cryptography, the message or encrypted message is embedded in a digital host before passing it through the network, thus the existence of the message is unknown. Besides hiding data for confidentiality, this approach of information hiding can be extended to copyright protection for digital media: audio, video and images.

The growing possibilities of modern communications need the special means of security especially on computer network. The network security is becoming more important as the number of data being exchanged on the internet increases. Therefore, the confidentiality and data integrity are requires to protect against unauthorized access and use. This has resulted in an explosive growth of the field of information hiding

Information hiding is an emerging research area, which encompasses applications such as copyright protection for digital media, watermarking, fingerprinting, and steganography.

In watermarking applications, the message contains information such as owner identification and a digital time stamp, which usually applied for copyright protection.

Fingerprint, the owner of the data set embeds a serial number that uniquely identifies the user of the data set. This adds to copyright information to makes it possible to trace any unauthorized use of the data set back to the user.

Steganography hide the secrete message within the host data set and presence imperceptible and 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.

What is Steganography?

Steganography is the practice of hiding private or sensitive information within something that appears to be nothing out to the usual. Steganography is often confused with cryptology because the two are similar in the way that they both are used to protect important information. The difference between two is that steganography involves hiding information so it appears that no information is hidden at all. If a person or persons views the object that the information is hidden inside of he or she will have no idea that there is any hidden information, therefore the person will not attempt to decrypt the information.

What steganography essentially does is exploit human perception, human senses are not trained to look for files that have information inside of them, although this software is available that can do what is called Steganography. The most common use of steganography is to hide a file inside another file.

History of Steganography:

The first recorded uses of steganography can be traced back to 440 BC when Herodotus mentions two examples in his *Histories*. Histiaeus sent a message to his vassal, Aristagoras, by shaving the head of his most trusted servant, "marking" the message onto his scalp, then sending him on his way once his hair had regrown, with the instruction, "When thou art come to Miletus, bid Aristagoras shave thy head, and look thereon." Additionally, Demaratus sent a warning about a forthcoming attack to Greece by writing it directly on the wooden backing of a wax tablet before applying its beeswax surface. Wax tablets were in common use then as reusable writing surfaces, sometimes used for shorthand.

Through out history Steganography has been used to secretly communicate information between people.

Some examples of use of Steganography is past times are:

1. 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 are heated they darken and become visible to the human eye.
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 hair to see the secrete message

Why This Steganography?

This technique is chosen, because this system includes not only imperceptibility but also un-delectability by any steganalysis tool.

Project Scope:

This project is developed for hiding information in any image file. The scope of the project is implementation of steganography tools for hiding information includes any type of information file and image files and the path where the user wants to save Image and extruded file.

METHODOLOGY:

User needs to run the application. The user has 2 tab options – encrypt and decrypt. If user select encrypt, application give the screen to select image file, information file and option to save the image file. If user select decrypt, application gives the screen to select only image file and ask path where user want to save the secrete file.

This project has two methods – Encrypt and Decrypt.

In encryption the secret information is hiding in with any type of image file.

Decryption is getting the secret information from image file.

SYSTEM SPECIFICATIONS

Software Requirements:

Operating System	Windows 7 or above
Front End Software	NetBeans IDE 8.2
Backend Software	JAVA

Hardware Requirements:

Processor	Preferably 1.0 GHz or Greater
RAM Size	512 MB or Greater

Limitations of the Software:

This project has an assumption that is both the sender and receiver must have shared some secret information before imprisonment. Pure steganography means that there is none prior information shared by two communication parties.

MODULES:

In this project there are 2 modules, namely

1.”Making stegano medium”.

2. “Getting secret information from stegano medium”.

SOFTWARE USED

What is Netbeans:

NetBeans is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called *modules*. NetBeans runs on Windows, macOS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5, and JavaScript. Applications based on NetBeans, including the NetBeans IDE, can be extended by third party developers.

History of Netbeans:

NetBeans began in 1996 as Xelfi (word play on *Delphi*), a Java IDE student project under the guidance of the Faculty of Mathematics and Physics at Charles University in Prague. In 1997, Roman Staněk formed a company around the project and produced commercial versions of the NetBeans IDE until it was bought by Sun Microsystems in 1999. Sun open-sourced the NetBeans IDE in June of the following year. Since then, the NetBeans community has continued to grow. In 2010, Sun (and thus NetBeans) was acquired by Oracle Corporation. Under Oracle, NetBeans competed with JDeveloper, a freeware IDE that has historically been a product of the company. In September 2016, Oracle submitted a proposal to donate the NetBeans project to the Apache Software Foundation, stating that it was "opening up the NetBeans governance model to give NetBeans constituents a greater voice in the project's direction and future success through the upcoming release of Java 9 and NetBeans 9 and beyond". The move was endorsed by Java creator James Gosling. The project entered the Apache Incubator in October 2016.

NetBeans IDE releases :

NetBeans IDE 6.0 introduced support for developing IDE modules and rich client applications based on the NetBeans platform, a Java Swing GUI builder (formerly known as "Project Matisse"), improved CVS support, WebLogic 9 and JBoss 4 support, and many editor enhancements. NetBeans 6 is available in official repositories of major Linux distributions.

NetBeans IDE 6.5, released in November 2008, extended the existing Java EE features (including Java Persistence support, EJB 3 and JAX-WS). Additionally, the NetBeans Enterprise Pack supports the development of Java EE 5 enterprise applications, including SOA visual design tools, XML schema tools, web services orchestration (for BPEL), and UML modeling. The NetBeans IDE Bundle for C/C++ supports C/C++ and FORTRAN development.

NetBeans IDE 6.8 is the first IDE to provide complete support of Java EE 6 and the GlassFish Enterprise Server v3. Developers hosting their open-source projects on kenai.com additionally benefit from instant messaging and issue tracking integration and navigation right in the IDE, support for web application development with PHP 5.3 and the Symfony framework, and improved code completion, layouts, hints and navigation in JavaFX projects.

NetBeans IDE 6.9, released in June 2010, added support for OSGi, Spring Framework 3.0, Java EE dependency injection (JSR-299), Zend Framework for PHP, and easier code navigation (such as "Is Overridden/Implemented" annotations), formatting, hints, and refactoring across several languages.

NetBeans IDE 7.0 was released in April 2011. On August 1, 2011, the NetBeans Team released NetBeans IDE 7.0.1, which has full support for the official release of the Java SE7 platform.

NetBeans IDE 7.3 was released in February 2013 which added support for HTML5 and web technologies.

NetBeans IDE 7.4 was released on 15 October 2013.

NetBeans IDE 8.0 was released on 18 March 2014.

NetBeans IDE 8.1 was released on 4 November 2015.

NetBeans IDE 8.2 was released on 3 October 2016.

Netbeans 9.0, which adds support for Java 9 and 10, was released on 29 July 2018, by the Apache Incubator project.

NetBeans 10.0 was released on 27 December 2018. It brings support for Java 11 and improved support for PHP (7.0–7.3).

NetBeans 11.0 was released on 4 April 2019.

NetBeans 11.1 was released on 22 July 2019.

NetBeans platform:

The NetBeans Platform is a framework for simplifying the development of Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

- User interface management (e.g. menus and toolbars)
- User settings management
- Storage management (carries out efficient storage)
- Window management
- Wizard framework (supports step-by-step dialogs)
- NetBeans Visual Library
- Integrated development tools

A showcase of applications developed on top of NetBeans Platform is available at https://netbeans.org/features/platform/showcase.html

NetBeans JavaScript editor:

The NetBeans JavaScript editor provides extended support for JavaScript, Ajax, and CSS.

JavaScript editor features comprise syntax highlighting, refactoring, code completion for native objects and functions, generation of JavaScript class skeletons, generation of Ajax callbacks from a template; and automatic browser compatibility checks.

CSS editor features comprise code completion for styles names, quick navigation through the navigator panel, displaying the CSS rule declaration in a List View and file structure in a Tree View, sorting the outline view by name, type or declaration order (List & Tree), creating rule declarations (Tree only), refactoring a part of a rule name (Tree only).

The NetBeans 7.4 and later uses the new Nashorn JavaScript engine developed by Oracle.

NetBeans IDE Bundle for Web and Java EE:

The **NetBeans IDE Bundle for Web & Java EE** provides complete tools for all the latest Java EE 6 standards, including the new Java EE 6 Web Profile, Enterprise Java Beans (EJBs), servlets, Java Persistence API, web services, and annotations. NetBeans also supports the JSF 2.0 (Facelets), JavaServer Pages (JSP), Hibernate, Spring, and Struts frameworks, and the Java EE 5 and J2EE 1.4 platforms. It includes GlassFish and Apache Tomcat.

Some of its features with Java EE include:

- Improved support for CDI, REST services and Java Persistence
- New support for Bean Validation
- Support for JSF component libraries, including bundled PrimeFaces library
- Improved editing for Expression Language in JSF, including code completion, refactoring and hints.

Netbeans IDE Bundle for PHP:

NetBeans supports PHP since version 6.5. The bundle for PHP includes:

- syntax highlighting, code completion, occurrence highlighting, error highlighting, CVS version control
- semantic analysis with highlighting of parameters and unused local variables
- PHP code debugging with xdebug
- PHP Unit testing with PHPUnit and Selenium
- Code coverage
- Symfony framework support (since version 6.8)
- Zend Framework support (since version 6.9)
- Yii Framework support (since version 7.3)
- PHP 5.3 namespace and closure support (since version 6.8)
- Code Folding for Control Structures (since version 7.2 dev)

NetBeans IDE complete Bundle:

Oracle also releases a version of NetBeans that includes all of the features of the above bundles. This bundle includes:

- NetBeans Base IDE
- Java SE, JavaFX
- Web and Java EE
- Java ME
- C/C++
- PHP (Version 5.5 and later)
- asd
- Apache Groovy
- GlassFish

- Apache Tomcat

  Official Ruby support was removed with the release of 7.0.

What is JAVA:

Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible. It is intended to let application developers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GNU General Public License. Meanwhile, others have developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

The latest versions are Java 14, released in March 2020, and Java 11, a currently supported long-term support (LTS) version, released on September 25, 2018; Oracle released for the legacy Java 8 LTS the last free public update in January 2019 for commercial use, while it will otherwise still support Java 8 with public updates for personal use up to at least December 2020. Oracle (and others) highly recommend uninstalling older versions of Java because of serious risks due to unresolved security issues. Since Java 9, 10, 12 and 13 are no longer supported, Oracle advises its users to immediately transition to the latest version (currently Java 14) or an LTS release.

History of JAVA:

Sun Microsystems released the first public implementation as Java 1.0 in 1996. It promised **Write Once, Run Anywhere** (WORA) functionality, providing no-cost run-times on popular platforms. Fairly secure and featuring configurable security, it allowed network- and file-access restrictions. Major web browsers soon incorporated the ability to run Java applets within web pages, and Java quickly became popular. The Java 1.0 compiler was re-written in Java by Arthur van Hoff to comply strictly with the Java 1.0 language specification. With the advent of Java 2 (released initially as J2SE 1.2 in December 1998 – 1999), new versions had multiple configurations built for different types of platforms. J2EE included technologies and APIs for enterprise applications typically run in server environments, while

J2ME featured APIs optimized for mobile applications. The desktop version was renamed J2SE. In 2006, for marketing purposes, Sun renamed new J2 versions as *Java EE*, *Java ME*, and *Java SE*, respectively.

In 1997, Sun Microsystems approached the ISO/IEC JTC 1 standards body and later the Ecma International to formalize Java, but it soon withdrew from the process. Java remains a *de facto* standard, controlled through the Java Community Process. At one time, Sun made most of its Java implementations available without charge, despite their proprietary software status. Sun generated revenue from Java through the selling of licenses for specialized products such as the Java Enterprise System.

On November 13, 2006, Sun released much of its Java virtual machine (JVM) as free and open-source software (FOSS), under the terms of the GNU General Public License (GPL). On May 8, 2007, Sun finished the process, making all of its JVM's core code available under free software/open-source distribution terms, aside from a small portion of code to which Sun did not hold the copyright.

Sun's vice-president Rich Green said that Sun's ideal role with regard to Java was as an *evangelist*. Following Oracle Corporation's acquisition of Sun Microsystems in 2009–10, Oracle has described itself as the steward of Java technology with a relentless commitment to fostering a community of participation and transparency. This did not prevent Oracle from filing a lawsuit against Google shortly after that for using Java inside the Android SDK (see the *Android* section).

On April 2, 2010, James Gosling resigned from Oracle.

In January 2016, Oracle announced that Java run-time environments based on JDK 9 will discontinue the browser plugin.

Java software runs on everything from laptops to data centers, game consoles to scientific super computers.

Principles of JAVA:

There were five primary goals in the creation of the Java language:

1. It must be simple, object-oriented, and familiar.
2. It must be robust and secure.
3. It must be architecture-neutral and portable.
4. It must execute with high performance.
5. It must be interpreted, threaded, and dynamic.

VERSIONS IN JAVA:

As of March 2020, Java 8 and 11 are supported as Long Term Support (LTS) versions, and one later non-LTS version is supported. Major release versions of Java, along with their release dates:

- JDK 1.0 (January 23, 1996)
- JDK 1.1 (February 19, 1996)
- J2SE 1.2 (December 8, 1998)
- J2SE 1.3 (May 8, 2000)
- J2SE 1.4 (February 6, 2002)
- J2SE 5.0 (September 30, 2004)
- Java SE 6 (December 11, 2006)
- Java SE 7 (July 28, 2011)
- Java SE 8 (March 18, 2014)
- Java SE 9 (September 21, 2017)
- Java SE 10 (March 20, 2018)
- Java SE 11 (September 25, 2018)
- Java SE 12 (March 19, 2019)
- Java SE 13 (September 17, 2019)
- Java SE 14 (March 17, 2020)

JAVA Editions:

Sun has defined and supports four editions of Java targeting different application environments and segmented many of its APIs so that they belong to one of the platforms. The platforms are:

- Java Card for smart-cards.[41]
- Java Platform, Micro Edition (Java ME) – targeting environments with limited resources.[42]
- Java Platform, Standard Edition (Java SE) – targeting workstation environments.[43]
- Java Platform, Enterprise Edition (Java EE) – targeting large distributed enterprise or Internet environments.

SYSTEM ANALYSIS:

Detecting Steganography:

The art of detecting Steganography is referred to as **Steganalysis.**

To put is simply Steganalysis involves detecting the use of Steganography inside of a file. Steganalysis does not deal with trying to decrypt the hidden information inside of a file, just discovering it.

There are many methods that can be used to detect Steganography such as:

"Viewing the file and comparing it to another copy of the file found on the Internet (Picture file). There are usually multiple copies of images on the internet, so you may want to look for several of them and try and compare the suspect file to them. For example if you download a JPED and your suspect file is also a JPED and the two files look almost identical apart from the fact that one is larger than the other, it is most probable you suspect file has hidden information inside of it.

Future Enhancements:

To make it pure steganography application.

Problem Statement:

The former consists of linguistic or language forms of hidden writing. The later, such as invisible ink, try of hide messages physically. One disadvantage of linguistic steganography is that users must equip themselves to have a good knowledge of linguistry. In recent years, everything is trending toward digitization. And with the development of the internet technology, digital media can be transmitted conveniently over the network. Therefore, messages can be secretly carried by digital media by using the steganography techniques, and then be transmitted through the internet rapidly

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 steganography techniques some are more complex than others and all of them have respective strong and weak points.

So we prepare this application, to make the information hiding more simple and user friendly.

Objective:

The goal of steganography is covert communication. So, a fundamental requirement of this steganography system is that the hider message carried by stego-media should not be sensible to human beings.

The other goad of steganography is to avoid drawing suspicion to the existence of a hidden message. This approach of information hiding technique has recently became important in a number of application area

This project has following objectives:

- To product security tool based on steganography techniques.
- To explore techniques of hiding data using encryption module of this project.
- To extract techniques of getting secret data using decryption module.

Steganography sometimes is used when encryption is not permitted. Or, more commonly, steganography is used to supplement encryption. An encrypted file may still hide information using steganography, so even if the encrypted file is deciphered, the hidden message is not seen.

Overview

The word steganography comes from the Greek “Steganos”, which mean covered or secret and – “graphy” mean writing or drawing. Therefore, steganography mean, literally, covered writing. It is the art and science of hiding information such its presence cannot be detected and a communication is happening. A secrete information is encoding in a manner such that the very existence of the information is concealed. Paired with existing communication methods, steganography can be used to carry out hidden exchanges.

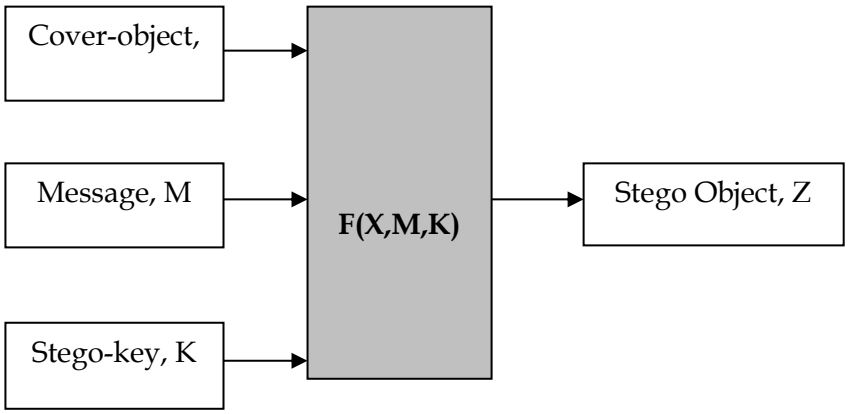
The main goal of this projects it to communicate securely in a completely undetectable manner and to avoid drawing suspicion to the transmission of a hider data. There has been a rapid growth of interest in steganography for two reasons:

The publishing and broadcasting industries have become interested in techniques for hiding encrypted copyright marks and serial numbers in digital films, audio recordings, books and multimedia products

Moves by various governments to restrict the availability of encryption services have motivated people to study methods by which private messages can be embedded in seemingly innocuous cover messages.

The basic model of steganography consists of Carrier, Message and password. Carrier is also known as cover-object, which the message is embedded and serves to hide the presence of the message.

Basically, the model for steganography is shown on following figure:



Message is the data that the sender wishes to remain it confidential. It can be plain text, ciphertext, 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 know 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*.

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.

There are several suitable carriers below to be the *cover-object:*

- Network protocols such as TCP, IP and UDP
- Audio that using digital audio formats such as wav, midi, avi, mpeg, mpi and voc
- File and Disk that can hides and append files by using the slack space
- Text such as null characters, just alike morse code including html and java
- Images file such as bmp, gif and jpg, where they can be both color and gray-scale.

In general, the information hiding process extracts redundant bits from *cover-object.* The process consists of two steps:

- Identification of redundant bits in a *cover-object.* Redundant bits are those bits that can be modified without corrupting the quality or destroying the integrity of the *cover-object.*
- Embedding process then selects the subset of the redundant bits to be replaced with data from a secret message. The *stego-object* is created by replacing the selected redundant bits with message bits

Steganography vs Cryptography:

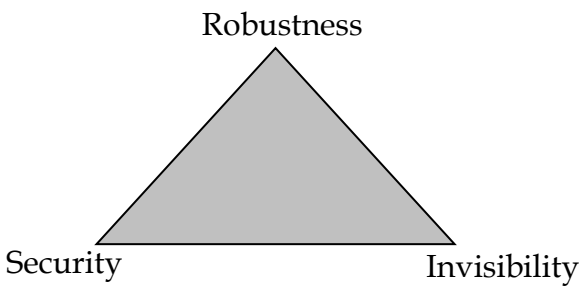
Basically, the purpose of cryptography and steganography is to provide secret communication. However, steganography is not the same as cryptography. Cryptography hides the contents of a secrete message from a malicious people, whereas steganography even conceal the existence of the message. In cryptography, the system is broken when the attacker can read the secret message. Breaking a steganography system need the attacker to detect that steganography has been used.

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.

Steganography vs Watermarking:

Steganography pay attention to the degree of Invisibility while watermarking pay most of its attribute to the robustness of the message and its ability to withstand attacks of removal, such as image operations(rotation, cropping, filtering), audio operations(rerecording, filtering)in the case of images and audio files being watermarked respectively.

It is a non-questionable fact that delectability of a vessel with an introduced data (steganographic message or a watermark) is a function of the changeability function of the algorithm over the vessel.



That is the way the algorithm changes the vessel and the severity of such an operation determines with no doubt the delectability of the message, since delectability is a function of file characteristics deviation from the norm, embedding operation attitude and change severity of such change decides vessel file delectability.

A typical triangle of conflict is message Invisibility, Robustness, and Security. Invisibility is a measure of the in notability of the contents of the message within the vessel.
Security is anonymous to the cryptographic idea to message security, meaning inability of reconstruction of the message without the proper secret key material shared.

Robustness refers to the endurance capability of the message to survive distortion or removal attacks intact. It is often used in the watermarking field since watermarking seeks the persistence of the watermark over attacks, steganographic messages on the other hand tend to be of high sensitivity to such attacks. The more invisible the message is the less secure it is (cryptography needs space) and the less robust it is (no error checking/recovery introduced).The more robust the message is embedded the more size it requires and the more visible it is.

Steganography Techniques:

Over the past few years, numerous steganography techniques that embed hidden messages in multimedia objects have been proposed. There have been many techniques for hiding information or messages in images in such a manner that alteration made to the image is perceptually indiscernible. Commonly approaches are include LSB, Masking and filtering and Transform techniques.

Least significant bit (LSB) insertion is a simple approach to embedding information in image file. The simplest steganography techniques embed the bits of the message directly into least significant bit plane of the cover-image in a

deterministic sequence. Modulating the least significant bit does not result in human perceptible difference because the amplitude of the change is small. In this technique, the embedding capacity can be increased by using two or more least significant bits. At the same time, not only the risk of making the embedded message statistically detectable increase but also the image fidelity degrades. Hence a variable size LSB embedding schema is presented, in which the number of LSBs used for message embedding/extracting depends on the local characteristics of the pixel. The advantage of LSB-based method is easy to implement and high message pay-load.

Although LSB hides the message in such way that the humans do not perceive it, it is still possible for the opponent to retrieve the message due to the simplicity of the technique. Therefore, malicious people can easily try to extract the message from the beginning of the image if they are suspicious that there exists secret information that was embedded in the image.

Therefore, a system named Secure Information Hiding System (SIHS) is proposed to improve the LSB scheme. It overcomes the sequence-mapping problem by embedding the massage into a set of random pixels, which are scattered on the cover-image.

Masking and filtering techniques, usually restricted to 24 bits and gray scale image, hide information by marking an image, in a manner similar to paper watermarks. The technique perform analysis of the image, thus embed the information in significant areas so that the hidden message is more integral to cover image than just hiding it in the noise level.

Transform techniques embed the message by modulating coefficient in a transform domain, such as the Discrete Fourier Transform, or Wavelet Transform. These methods hide messages in significant areas

of the cover image, which make them more robust to attack. Transformations can be applied over the entire image, to block throughout the image, or other variant.

Image Steganography and bitmap pictures:

Using bitmap pictures for hiding secret information is one of most popular choices for Steganography. Many types of software built for this purpose, some of these software use password protection to encrypting information on picture. To use these software you must have a 'BMP' format of a pictures to use it, but using other type of pictures like "JPEG", "GIF" or any other types is rather or never used, because of algorithm of "BMP" pictures for Steganography is simple. Also we know that in the web most popular of image types are "JPEG" and other types not "BPM", so we should have a solution for this problem.

This software provide the solution of this problem, it can accept any type of image to hide information file, but finally it give the only "BMP" image as an output that has hidden file inside it.

Bitmap Steganography:

Bitmap type is the simplest type of picture because that it doesn't have any technology for decreasing file size. Structure of these files is that a bitmap image created from pixels that any pixel created from three colors (red, green and blue said RGB) each color of a pixel is one byte information that shows the density of that color. Merging these three color makes every color that we see in these pictures. We know that every byte in computer science is created from 8 bit that first bit is Most-Significant-Bit (MSB) and last bit Least-Significant-Bit (LSB), the idea of using Steganography science is in this place; we use LSB bit for writing our security information inside

BMP pictures. So if we just use last layer (8st layer) of information, we should change the last bit of pixels, in other hands we have 3 bits in each pixel so we have 3*height*width bits memory to write our information. But before writing our data we must write name of data(file), size of name of data & size of data. We can do this by assigning some first bits of memory (8st layer).

(00101101 00011101 11011100)

(10100110 11000101 00001100)

(11010010 10101100 01100011)

Using each 3 pixel of picture to save a byte of data

System Analysis & Design

Steganography system requires any type of image file and the information or message that is to be hidden. It has two modules encrypt and decrypt.

Microsoft .Net framework prepares a huge amount of tool and options for programmers that they simples programming. One of .Net tools for pictures and images is auto-converting most types of pictures to BMP format. I used this tool in this software called “Steganography” that is written in C#.Net language and you can use this software to hide your information in any type of pictures without any converting its format to BMP (software converts inside it).

The algorithm used for Encryption and Decryption in this application provides using several layers lieu of using only LSB layer of image. Writing data starts from last layer (8st or LSB layer); because

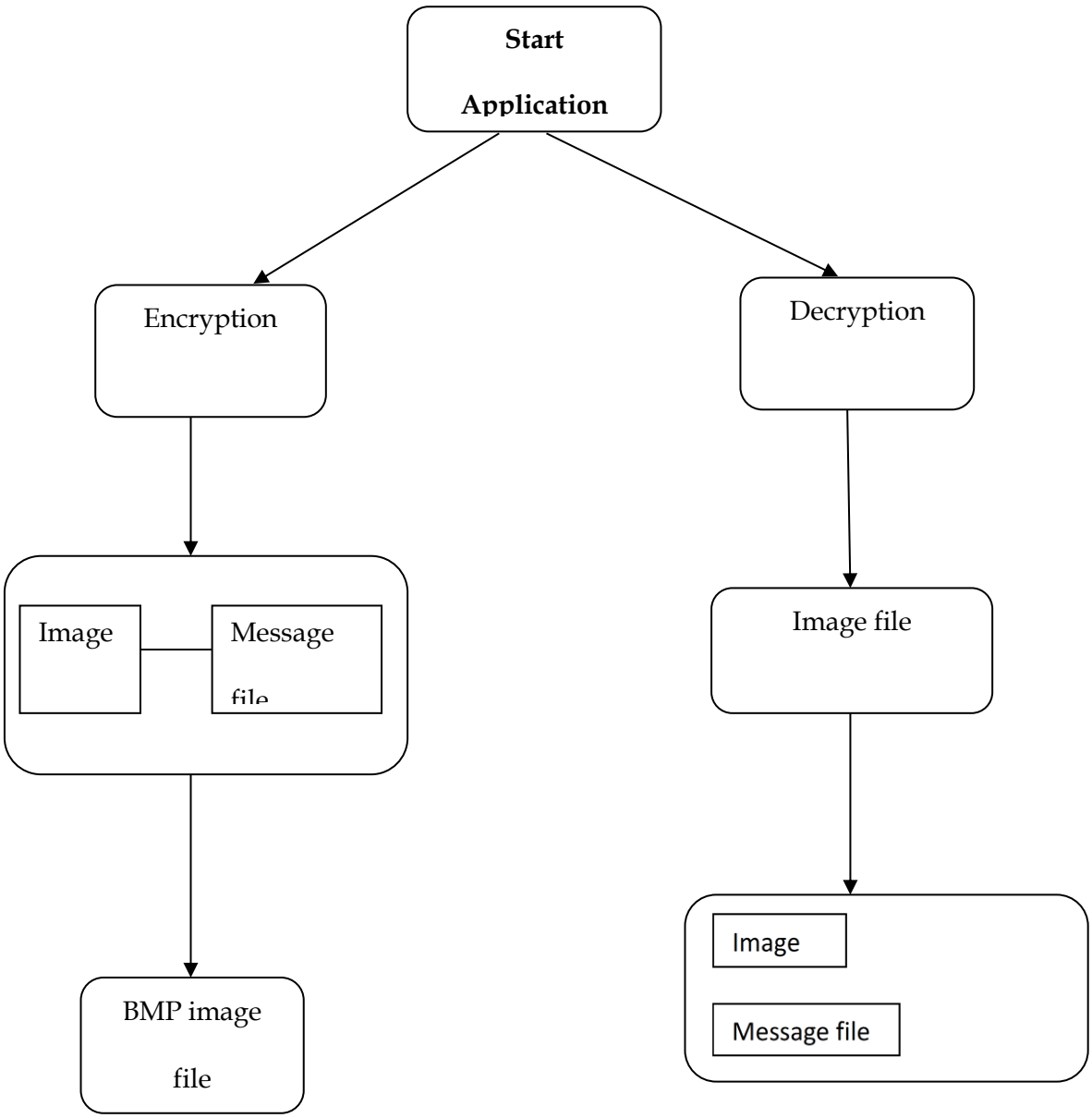
significant of this layer is least and every upper layer has doubled significant from its down layer. So every step we go to upper layer image quality decreases and image retouching transpires.

The encrypt module is used to hide information into the image; no one can see that information or file. This module requires any type of image and message and gives the only one image file in destination.

The decrypt module is used to get the hidden information in an image file. It take the image file as an output, and give two file at destination folder, one is the same image file and another is the message file that is hidden it that.

Before encrypting file inside image we must save name and size of file in a definite place of image. We could save file name before file information in LSB layer and save file size and file name size in most right-down pixels of image. Writing this information is needed to retrieve file from encrypted image in decryption state.

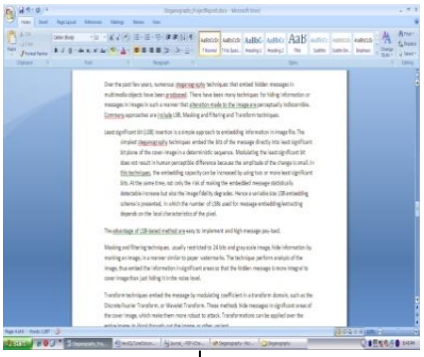
The graphical representation of this system is as follows:


Start
Application
Encryption
Decryption
Image
Message
file
Image file
BMP image
file
Image
Message file

Encryption Process

IMAGE FILE

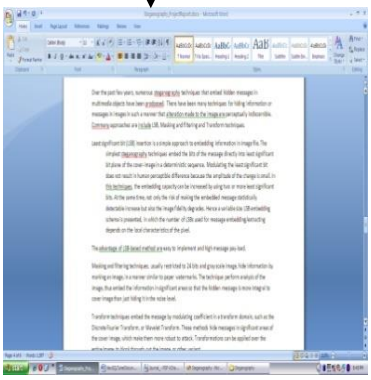
INFORMATION FILE



ENCRYPTED FILE

Decryption Process

ENCRYPTED FILE



INFORMATION FILE **IMAGE FILE**

SYSTEM DESIGN

UML DIAGRAMS

UML stands for Unified Modelling Language. This object-oriented system of notation has evolved from the work of Grady Booch, James Rumbaugh, Ivar Jacobson, and the Rational Software Corporation. These renowned computer scientists fused their respective technologies into a single, standardized model. Today, UML is accepted by the Object Management Group (OMG) as the standard for modelling object oriented programs. The unified modelling language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

Types of UML Diagrams we draw in this project are:

- Use case diagram
- Sequence diagram
- Class diagram
- Activity diagram
- State diagram

USE CASE DIAGRAM

Use case diagram model the functionality of system using actors and use case.

Use Case:

Each use case on the diagram represents a single task that the system needs to carry out. Buy a Product, Add Client, Make Purchase and Validate Order Information are all examples of use cases. Some use cases may include or extend a task represented by another use case. For example, in order to make a purchase, the order information will need to be validated.

State Diagram:

A **state diagram** is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

Actor:

An actor is anything outside the system that interacts with the system to complete a task. It could be a user or another system. The actor "uses" the use case to complete a task. System Administrator, Credit Authentication System, Accounting System and Web Client are all examples of actors. Often, it is useful to look at the set of use cases that an actor has access to -- this defines the actor's overall role in the system.

Association:

The association is the link that is drawn between and actor and a use case. It indicates which actors interact with the system to complete the various tasks.

SEQUENCE DIAGRAMS:

Sequence diagrams describe the interactions among classes in terms of an exchange of messages over time. It is an easy and intuitive way of describing the behaviour of system by viewing the interaction between the system and its environment.

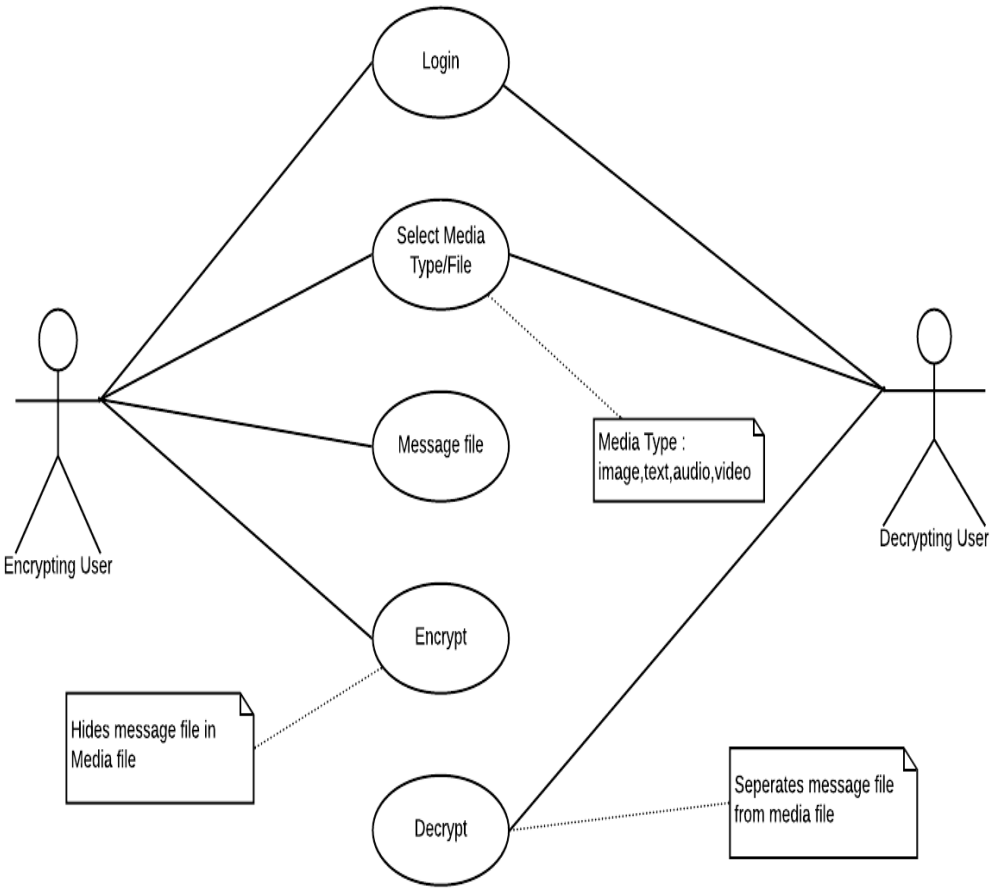
CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

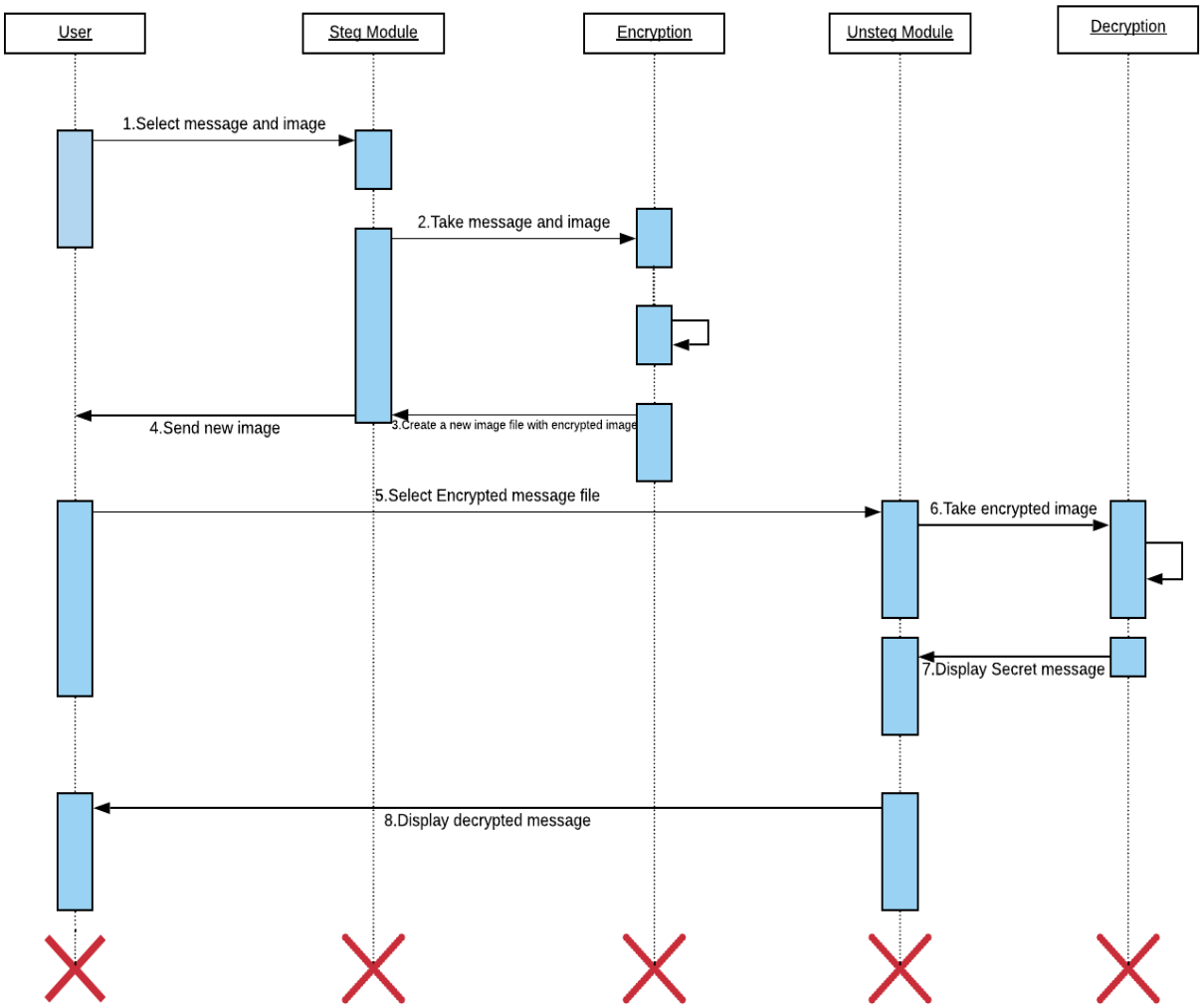
ACTIVITY DIAGRAM:

Activity diagrams illustrate the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation.

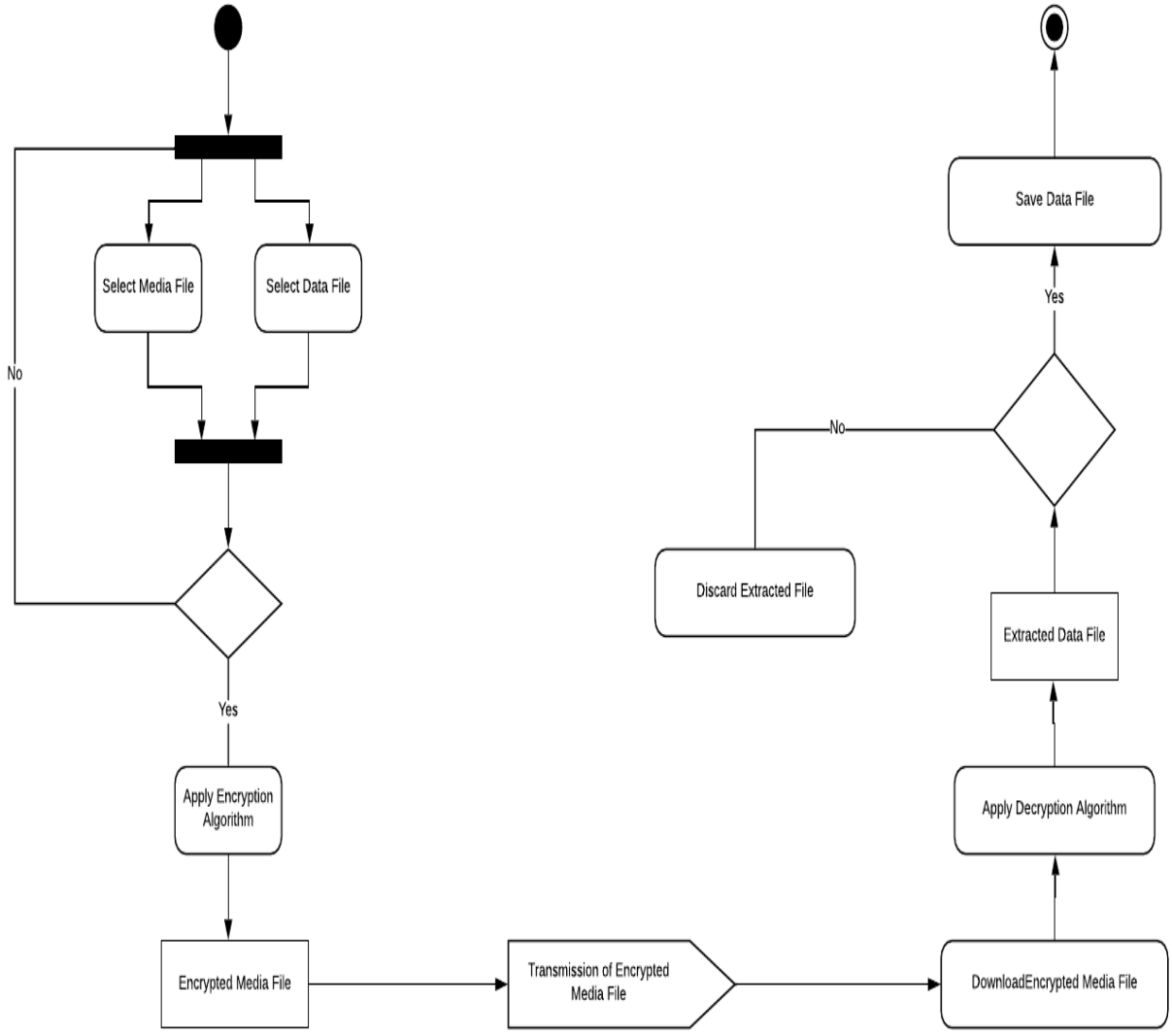
USE CASE DIAGRAM:



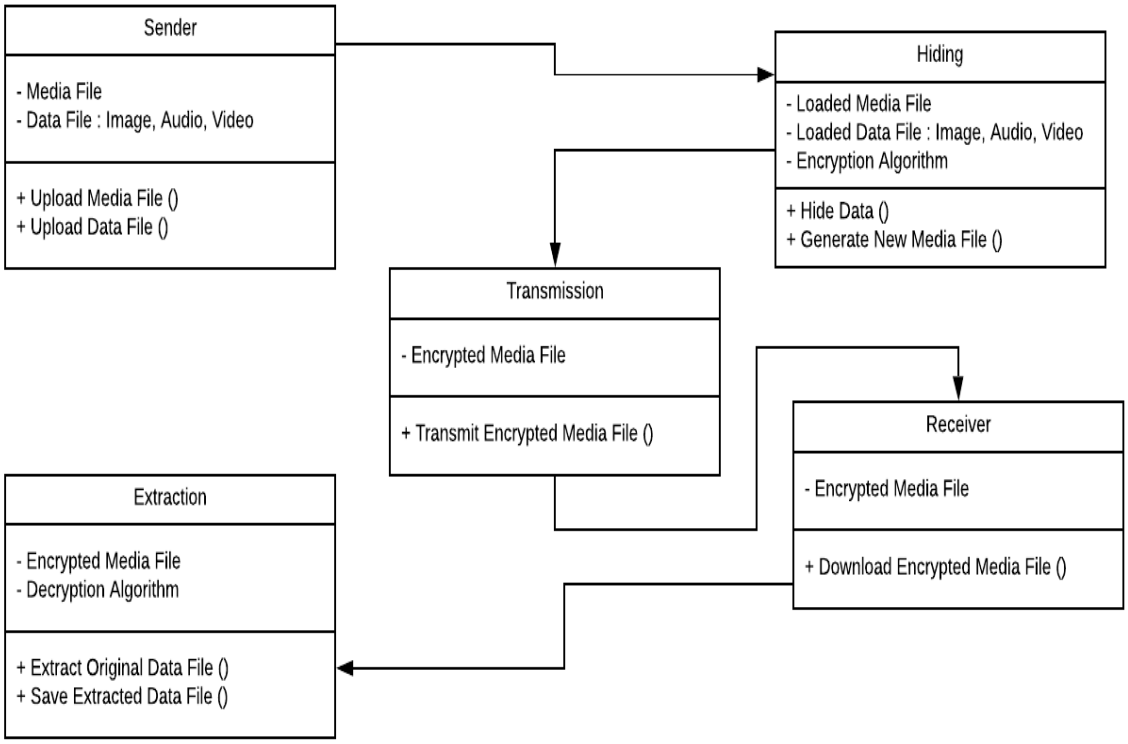
SEQUENCE DIAGRAM:



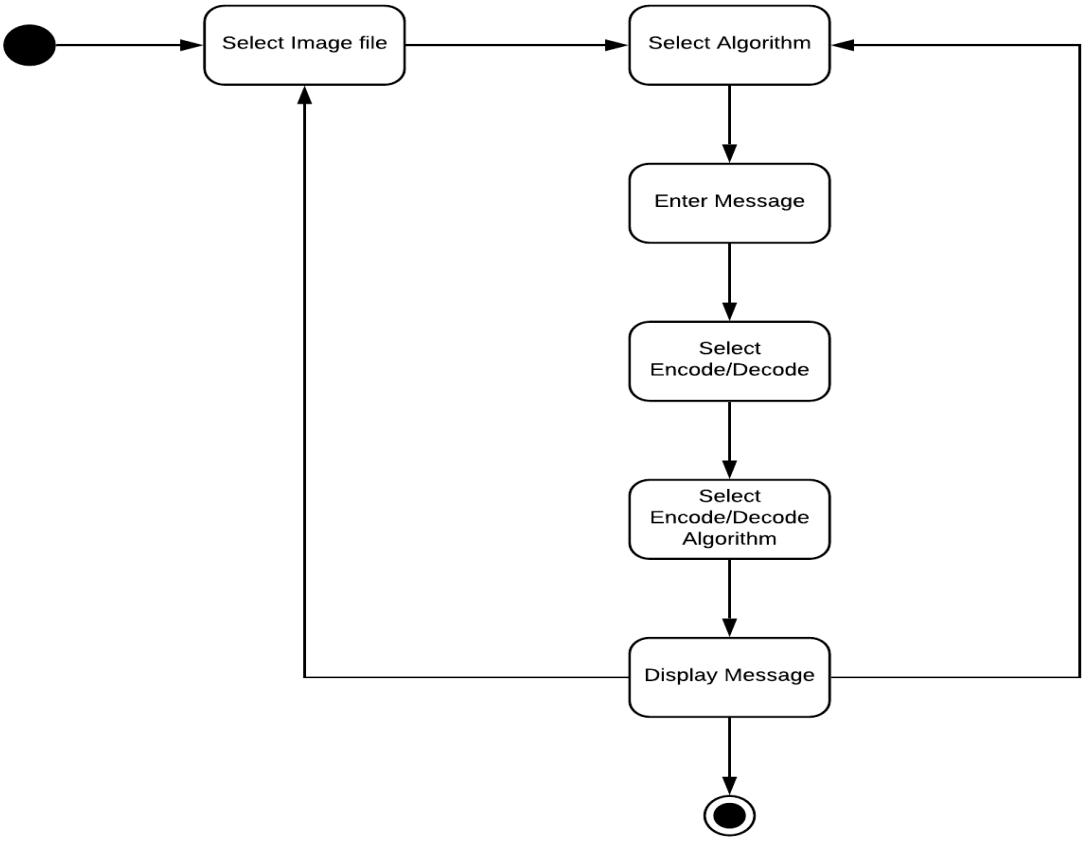
ACTIVITY DIAGRAM:



CLASS DIAGRAM:



STATE DIAGRAM:



Code Analysis

New Jframe.java:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author mohit
 */
public class NewJFrame extends javax.swing.JFrame {

    /**
     * Creates new form NewJFrame
     */
    public NewJFrame() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jButton1.setText("DECODE");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setText("ENCODE");
```

```
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(131, 131, 131)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 128,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 130,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(139, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(52, 52, 52)
                .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 51,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(60, 60, 60)
                .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(85, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton2ActionPerformed
        EmbedMessage a=new EmbedMessage();

                a.setVisible(true);
    }//GEN-LAST:event_jButton2ActionPerformed

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton1ActionPerformed
        NewClass a=new NewClass();

                a.setVisible(true);
    }//GEN-LAST:event_jButton1ActionPerformed
```

```
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVE
RE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new NewJFrame().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
```

```
    // End of variables declaration//GEN-END:variables
}
```

NewClass.java:

```
//DecodeMessage.java
 import java.awt.image.*;
 import javax.swing.*;
 import java.awt.*;
 import java.awt.event.*;
 import javax.imageio.*;

 public class NewClass extends JFrame implements ActionListener
 {
 JButton open = new JButton("Open"), decode = new JButton("Decode"),
    reset = new JButton("Reset");
 JTextArea message = new JTextArea(10,3);
 BufferedImage image = null;
 JScrollPane imagePane = new JScrollPane();

 public NewClass() {
    super("Decode stegonographic message in image");
    assembleInterface();
    this.setSize(500, 500);
    this.setLocationRelativeTo(null);
    this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
 //   this.setBounds(GraphicsEnvironment.getLocalGraphicsEnvironment().
   //    getMaximumWindowBounds());
    this.setVisible(true);
    }

 private void assembleInterface() {
    JPanel p = new JPanel(new FlowLayout());
    p.add(open);
    p.add(decode);
    p.add(reset);
    this.getContentPane().add(p, BorderLayout.NORTH);
    open.addActionListener(this);
    decode.addActionListener(this);
    reset.addActionListener(this);
    open.setMnemonic('O');
    decode.setMnemonic('D');
    reset.setMnemonic('R');

    p = new JPanel(new GridLayout(1,1));
    p.add(new JScrollPane(message));
    message.setFont(new Font("Arial",Font.BOLD,20));
```

```
  p.setBorder(BorderFactory.createTitledBorder("Decoded message"));
  message.setEditable(false);
  this.getContentPane().add(p, BorderLayout.SOUTH);

  imagePane.setBorder(BorderFactory.createTitledBorder("Steganographed Image"));
 this.getContentPane().add(imagePane, BorderLayout.CENTER);
  }
public void actionPerformed(ActionEvent ae) {
  Object o = ae.getSource();
  if(o == open)
    openImage();
  else if(o == decode)
    decodeMessage();
  else if(o == reset)
    resetInterface();
  }

private java.io.File showFileDialog(boolean open) {
  JFileChooser fc = new JFileChooser("Open an image");
  javax.swing.filechooser.FileFilter ff = new javax.swing.filechooser.FileFilter() {
    public boolean accept(java.io.File f) {
      String name = f.getName().toLowerCase();
      return f.isDirectory() ||   name.endsWith(".png") || name.endsWith(".bmp");
      }
    public String getDescription() {
      return "Image (*.png, *.bmp)";
      }
    };
  fc.setAcceptAllFileFilterUsed(false);
  fc.addChoosableFileFilter(ff);

  java.io.File f = null;
  if(open && fc.showOpenDialog(this) == fc.APPROVE_OPTION)
    f = fc.getSelectedFile();
  else if(!open && fc.showSaveDialog(this) == fc.APPROVE_OPTION)
    f = fc.getSelectedFile();
  return f;
 }

private void openImage() {
  java.io.File f = showFileDialog(true);
  try {
    image = ImageIO.read(f);
    JLabel l = new JLabel(new ImageIcon(image));
   imagePane.getViewport().add(l);
    this.validate();
    } catch(Exception ex) { ex.printStackTrace(); }
```

```
  }

private void decodeMessage() {
  int len = extractInteger(image, 0, 0);
  byte b[] = new byte[len];
  for(int i=0; i<len; i++)
    b[i] = extractByte(image, i*8+32, 0);
  message.setText(new String(b));
  }

private int extractInteger(BufferedImage img, int start, int storageBit) {
 int maxX = img.getWidth(), maxY = img.getHeight(),
    startX = start/maxY, startY = start - startX*maxY, count=0;
  int length = 0;
  for(int i=startX; i<maxX && count<32; i++) {
    for(int j=startY; j<maxY && count<32; j++) {
      int rgb = img.getRGB(i, j), bit = getBitValue(rgb, storageBit);
      length = setBitValue(length, count, bit);
      count++;
      }
    }
  return length;
  }

private byte extractByte(BufferedImage img, int start, int storageBit) {
  int maxX = img.getWidth(), maxY = img.getHeight(),
    startX = start/maxY, startY = start - startX*maxY, count=0;
  byte b = 0;
  for(int i=startX; i<maxX && count<8; i++) {
    for(int j=startY; j<maxY && count<8; j++) {
      int rgb = img.getRGB(i, j), bit = getBitValue(rgb, storageBit);
      b = (byte)setBitValue(b, count, bit);
      count++;
      }
    }
  return b;
  }

private void resetInterface() {
  message.setText("");
  imagePane.getViewport().removeAll();
  image = null;
  this.validate();
  }

private int getBitValue(int n, int location) {
  int v = n & (int) Math.round(Math.pow(2, location));
```

```
    return v==0?0:1;
    }

  private int setBitValue(int n, int location, int bit) {
    int toggle = (int) Math.pow(2, location), bv = getBitValue(n, location);
    if(bv == bit)
      return n;
    if(bv == 0 && bit == 1)
      n |= toggle;
    else if(bv == 1 && bit == 0)
      n ^= toggle;
    return n;
    }

  public static void main(String arg[]) {
    new NewClass();
    }
  }
```

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author mohit
 */
public class Main {
    public static void main(String arg[]) {
    NewJFrame a=new NewJFrame();
    a.setVisible(true);
    a.setLocationRelativeTo(null);
    }
}
```

Main.java:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author mohit
 */
```

```
public class Main {
   public static void main(String arg[]) {
   NewJFrame a=new NewJFrame();
   a.setVisible(true);
   a.setLocationRelativeTo(null);
   }
}
```

Embed Message.java:

```
//EmbedMessage.java
 import java.awt.image.*;
 import javax.swing.*;
 import java.awt.*;
 import java.awt.event.*;
 import javax.imageio.*;

 public class EmbedMessage extends JFrame implements ActionListener
 {
 JButton open = new JButton("Open"), embed = new JButton("Embed"),
   save = new JButton("Save into new file"), reset = new JButton("Reset");
 JTextArea message = new JTextArea(10,3);
 BufferedImage sourceImage = null, embeddedImage = null;
 JSplitPane sp = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT);
 JScrollPane originalPane = new JScrollPane(),
   embeddedPane = new JScrollPane();

 public EmbedMessage() {
   super("Embed stegonographic message in image");
   assembleInterface();

//    this.setBounds(GraphicsEnvironment.getLocalGraphicsEnvironment().
//       getMaximumWindowBounds());
   this.setSize(500, 500);
   this.setLocationRelativeTo(null);
   this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
   this.setVisible(true);
   sp.setDividerLocation(0.5);
   this.validate();
   }

 private void assembleInterface() {
   JPanel p = new JPanel(new FlowLayout());
   p.add(open);
   p.add(embed);
   p.add(save);
```

```
  p.add(reset);
  this.getContentPane().add(p, BorderLayout.SOUTH);
  open.addActionListener(this);
  embed.addActionListener(this);
 save.addActionListener(this);
  reset.addActionListener(this);
  open.setMnemonic('O');
  embed.setMnemonic('E');
  save.setMnemonic('S');
  reset.setMnemonic('R');

  p = new JPanel(new GridLayout(1,1));
  p.add(new JScrollPane(message));
  message.setFont(new Font("Arial",Font.BOLD,20));
  p.setBorder(BorderFactory.createTitledBorder("Message to be embedded"));
  this.getContentPane().add(p, BorderLayout.NORTH);

  sp.setLeftComponent(originalPane);
  sp.setRightComponent(embeddedPane);
  originalPane.setBorder(BorderFactory.createTitledBorder("Original Image"));
  embeddedPane.setBorder(BorderFactory.createTitledBorder("Steganographed Image"));
  this.getContentPane().add(sp, BorderLayout.CENTER);
  }

public void actionPerformed(ActionEvent ae) {
  Object o = ae.getSource();
  if(o == open)
    openImage();
  else if(o == embed)
    embedMessage();
  else if(o == save)
    saveImage();
  else if(o == reset)
    resetInterface();
  }

private java.io.File showFileDialog(final boolean open) {
  JFileChooser fc = new JFileChooser("Open an image");
  javax.swing.filechooser.FileFilter ff = new javax.swing.filechooser.FileFilter() {
    public boolean accept(java.io.File f) {
      String name = f.getName().toLowerCase();
      if(open)
        return f.isDirectory() || name.endsWith(".jpg") || name.endsWith(".jpeg") ||
          name.endsWith(".png") || name.endsWith(".gif") || name.endsWith(".tiff") ||
          name.endsWith(".bmp") || name.endsWith(".dib");
      return f.isDirectory() || name.endsWith(".png") ||    name.endsWith(".bmp");
      }
```

```
      public String getDescription() {
        if(open)
          return "Image (*.jpg, *.jpeg, *.png, *.gif, *.tiff, *.bmp, *.dib)";
        return "Image (*.png, *.bmp)";
        }
      };
    fc.setAcceptAllFileFilterUsed(false);
    fc.addChoosableFileFilter(ff);

    java.io.File f = null;
   if(open && fc.showOpenDialog(this) == fc.APPROVE_OPTION)
      f = fc.getSelectedFile();
    else if(!open && fc.showSaveDialog(this) == fc.APPROVE_OPTION)
      f = fc.getSelectedFile();
    return f;
    }

  private void openImage() {
    java.io.File f = showFileDialog(true);
    try {
      sourceImage = ImageIO.read(f);
      JLabel l = new JLabel(new ImageIcon(sourceImage));
      originalPane.getViewport().add(l);
      this.validate();
      } catch(Exception ex) { ex.printStackTrace(); }
    }

  private void embedMessage() {
    String mess = message.getText();
    embeddedImage = sourceImage.getSubimage(0,0,
      sourceImage.getWidth(),sourceImage.getHeight());
    embedMessage(embeddedImage, mess);
    JLabel l = new JLabel(new ImageIcon(embeddedImage));
    embeddedPane.getViewport().add(l);
    this.validate();
    }

  private void embedMessage(BufferedImage img, String mess) {
    int messageLength = mess.length();

    int imageWidth = img.getWidth(), imageHeight = img.getHeight(),
      imageSize = imageWidth * imageHeight;
    if(messageLength * 8 + 32 > imageSize) {
      JOptionPane.showMessageDialog(this, "Message is too long for the chosen image",
        "Message too long!", JOptionPane.ERROR_MESSAGE);
      return;
      }
```

```
  embedInteger(img, messageLength, 0, 0);

  byte b[] = mess.getBytes();
  for(int i=0; i<b.length; i++)
    embedByte(img, b[i], i*8+32, 0);
  }

private void embedInteger(BufferedImage img, int n, int start, int storageBit) {
  int maxX = img.getWidth(), maxY = img.getHeight(),
    startX = start/maxY, startY = start - startX*maxY, count=0;
  for(int i=startX; i<maxX && count<32; i++) {
    for(int j=startY; j<maxY && count<32; j++) {
      int rgb = img.getRGB(i, j), bit = getBitValue(n, count);
      rgb = setBitValue(rgb, storageBit, bit);
      img.setRGB(i, j, rgb);
      count++;
      }
    }
  }

private void embedByte(BufferedImage img, byte b, int start, int storageBit) {
  int maxX = img.getWidth(), maxY = img.getHeight(),
    startX = start/maxY, startY = start - startX*maxY, count=0;
  for(int i=startX; i<maxX && count<8; i++) {
    for(int j=startY; j<maxY && count<8; j++) {
      int rgb = img.getRGB(i, j), bit = getBitValue(b, count);
      rgb = setBitValue(rgb, storageBit, bit);
      img.setRGB(i, j, rgb);
      count++;
      }
    }
  }

private void saveImage() {
  if(embeddedImage == null) {
    JOptionPane.showMessageDialog(this, "No message has been embedded!",
      "Nothing to save", JOptionPane.ERROR_MESSAGE);
    return;
    }
 java.io.File f = showFileDialog(false);
  String name = f.getName();
  String ext = name.substring(name.lastIndexOf(".")+1).toLowerCase();
  if(!ext.equals("png") && !ext.equals("bmp") &&   !ext.equals("dib")) {
      ext = "png";
      f = new java.io.File(f.getAbsolutePath()+".png");
      }
  try {
```

```
      if(f.exists()) f.delete();
      ImageIO.write(embeddedImage, ext.toUpperCase(), f);
      } catch(Exception ex) { ex.printStackTrace(); }
    }

  private void resetInterface() {
    message.setText("");
    originalPane.getViewport().removeAll();
    embeddedPane.getViewport().removeAll();
    sourceImage = null;
    embeddedImage = null;
    sp.setDividerLocation(0.5);
    this.validate();
    }

  private int getBitValue(int n, int location) {
    int v = n & (int) Math.round(Math.pow(2, location));
    return v==0?0:1;
    }

  private int setBitValue(int n, int location, int bit) {
    int toggle = (int) Math.pow(2, location), bv = getBitValue(n, location);
    if(bv == bit)
      return n;
    if(bv == 0 && bit == 1)
      n |= toggle;
    else if(bv == 1 && bit == 0)
      n ^= toggle;
    return n;
    }

  public static void main(String arg[]) {
    new EmbedMessage();
    }
}
```

Decode Message.java:

```
//DecodeMessage.java

 import java.awt.image.*;
 import javax.swing.*;
 import java.awt.*;
 import java.awt.event.*;
 import javax.imageio.*;
```

```
public class DecodeMessage extends JFrame implements ActionListener
{
JButton open = new JButton("Open"), decode = new JButton("Decode"),
  reset = new JButton("Reset");
JTextArea message = new JTextArea(10,3);
BufferedImage image = null;
JScrollPane imagePane = new JScrollPane();

public DecodeMessage() {
  super("Decode stegonographic message in image");
  assembleInterface();

  this.setDefaultCloseOperation(EXIT_ON_CLOSE);
  this.setBounds(GraphicsEnvironment.getLocalGraphicsEnvironment().
    getMaximumWindowBounds());
  this.setVisible(true);
  }

private void assembleInterface() {
  JPanel p = new JPanel(new FlowLayout());
  p.add(open);
  p.add(decode);
  p.add(reset);
  this.getContentPane().add(p, BorderLayout.NORTH);
  open.addActionListener(this);
  decode.addActionListener(this);
  reset.addActionListener(this);
  open.setMnemonic('O');
  decode.setMnemonic('D');
  reset.setMnemonic('R');

  p = new JPanel(new GridLayout(1,1));
  p.add(new JScrollPane(message));
  message.setFont(new Font("Arial",Font.BOLD,20));
  p.setBorder(BorderFactory.createTitledBorder("Decoded message"));
  message.setEditable(false);
  this.getContentPane().add(p, BorderLayout.SOUTH);

  imagePane.setBorder(BorderFactory.createTitledBorder("Steganographed Image"));
 this.getContentPane().add(imagePane, BorderLayout.CENTER);
  }
public void actionPerformed(ActionEvent ae) {
  Object o = ae.getSource();
  if(o == open)
    openImage();
  else if(o == decode)
    decodeMessage();
```

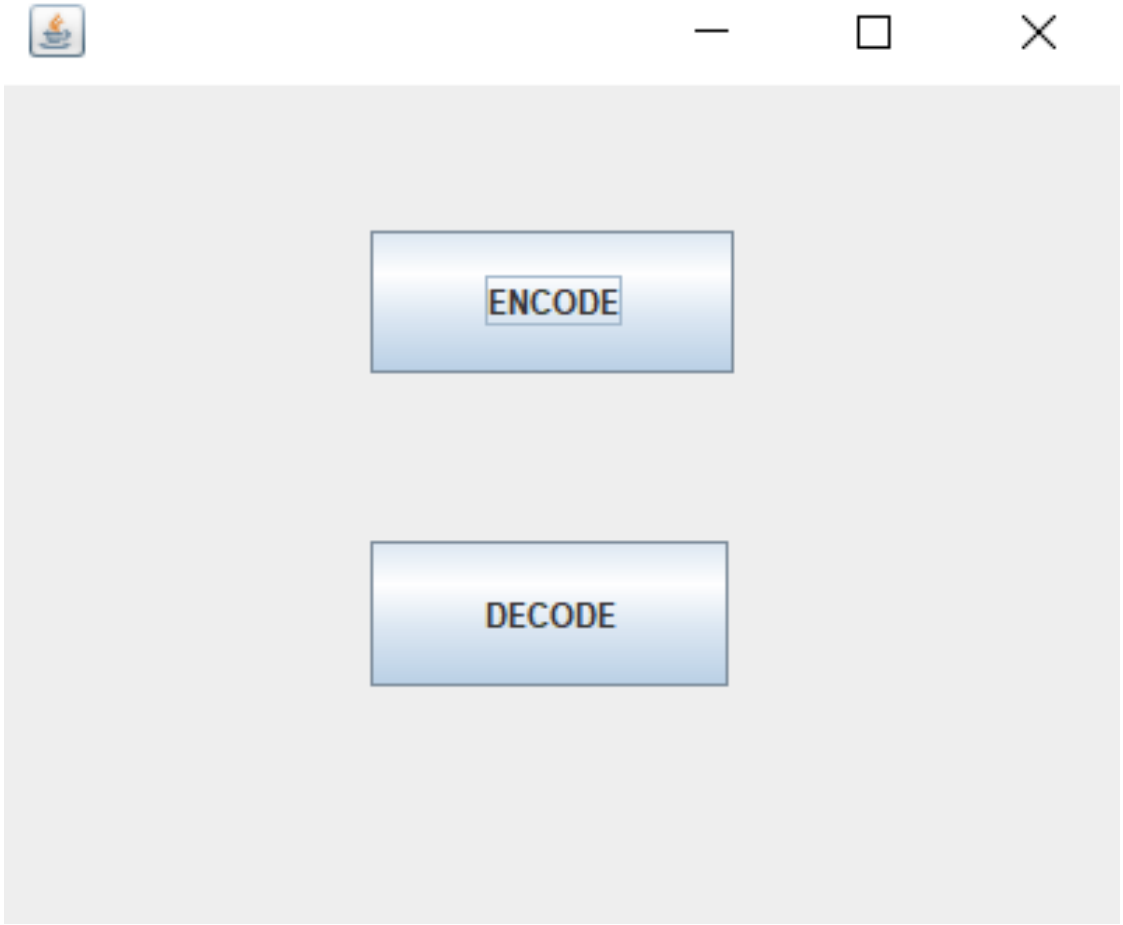
```
  else if(o == reset)
    resetInterface();
  }

private java.io.File showFileDialog(boolean open) {
  JFileChooser fc = new JFileChooser("Open an image");
  javax.swing.filechooser.FileFilter ff = new javax.swing.filechooser.FileFilter() {
    public boolean accept(java.io.File f) {
      String name = f.getName().toLowerCase();
      return f.isDirectory() ||   name.endsWith(".png") || name.endsWith(".bmp");
      }
    public String getDescription() {
      return "Image (*.png, *.bmp)";
      }
    };
  fc.setAcceptAllFileFilterUsed(false);
  fc.addChoosableFileFilter(ff);

  java.io.File f = null;
  if(open && fc.showOpenDialog(this) == fc.APPROVE_OPTION)
    f = fc.getSelectedFile();
  else if(!open && fc.showSaveDialog(this) == fc.APPROVE_OPTION)
    f = fc.getSelectedFile();
  return f;
 }

private void openImage() {
  java.io.File f = showFileDialog(true);
  try {
    image = ImageIO.read(f);
    JLabel l = new JLabel(new ImageIcon(image));
   imagePane.getViewport().add(l);
    this.validate();
    } catch(Exception ex) { ex.printStackTrace(); }
  }

private void decodeMessage() {
  int len = extractInteger(image, 0, 0);
  byte b[] = new byte[len];
  for(int i=0; i<len; i++)
    b[i] = extractByte(image, i*8+32, 0);
  message.setText(new String(b));
  }

private int extractInteger(BufferedImage img, int start, int storageBit) {
 int maxX = img.getWidth(), maxY = img.getHeight(),
    startX = start/maxY, startY = start - startX*maxY, count=0;
```

```
  int length = 0;
  for(int i=startX; i<maxX && count<32; i++) {
    for(int j=startY; j<maxY && count<32; j++) {
      int rgb = img.getRGB(i, j), bit = getBitValue(rgb, storageBit);
      length = setBitValue(length, count, bit);
      count++;
      }
    }
  return length;
  }

private byte extractByte(BufferedImage img, int start, int storageBit) {
  int maxX = img.getWidth(), maxY = img.getHeight(),
    startX = start/maxY, startY = start - startX*maxY, count=0;
  byte b = 0;
  for(int i=startX; i<maxX && count<8; i++) {
    for(int j=startY; j<maxY && count<8; j++) {
      int rgb = img.getRGB(i, j), bit = getBitValue(rgb, storageBit);
      b = (byte)setBitValue(b, count, bit);
      count++;
      }
    }
  return b;
  }

private void resetInterface() {
  message.setText("");
  imagePane.getViewport().removeAll();
  image = null;
  this.validate();
  }

private int getBitValue(int n, int location) {
  int v = n & (int) Math.round(Math.pow(2, location));
  return v==0?0:1;
  }

private int setBitValue(int n, int location, int bit) {
  int toggle = (int) Math.pow(2, location), bv = getBitValue(n, location);
  if(bv == bit)
    return n;
  if(bv == 0 && bit == 1)
    n |= toggle;
  else if(bv == 1 && bit == 0)
    n ^= toggle;
  return n;
  }
```

```
public static void main(String arg[]) {
   new DecodeMessage();
   }
}
```

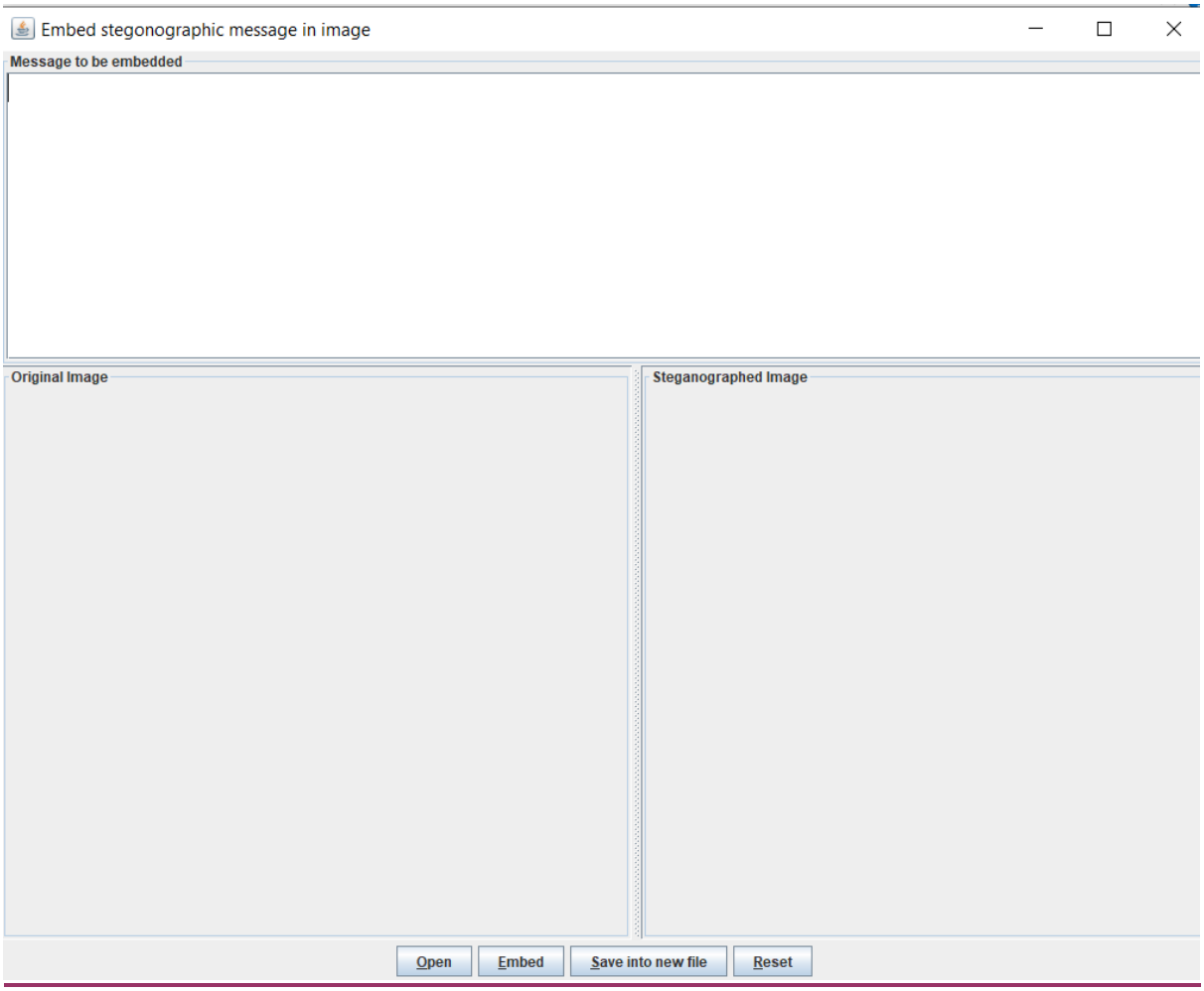
IMPLEMENTATION

This is the first screen which has two tab options – one is Encode for encryption and another is Decode for decryption.

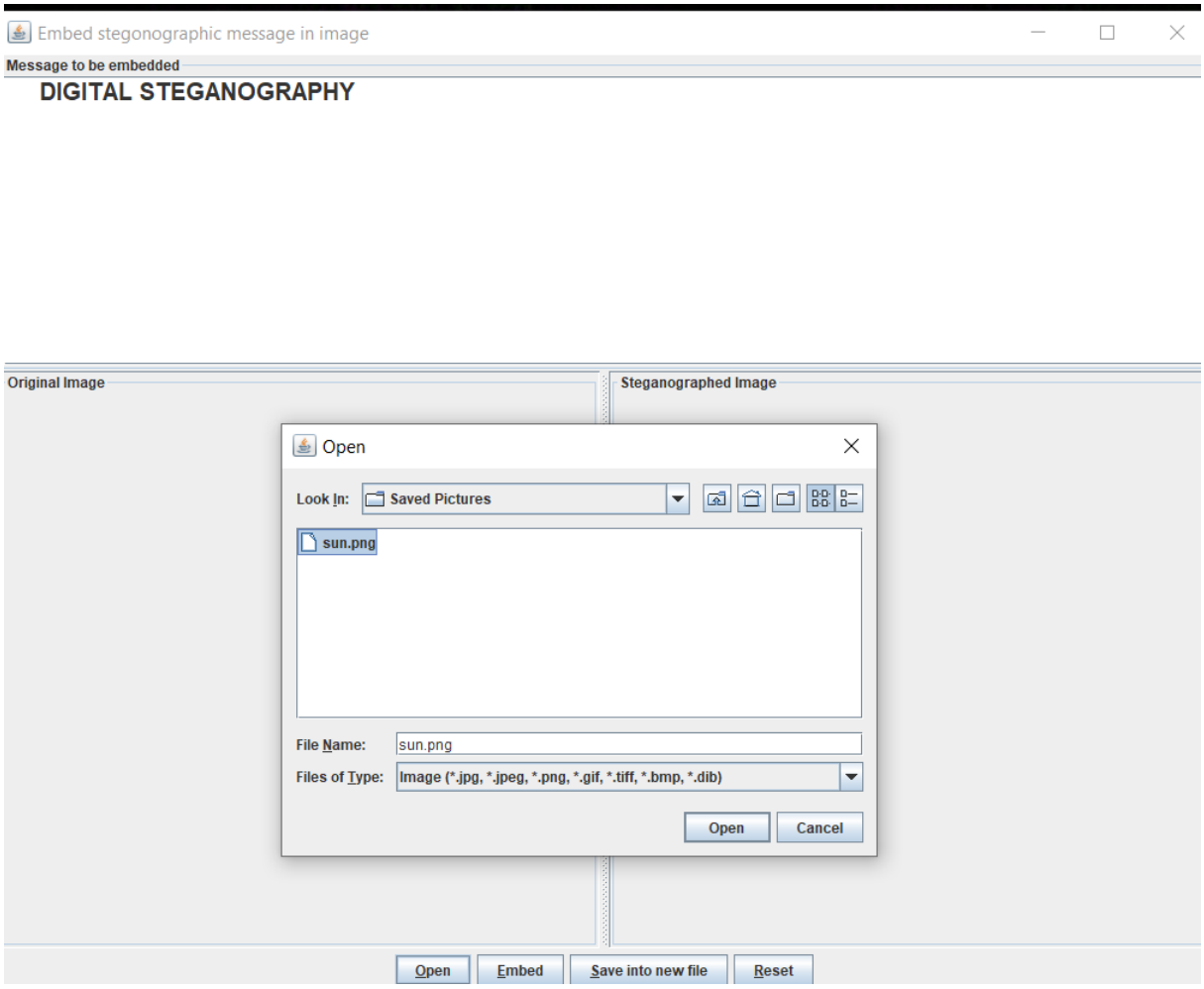


Encryption

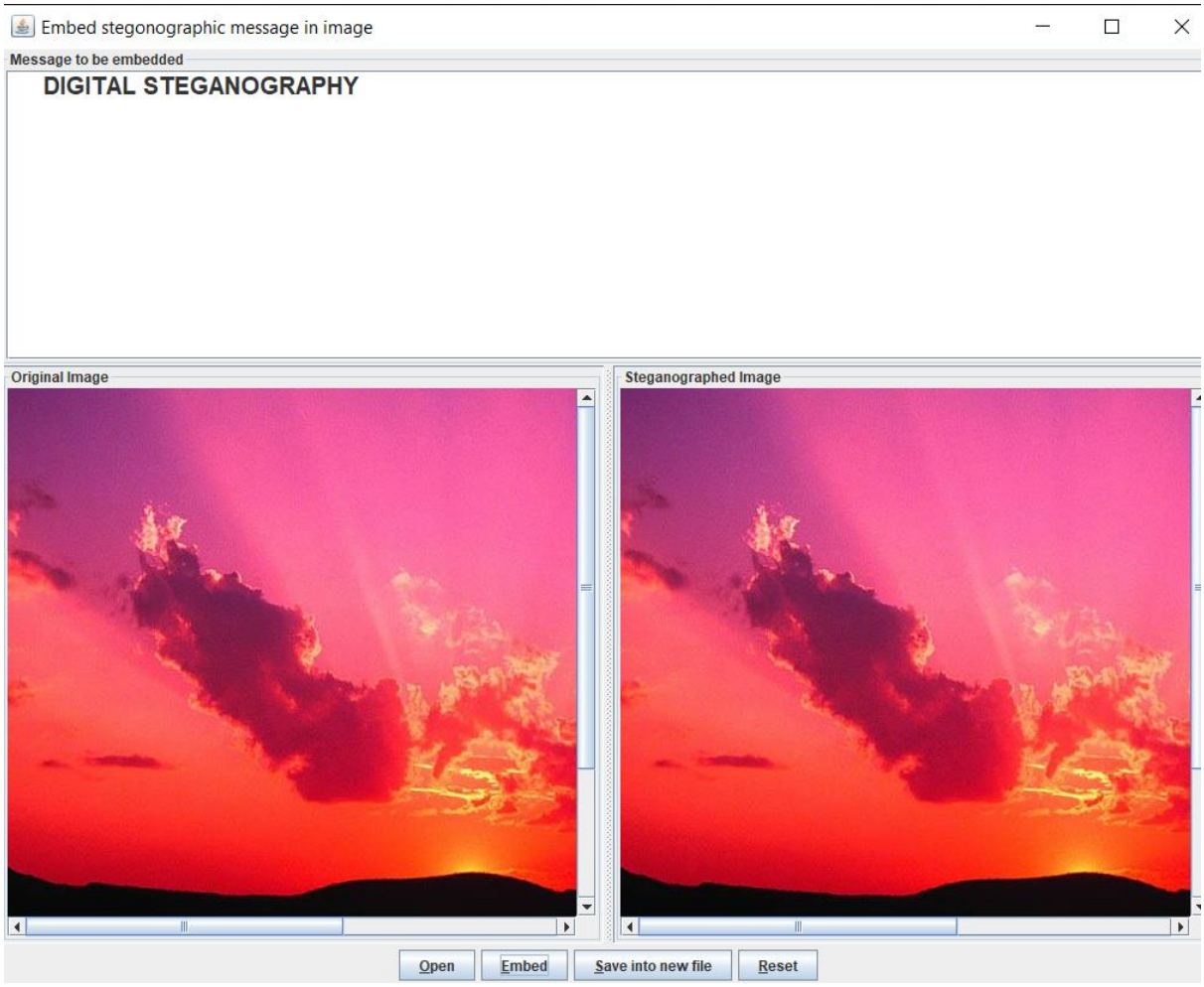
For Encryption select Encode tab option.



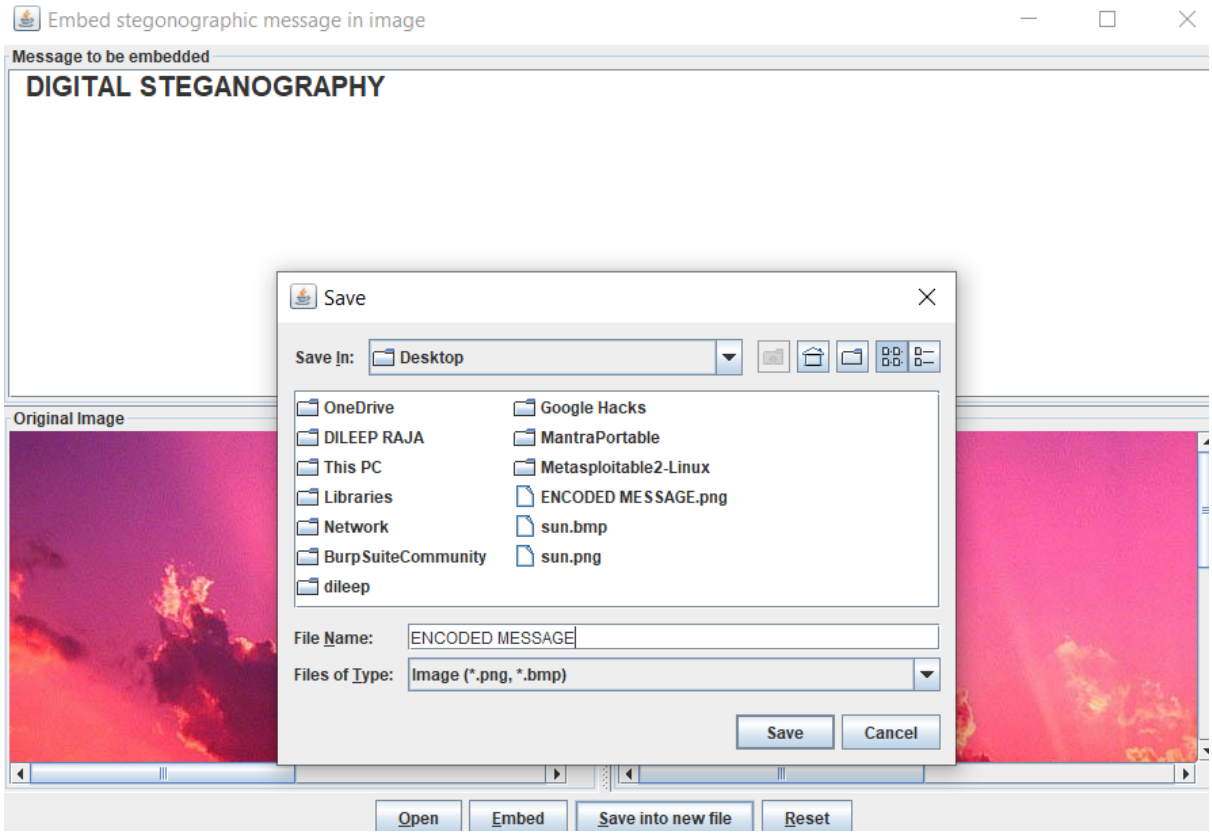
First write the message to be encrypted and For load image click on button “open”. The file open dialog box will displays as follows, select the Image file, which you want to use hide information and click on Open button.



The image file will opened and is displays as follows.

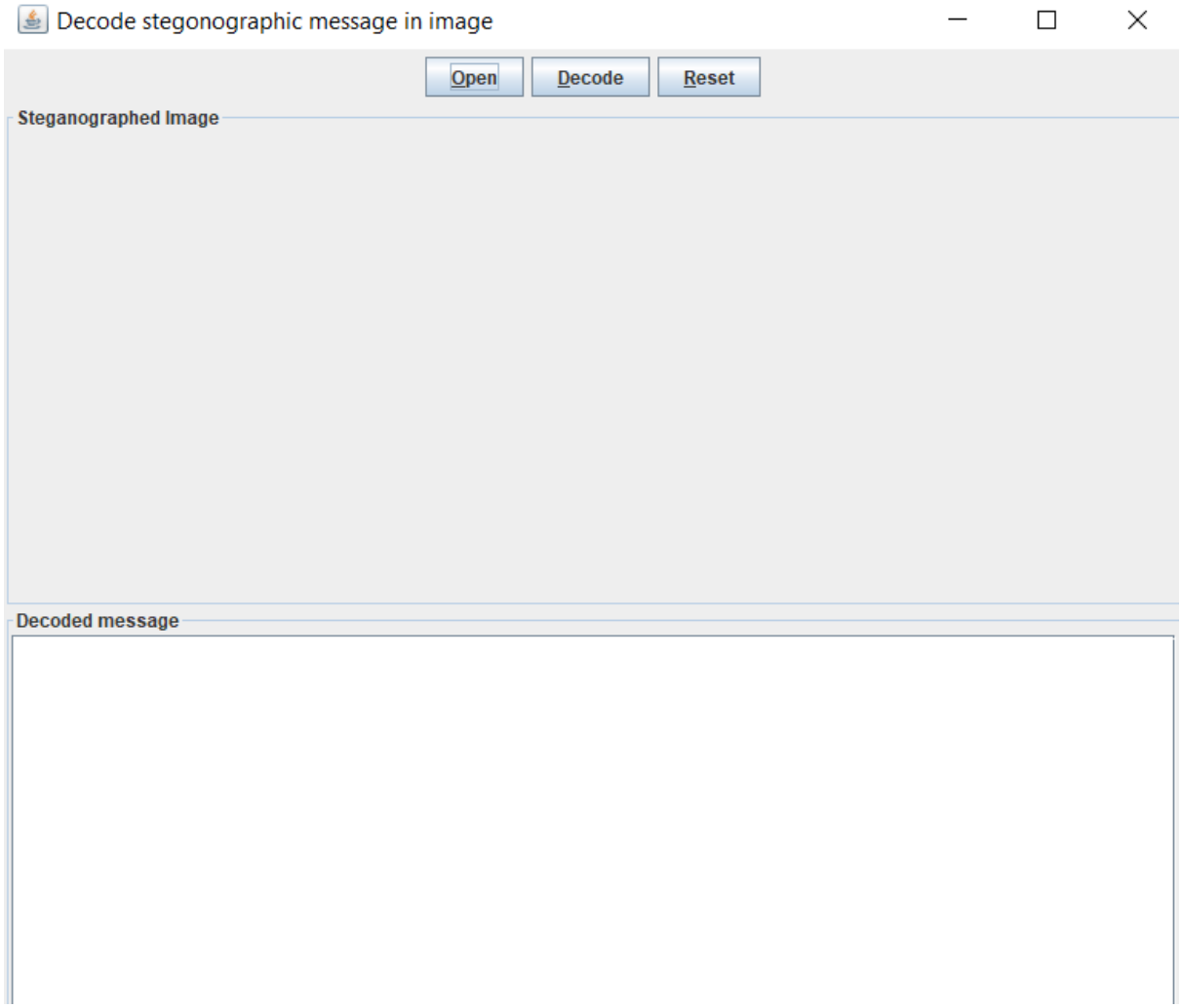


Next, Click on button ”save into new file “ to save Encrypted message as displays and save the file in a desired location

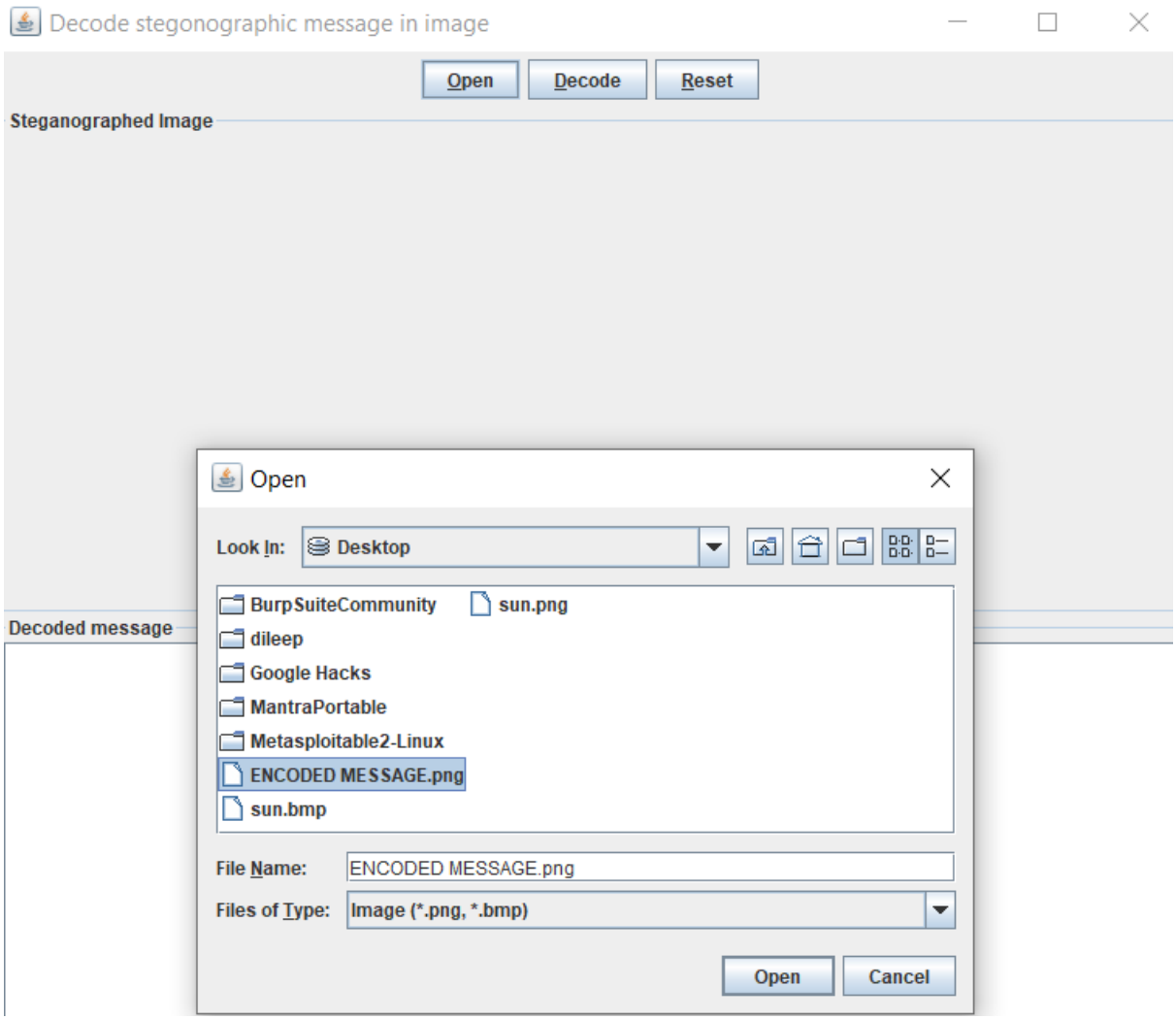


Decryption

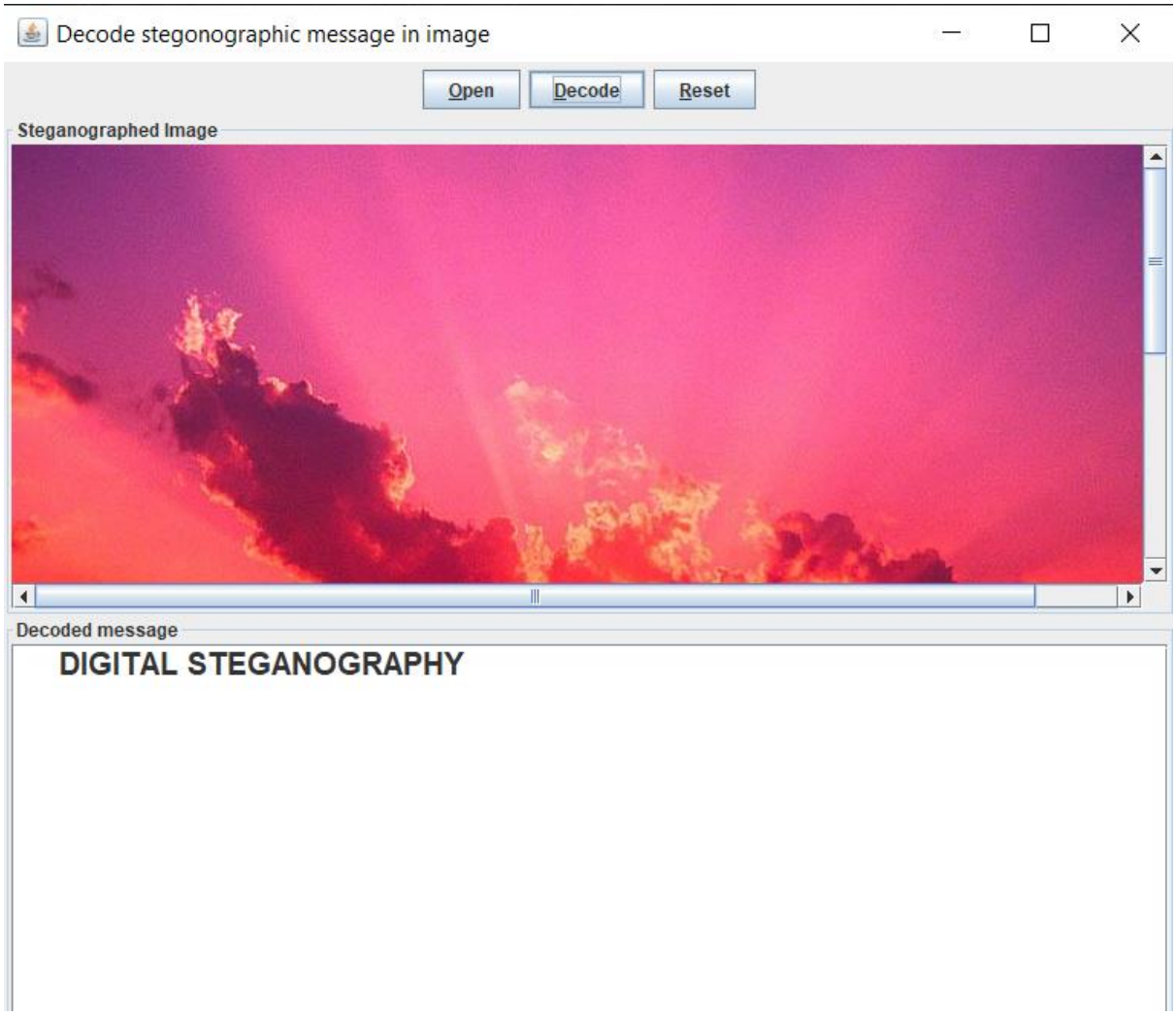
Select the Decode tab option



Click on button “open” and load the encrypted image file to extract the information and click open



Finally, the encrypted message will be displayed.



Conclusion and Future Scope:

To hide confidential information steganography can be effectively used. The objective of any Steganographic method is to hide maximum secret information which is immune to external attacks and also should not convey the fact that the cover medium is carry secret information. This thesis has used LSB substitution technique for time domain and different transforms for frequency domain.

The random key made use of while LSB technique is employed has proved to be better than simple LSB substitution. When the personal is key is made use of in the techniques have not altered the resolution of the image much and appears to be negligible as been seen with the obtained results. Hence the hidden data getting damaged buy the third person is almost impossible. The algorithm can be implemented in both grayscale and colour image since it has made use of 8 bit and 24bit images of size for both cover and secret image.

In Spatial domain methods one can get high payload capacity. The edge detection techniques which are used in the current methods do not recognizes the shades of the edge region, which can also be considered to embed the extra bits. The number of edge pixels can be increased by identifying the edges and shades of edges. The Transform domain methods are highly robust. They embed the message bits in the regions which are highly insensitive to compression, filtration or transformation .But their payload capacity is low. Also the visual qualities of the Stego-image are poor. Spatial domain is better compared to transform domain.

In Spatial Domain techniques simple LSB substitution methods are less secure and payload capacity is less. To increase security Randomization techniques are used. These methods spread the message randomly into the cover image but payload capacity is still less. Stego and crypto way shows new way of embedding the data, especially in Multiresolution analysis, there are different ways of getting Multiresolution; this thesis has made use of Multiresolution analysis on wavelets and Curvelets. The experimental analysis has proved that Curvelets is the best Multiresolution transformation available. This work has been implemented with the library which has an accuracy of 15 fractional digits. The results obtained have a good PSNR value, along with the crypto style of embedding.

Future Enhancement:

In this work it explores only a small part of the science of steganography. As new discipline, there is a great deal more research and development to do The following section describe areas for research which were offshoots of, or tangential ,to our main objectives.

1. Detecting Steganography in Image Files
Can steganography be detected in images files? This is difficult question. It may be possible to detect a simple Steganographic technique by simple
analyzing the low order bits of the image bytes. If the Steganographic algorithm is more complex, however, and spreads the embedded data over the image is random way or encrypts the data before embedding, it may be nearly impossible to detect.

2. How widespread is the Use of Steganography?

If a technique or set of techniques could be devised to detect steganography, it would be interesting to conduct a survey of images available on the internet to determine if steganography is used, by whom and for what purposes. Steganographic applications are available on the Internet, but it is not known if they are being used.

3. Steganography on the World Wide Web

The world wide web(www) makes extensive use of inline images. There are literally millions of images on various web pages worldwide. It may be possible to develop an application to serve as a web browser to retrieve data embedded in web page images. This " stego-web" could operate on top of the existing WWW and be a means of covertly disseminating information.

4. Steganography in printed media.

If the data is embedded in an image, the image printed, then scanned and stored in a file can the embedded data be recovered? This would require a special form of a steganography to which could allow for in accuracies in the printing and scanning equipment.

5. Anti-steganography measures

As was seen in this thesis, JPEG garbles any unencoded stegano graphically embedded data. Also, palettization (mapping a large number of colors in an image to a smaller subset of colors) of an image will it unsuitable for steganography. It is likely, as with JPEG, that some means may be employed to prevent loss of steganographically embedded data when its wrapper file is processed. The question remains open as to what is the most effective anti Steganographic tools or set of tools.

BIBLIOGRAPHY

https://www.w3schools.com

W3Schools is an educational website for learning web technologies online. Contents include tutorials and references relating to HTML, CSS , JAVASCRIPT, JSON, PYTHON, JAVA.

https://www.tutorialspoint.com

Tutorials point originated from the idea that there exists a class of readers who respond better to online content and prefer to learn new skills at their own place form the comforts of their drawing room.

https://www.stackoverflow.com

Stack overflow is a question and answer site for professional and enthusiast programmers.

https://www.github.com

GitHub, Inc**.** is a United States-based global company that provides hosting for software development and version control using Git. It has been a subsidiary of Microsoft since 2018.

https://www.programmer2programmer.com

This web site having programming source code, technical and personal content which is useful for every programmer.