Digital Steganography

****

***A project report submitted to***

***Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal***

***In partial fulfillment for the award of***

***The degree of***

***Bachelor of Engineering***

***In***

***Computer Science & Engineering***

**SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY**

**INDORE- 453331**

**Jul-Dec 2022**

**CERTIFICATE**

This is to certify that **Hemansh Agnihotri (0829CS201077 ) , Kanwaljeet Singh Panwar(0829CS201088)** their project work, titled **“Digital Steganography”**  as per the syllabus and have submitted a satisfactory report on this project as a part of fulfillment towards the degree of **“BACHELOR OF ENGINEERING” (Computer Science & Engineering)**  from **RAJIV GANDHI PROUDYOGIKI VISHWAVIDHYALAYA, BHOPAL.**

**HEAD OF THE DEPARTMENT PROJECT GUIDE**

**DIRECTOR**

|  |
| --- |
| [Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.] |

**(A typical specimen of table of contents)**

<Font Style Times New Roman>

**TABLE OF CONTENTS**

**CHAPTER TITLE PAGE NO.**

**ABSTRACT i**

**LIST OF TABLES ii**

**LIST OF FIGURES iii**

**LIST OF SYMBOLS iv**

1. **Introduction**  1
   1. Purpose of the project
   2. Scope
   3. Problem in existing system
2. **System Requirement Analysis**

2.1 Introduction

2.1.1 Purpose of SRS

2.1.2 Intended Audience and Reading Suggestions

2.1.3 Product Scope

2.2 Overall Description

2.2.1 Product Perspective

2.2.2 Product Functions

2.2.3 User Classes and Characteristics

2.2.4 Operating Environment

2.3 External Interface Requirements

2.3.1 User Interfaces

2.3.2 Hardware Interfaces

2.3.3 Software Interfaces

2.3.4 Communications Interfaces

2.4 Functional Requirement *(description of product functions)*

2.4.1 System Feature 1

2.4.2 System Feature 2 (and so on)

2.5 Nonfunctional Requirements *(any two)*

2.5.1 Performance Requirements

2.5.2 Safety Requirements

2.5.3 Security Requirements

2.5.4 Availability Requirements

2.6 Project Plan

2.6.1 Team Members

2.6.2 Division of Work

2.6.3 Time Schedule *(Show with the help of gantt chart)*

1. **Analysis**
   1. Methodology Used *( Object Oriented)*
   2. Use Case Diagrams *(Also include use case specifications)*
   3. Activity Diagram
   4. Sequence Diagram
   5. Class Diagram *(Description of Identified Classes, Identified Attributes, Identified Methods, Identified Relationships)*

1. **Design**
   1. Architectural Design
      1. System Architecture Diagram
      2. Description Of Architectural Design
   2. Database Design
      1. Normalization
   3. Interface Design
      1. Screenshots
2. **Implementation**
   1. Language and database system used for the implementation
   2. Features of language and database used for the project
   3. Description of third party tools used (If any)
3. **Testing** (*Theory of testing is not required. Only write how u have tested the system)*
   1. White Box Testing (Basis path testing)

Test Case-*n*

1. Purpose
2. Inputs
3. Expected Outputs & Actual Outputs
   1. Black Box Testing
      1. Interface testing

8. **Conclusion**

9. **Reference**s *(include references of the project in IEEE format*)

# Introduction

## Purpose

Data hiding is of importance in many applications. For hobbyists, secretive data transmission, for privacy of users etc. the basic methods are: Steganography and Cryptography. Steganography is a simple security method. Generally there are three different methods used for hiding information: steganography, cryptography, watermarking. In cryptography, the information to be hidden is encoded using certain techniques; this information is generally understood to be coded as the data appears nonsensical. Steganography is hiding information; this generally cannot be identified because the coded information doesn’t appear to be abnormal i.e. its presence is undetectable by sight. Detection of steganography is called Steganalysis.

Steganography is of different types:

1. Text steganography

2. Image steganography

3. Audio steganography

4. Video steganography

In all of these methods, the basic principle of steganography is that a secret message is to be embedded in another cover object which may not be of any significance in such a way that the encrypted data would finally display only the cover data. So it cannot be detected easily to be containing hidden information unless proper decryption is used.

* 1. **Scope**

Our project scope is developed for hiding information in any image file to ensure the safety of exchange the data between different parties and provide better security during message transmission.

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. We will use LSB technique; the proposed approach is to use the suitable algorithm for embedding the data in an image files.

**1.3 Problem in existing system**

This project addresses the security problem of transmitting the data over internet network, the main idea coming when we start asking that how can we send a message secretly to the destination? The science of steganography answers this question. Using steganography, information can be hidden in carriers such as images, audio files, text files, videos and data transmissions. In this document, we proposed some methods and algorithms of an image steganography system to hide a digital text of a secret message.

# Introduction

## Purpose

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 simpler and user friendly.

## Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

## Intended Audience and Reading Suggestions

The document is intended for requirements skill-based, domain expert, professional, non- Professional and it is restricted within the college premises. Before reading this document it

Highly recommended to read the Abstract document to get an overview of the product.

## Product 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.

## References

• Concepts

<http://en.wikipedia.org/wiki/Steganography>

http://en.wikipedia.org/wiki/StegoText

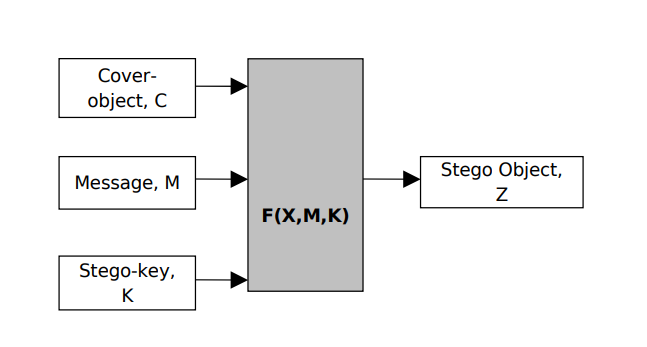
# Overall Description

## Product Perspective

The Steganography product is designed to provide free encryption/decryption to its end users. The product works with Windows, Mac and Linux and intends to provide the user basics of steganography. The product is designed to provide a consistent encryption and decryption feature with alternative backup file measure so that all the activities are secure in terms of use. 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.

## Product Functions

Its functions includes Users will able to perform various types of steganography like text inside image, image inside image, text inside audio files, text inside video files

To perform steganography we need a cover object which can be an image, and message to be hidden inside the cover object, and a key for encryption of the image.

## User Classes and Characteristics

There are 4 classes in this Project: Encrypt, Decrypt, Util, Gui

**Gui** is the main class it has all need GUI elements defined needed for the project. It is also a main class

**Encrypt** class handles the encoding part this is it take the cover object and hide the required text message into cover object lastly this class also handles the task of exporting the file.

**Decrypt** class handles the decryption part that is it handles the part of extracting the hidden message from the cover object.

**Util** class provides the utility functions required to perform the successful encryption and decryption.

## Operating Environment

This software can run on any operation system that is Windows, Linux and Mac.

It require some addition tools to able to use it:

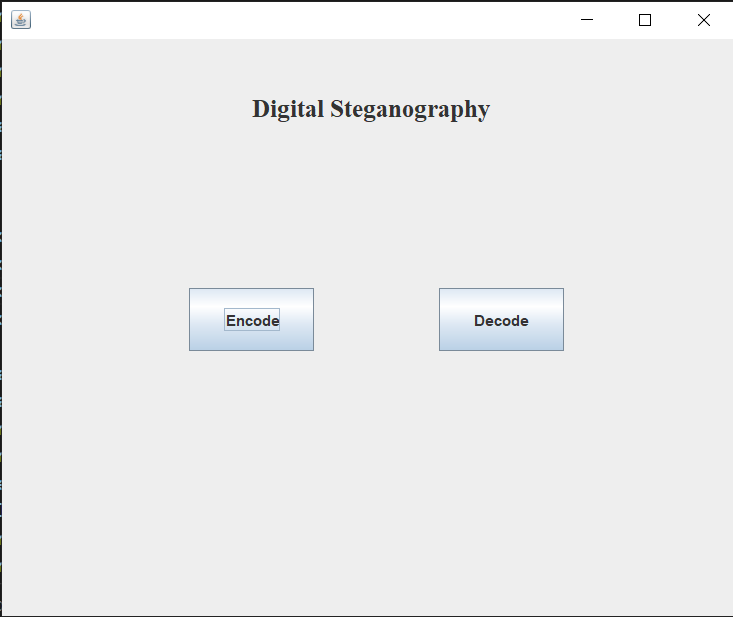
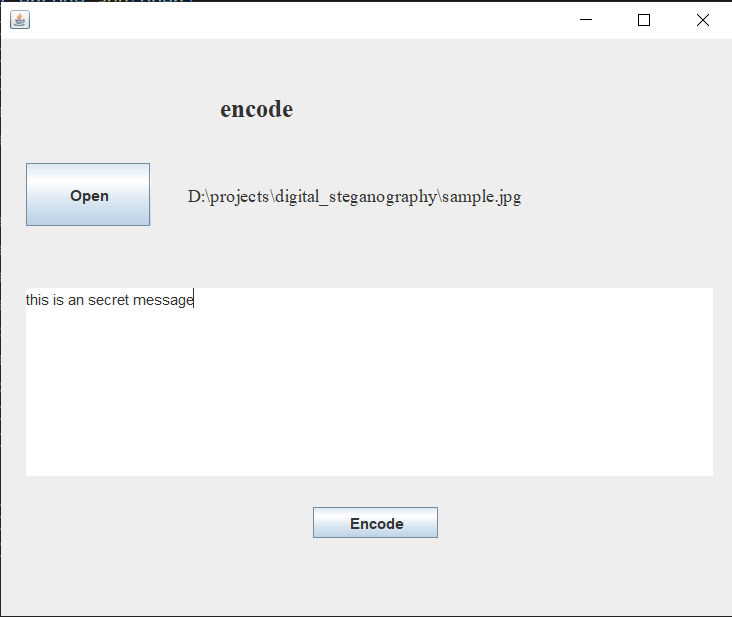
For **JAVA**: User need to install java JRE11+ or JDK 11+ to able to successfully run this software.

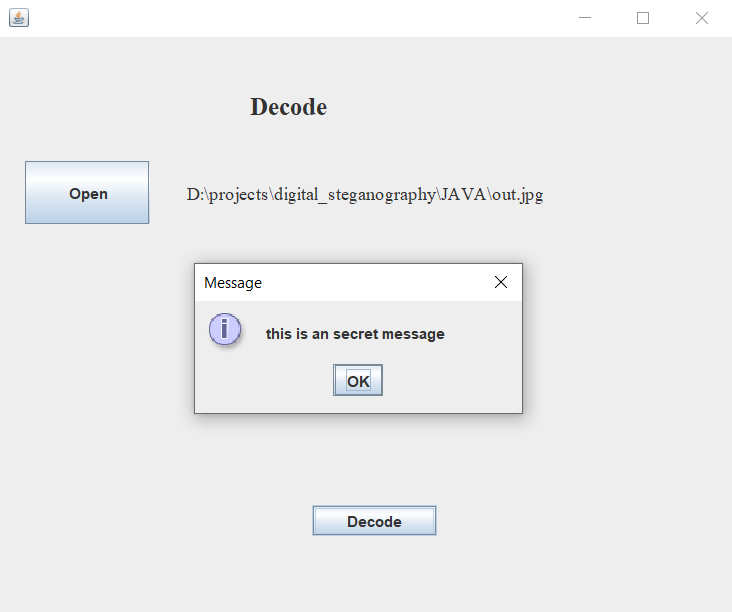
For **C/C++**: User need to install the C/C++ compiler to compile this application.

For **Python:** User will need to install python interpreter and Pillow to be able to use.

# External Interface Requirements

## User Interfaces

Home Screen Encode Screen

 Decode Screen

## Hardware Interfaces

None

## Software Interfaces

Java version relies on Oracle JRE 11+ host system must install java and required tools to be able to run this software.

Python version required Tkinter and Pillow must be install on the system.

C/C++ version require a c/c++ compiler.

## Communications Interfaces

None

# System Features

The product offers the following functions to its users –

• A user friendly Runtime environment.

• Encrypt any text file into any bitmap/JPEG/PNG image file.

• Decrypt any encrypted file to the text file on simple click

• Feature of making files public or private for sharing.

## ENCODING

4.1.1 Description

Provides encoding of the text then we hide the text message inside a cover object which maybe an Image/audio/video file.

4.1.2 Stimulus/Response Sequences

User Must be needed to click on encode button then open the file then just need to click on encode.

4.1.3 Functional Requirements

* Message Requirements

Message must not exceed the image size as not able to fulfill this requirement.

* Cover object

Cover object must be in jpeg/jpg/bmp format for image file.

## DECODING

#### Description

Provides the functionality to users to decode their message from a cover object file.

#### Stimulus/Response Sequence

User must need to click on Decode button then click on open to select the encoded cover object.

#### Functional Requirement

User must ensure that the cover object file which may be for example an image file must not be delete (For example by antivirus software).

# Nonfunctional Requirements

* For user interfaces we take in consideration that they should has a standard look and being user friendly at the same time to make sure that users' attention will not be distracted and interface to provide more flexibility and scalability.
* The program will be in the English language
* The program must be fast in processing
* The program must to hide the image within the image and then extract image from the image properly.
* All function must be works well then system will be a high quality

# Project Plan

## Team Members

1. Hemansh Agnihotri
2. Kanwaljeet Singh Panwar

## Division of Work

1. Hemansh Agnihotri :- Coding, Designing, Documentation
2. Kanwaljeet Singh Panwar :- Idea, Designing, Documentation

## Time Schedule

Not yet decided

**3**. **Analysis Model**

**3.1 Methodology used**

*< Describe approach of the programming language (function oriented or object oriented) used for the development of the product). Also include the description of the Software Process model used for developing the software product.>*

**3.2 Use Case diagram**

*< Use case diagrams are behavior diagrams used to describe a set of actions (*[*use cases*](http://www.uml-diagrams.org/use-case-diagrams.html#use-case)*) that some system or systems (*[*subject*](http://www.uml-diagrams.org/use-case-diagrams.html#subject)*) should or can perform in collaboration with one or more external users of the system (*[*actors*](http://www.uml-diagrams.org/use-case-diagrams.html#actor)*). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system. >*

**Use Case Specification**

|  |  |
| --- | --- |
| Name Of Use Case |  |
| Actor |  |
| Pre Condition |  |
| Primary Flow of Events |  |
| Alternate Flow Of Events | Alternate Flow 1….. |
|  |
| Alternate Flow 2….. |
|  |
| Post Condition |  |
| Use Case termination |  |

**3.3 Activity Diagram**

*<* ***Activity diagram*** *is UML* [*behavior diagram*](http://www.uml-diagrams.org/uml-24-diagrams.html#behavior-diagram) *which shows* ***flow of control*** *or* ***object flow*** *with emphasis on the sequence and conditions of the flow.* ***Activity diagrams*** *are graphical representations of* [*workflows*](http://en.wikipedia.org/wiki/Workflow) *of stepwise activities and actions which are involved in a particular use case or may be a representation of overall activities in a project. In the* [*Unified Modeling Language*](http://en.wikipedia.org/wiki/Unified_Modeling_Language)*, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. >*

**3.4 Sequence Diagram**

*< A* ***sequence diagram*** *in a* [*Unified Modeling Language*](http://en.wikipedia.org/wiki/Unified_Modeling_Language) *(UML) is a kind of* [*interaction diagram*](http://en.wikipedia.org/wiki/Interaction_diagram) *that shows how processes operate with one another and in what order. It is a construct of a* [*Message Sequence Chart*](http://en.wikipedia.org/wiki/Message_Sequence_Chart)*. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams typically are associated with use case realizations in the Logical View of the system under development.>*

**3.5 Class Diagram**

*<In* [*software engineering*](http://en.wikipedia.org/wiki/Software_engineering)*, a* ***class diagram*** *in the* [*Unified Modeling Language*](http://en.wikipedia.org/wiki/Unified_Modeling_Language) *(UML) is a type of static structure diagram that describes the structure of a system by showing the system's* [*classes*](http://en.wikipedia.org/wiki/Class_%28computer_science%29)*, their attributes, operations (or methods), and the relationships among the classes.>*

**4. Design Model**

**4.1 Architectural Design**

**4.1.1 System Architectural Diagram**

*< System architectural Diagram shows how software will be implemented on hardware and also the runtime software elements. In addition it should also include the overall modular program structure and represent the control relationships between modules.>*

**4.1.2 Description of Architectural Diagram**

*< Describe the components that are shown in the Architectural Diagram and their inter relationship.>*

**4.2 Database Design**

**4.2.1 Data Dictionary**

*< The Data Dictionary is an organized listing of all data elements that are pertinent to the system, with precise rigorous definitions so that user and system analyst will have common understanding of inputs, outputs and components of stores. A data dictionary should contain the following information :*

* *Name- the primary name of data and control item, the data store or an external entity.*
* *Alias- other names used for the first entry.*
* *Where used / how used – a listing of processes that uses the data or control item and how it is used.*
* *Content description – a notation for representing content.*
* *Supplementary information – other information about data types, preset values, restriction or limitations, etc.>*

**4.2.1 Normalization**

*<Database normalization is the process of organizing the* [*fields*](http://en.wikipedia.org/wiki/Field_%28computer_science%29) *and* [*tables*](http://en.wikipedia.org/wiki/Table_%28database%29) *of a* [*relational database*](http://en.wikipedia.org/wiki/Relational_database) *to minimize redundancy and dependency. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database via the defined relationships.*

*Apply the normalization process on the tables of your database and also explain the normal form that has been applied on the database>*

**4.3 Component Design**

**4.3.1 Package Diagram**

*<A* ***package diagram*** *in the* [*Unified Modeling Language*](http://en.wikipedia.org/wiki/Unified_Modeling_Language) *depicts the* [*dependencies*](http://en.wikipedia.org/wiki/Dependency_%28UML%29) *between the* [*packages*](http://en.wikipedia.org/wiki/Package_%28UML%29) *that make up a model.****>***

**4.3.2 Component Diagram**

***<****In the* [*Unified Modeling Language*](http://en.wikipedia.org/wiki/Unified_Modeling_Language)*, a* ***component diagram*** *depicts how* [*components*](http://en.wikipedia.org/wiki/Component_%28UML%29) *are wired together to form larger components and or* [*software systems*](http://en.wikipedia.org/wiki/Software_system)*. They are used to illustrate the structure of arbitrarily complex systems.>*

**4.3.2 Deployment Diagram**

*<A* ***deployment diagram*** *in the Unified Modeling Language models the physical deployment of* [*artifacts*](http://en.wikipedia.org/wiki/Artifact_%28UML%29) *on* [*nodes*](http://en.wikipedia.org/wiki/Node_%28UML%29)*.*[*[1]*](http://en.wikipedia.org/wiki/Deployment_diagram#cite_note-0) *To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).>*

**4.4 Interface Design**

**4.4.1 Screen Shots**

*< Attach screen shots of your project here. >*

**5. Implementation**

**5.1 Language and database system used for the implementation**

*<Describe the programming language and database system used for the implementation of the project.>*

**5.2 Features of the Language and database system used for Project**

*<Describe the features of the programming language and database system used for the implementation of the project.>*

**5.3 Reasons for Selecting the Language and database system used for Project**

*<Describe the reasons for selecting the programming language and database system used for the implementation of the project.>*

**5.4 Describe the Third Party Tool Used (if any) in the project**

*<Describe the third party tool (either hardware or software or both) that are used in the implementation of the project.>*

**6. Test Plan**

**6.1 Features To Be Tested**

*<Description of the features (eg. Input domain) to be tested>*

**6.2 Features Not To Be Tested**

*< Description of the features that are ignored or skipped for testing or that will be tested in the later stage>*

**6.3 Testing Tools and Environment**

*< Description of the tools that are used for testing and the purpose for using them.****>***

**6.4. Testing Techniques**

**6.4.1 White Box Testing**

*<Explain white box testing method used for testing the internal structure of the project. Eg. If Cyclomatic testing is used then explain the method for performing cyclomatic complexity, draw neat flow chart and control flow diagram and find out cyclomatic complexity value of the project by all 3 methods.>*

**6.4..2 Black Box Testing**

*<Explain black box testing method used for testing the interfaces of the project. Eg. If boundry value analysis or equivalence partitioning method is used , write down test cases in following given format for each of these method.>*

**6.5 Test Cases and Results**

**6.5.1 *Case n***

*i.* ***Purpose*** *<describe purpose of testing in this case-n>*

*ii.* ***Test Procedure*** *<describe the test procedure applied in this case-n>*

*iii.* ***Input*** *<describe the inputs that are tested in this test case-n>*

*iv.* ***Expected and Actual result*** *<write the expected and actual results of the test>*