**Chapter -1**

**Abstract**

Security often requires that data be kept safe from unauthorized access. And the best line of defense is physical security (placing the machine to be protected behind physical walls). However, physical security is not always an option (due to cost and/or efficiency considerations). Instead, most computers are interconnected with each other openly, thereby exposing them and the communication channels that they use.

Steganography secures information by protecting its confidentiality. It can also be used to protect information about the integrity and authenticity of data. Stronger Steganographic techniques are needed to ensure the integrity of data stored on a machine that may be infected or under attack.

So far Steganography is used in many forms but using it with Audio, Video & Image files is another Stronger Technique. The process of Steganography happens with Audio, Video & Image File for transferring more secure sensitive data. The Sensitive Data is Encoded with an Audio, Video & Image File and Passed over Insecure Channels to other end of Systems. Here we can use any file Format for Encryption and Decryption of Message.

The given message will be encrypted with a given Audio, Video & Image file using a secret key. The System will then embed the secret message into the Audio, Video & Image file. The result will be a new Audio, Video & Image file, which has the secret message in it. While decrypting the same key should be given for encrypted Audio, Video & Image file to get the secret message from it.

**Functionalities:**

Encryption, Decryption with Audio, Video & Image File

Graphical User Interface

**INTRODUCTION**

**INTRODUCTION**

## Purpose

Steganography with Audio, Video & Image is a desktop application. The purpose of this application is to provide the security for the confidential information. This application doest allow the hackers to view the data, can view only Audio, Video & Image file when it is being passed over the internet. Then at the recipient side the original information i.e., plain text will be extracted from the Audio, Video & Image by performing decryption operations.

**1.2 Scope**

The scope of the project is very vast, as it targets large no of people residing over the world. It gained maximum users over the world.

* 1. **PROJECT OVERVIEW**

This application uses java and swings to develop GUI and perform encryption and decryption operations. In the encryption process each first the data will be encrypted by the key which was given by the source and then this encrypted data will be embed into the Audio, Video & Image file and generate the new Audio, Video & Image file which contains the plain text. At the recipient side this data will be extracted and decrypted then gives plain text.

**1.4 PROJECT DESCRIPTION**

Steganography with Audio, Video & Image is a desktop application. The purpose of this application is to provide the security for the confidential information. This application doest allow the hackers to view the data, can view only Audio, Video & Image file when it is being passed over the internet. Then at the recipient side the original information i.e., plain text will be extracted from the Audio, Video & Image by performing decryption operations.

In the Encryption process each LSB (Least Significant Byte) will be replaced by the encrypted data.

During Decryption process each data will be extracted from each LSB and then performs decryption operation which results into a plain text which was sent by the source side.

## Project Features

* It is a platform independent application so that it can be used on any operating system.
* There is no need to download and install as other software.
* Provides security for the information which is passed over the net.
* Information is embedded into the Audio, Video & Image file.

**Chapter -2**

**System Analysis**

**PROBLEM DEFINITION**

**2.1 Existing System**

If a person sends sensitive information over the insecure channels of the system then there may be a chance of hacking it, they can alter the information and sends it over the net. (Example is military persons sending sensitive information over the net.)

This problem has been solved by the proposed system.

**2.2 Proposed System**

In the proposed system the above problem has been solved by embedding the data into the Audio, Video & Image file. Before embedding it into the file, encryption operation will be performed by using the encryption key which is provided by the source. Then this Audio, Video & Image file will be passed over the net, even if hacker hacks it, can be able to see only an Audio, Video & Image file. At the destination side this data will be encrypted from Audio, Video & Image file and performs decryption to get original message.

## 2.3 User Classes and Characteristics

It is desktop application so only one user can use it that who has the computer. User has to browse the Audio, Video & Image file using the GUI which is provided by this application. User enters the data in the data into text box and gives encryption key to encrypt the data. During decryption time again user has to give the decrypted key to get the plain text.

## 2.4 Operating Environment:

Software Requirements are Windows as Operating System, Linux or Mac operating system. Java 2 standard edition and Swings.

Hardware Requirements are P2 above processor, 128MB+ of main memory (RAM) and 100MB hard disk and data base memory

## 2.5 Design and Implementation Constraints

## This project will bedeveloped using the technologies like Java 2 standard edition. Also I’ll be learning clearly about Software Development Life Cycle.

## 2.6 User Documentation

This document also includes a user manual which assists the new user to go about the project, he can even get the online help which caters the needs of a new user and makes this project more user friendly, a step by step approach online makes it easy to use software for a naïve user.

# 

# 2.7 Module Description

**Modules:**

**Graphical User Interface**

**Encryption, Decryption with Audio, Video & Image File**

**GUI Module: -**

This module generates the user interface through which a user browses the Audio, Video & Image file and can play and stop the Audio, Video & Image file. This GUI contains different fields such as text area for entering message and buttons for encryption and decryption.

**Encryption and Decryption Module: -**

During encryption, Audio, Video & Image file will be created and in this Audio, Video & Image file. In this Audio, Video & Image file LSB of the each byte will be replaced by the encrypted data which is generated by the combination of the encryption key and the plain text i.e., the original message. Then this Audio, Video & Image file will be sent to the recipient. At recipient side this encrypted data will be extracted from each LSB and performs decryption operation on it and gives original information.

**2.8 HARDWARE AND SOFTWARE REQUIREMENTS**

**Hard Ware Requirements**

* + - * + Processor:: Pentium-III (or) Higher
        + Ram:: 64MB (or) Higher
        + Cache:: 512MB
        + Hard disk:: 10GB

**Soft Ware Requirements**

* + - Tools:: Micro Soft Front
    - Operating System:: WindowsNT/2000
    - Languages:: Java 2 Standard Edition

**Chapter – 3**

**FEASIBILITY STUDY**

**3. FEASIBILITY STUDY:**

The next step in analysis is to verify the feasibility of the proposed system. “All projects are feasible given unlimited resources and infinite time“. But in reality both resources and time are scarce. Project should confirm to time bounce and should be optimal in there consumption of resources. This place a constant is approval of any project.

Feasibility has applied to **Steganography with Audio, Video & Image** pertains to the following areas:

* Technical feasibility
* Operational feasibility
* Economical feasibility

**3.1** **TECHNICAL FEASIBILITY:**

To determine whether the proposed system is technically feasible, we should take into consideration the technical issues involved behind the system.

Steganography with Audio, Video & Image uses the Java 2 standard edition and swings, which is rampantly employed these days worldwide. The world without the web is incomprehensible today. That goes to proposed system is technically feasible.

**3.2 OPERATIONAL FEASIBILITY:**

To determine the operational feasibility of the system we should take into consideration the awareness level of the users. This system is operational feasible since the users are familiar with the technologies and hence there is no need to gear up the personnel to use system. Also the system is very friendly and to use.

**3.3. ECONOMIC FEASIBILITY**

To decide whether a project is economically feasible, we have to consider various factors as:

* + - * Cost benefit analysis
      * Long-term returns
      * Maintenance costs

The proposed **Steganography with Audio, Video & Image** is computer based. It requires average computing capabilities and access to internet or LAN, which are very basic requirements hence it doesn’t incur additional economic overheads, which renders the system economically feasible.

**Chapter – 4 SYSTEM DESIGN**

**SYSTEM DESIGN**

System design is transition from a user oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

The database tables are designed by analyzing functions involved in the system and format of the fields is also designed. The fields in the database tables should define their role in the system. The unnecessary fields should be avoided because it affects the storage areas of the system. Then in the input and output screen design, the design should be made user friendly. The menu should be precise and compact.

**SOFTWARE DESIGN**

In designing the software following principles are followed:

1. **Modularity and partitioning**: software is designed such that, each system should consists of hierarchy of modules and serve to partition into separate function.

2. **Coupling:** modules should have little dependence on other modules of a system.

3. **Cohesion:** modules should carry out in a single processing function.

4. **Shared use:** avoid duplication by allowing a single module is called by other that need the function it provides

**DataFlow Diagrams**

Data flow diagram is a graphical tool used to describe analyze the movement of data through a system manual or automated including the processes, stores of data, and delays in the system.

Data flow diagrams are the central tool and basis for form which other components are developed. The data flow diagram is also known a data flow graph or bubble chart.

**Context diagram:**

A context diagram is a top level (also known as level 0) data flow diagram. It only contains one process node (process 0) that generalizes the function of the entire system in relationship to external entities.

The top-level diagram is often called a “*context diagram”*. It contains a single process, but it plays a very important role in studying the current system.

The context diagram defines the system that will be studied in the sense that it determines the boundaries.

Anything that is not inside the process identified in the context diagram will not be part of the system study.

It represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows respectively.

**First level DFD:**

The first level DFD shows the main processes within the system. Each of these processes can be broken into further processes until you reach pseudo code. The major processes in our system are issue raising, issue resolving, search process and report generation on issues and generating a first level DFD for each individual field.

**Detailed level DFD:**

This level explains each process of the system in a detailed manner. In first detailed level DFD (generation of individual fields): how data flows through individual process/fields in it are shown. In second detailed level DFD (generation of detailed process of the individual fields): how data flows through the system to form a detailed description of the individual processes.

**System components:**

Symbolizes process

Symbolizes data flow

Symbolizes External entity

Symbolizes data store

The data flow diagram is one of the most important tools used for the system analysis. ADEMACRO (1978) and SARSON (1979) populated the use of data flow diagrams as modeling tool through their structured analysis methodologies. They suggested that a data flow diagram should be the first tool used by the analyst to model the system components. There are four types of system components. They are

**Process:**

Process shows what system does. A process is represented by a circle as shown above. Each process is given a unique name and a unique number. Each process takes one or more data inputs and produces one or more data outputs.

**Data flows:**

Data flows made the passage of data and are represented by the line joining the system components. An arrow is used to indicate the direction of data flow shown in the above and the line is labeled by the name of the data flow.

**Data stores:**

Data store is used to represent the repository of the data that maintains in the system. A process can

**External Entities:**

External entities are the out side the system but they either supply input into system or use system output. The designer has to control over these entities. These are represented by a square or rectangle outputs as shown in the above.

The context level data flow diagram shown above gives a brief idea about the flow of data should be and it is also shows the control if some data are invalid.

They are represented by:

* **LEVEL – 0 :** SYSTEM INPUT/OUTPUT
* **LEVEL – 1 :** SUBSYSTEM LEVEL DATAFLOW FUNCTIONAL
* **LEVEL – 2 :** DETAIL DATAFLOW

The input and output data shown should be consistent from one level to the next.

**Level – 0 System Input/Output:**

A context diagram is a top level (also known as Level 0) data flow diagram. It only contains one process node (process 0) that generalizes the function of the entire system in relationship to external

User

**Level-1 Subsystem Level:**

A level – 1 DFD describes the next level of details within the system, detailing the data flows between subsystems, which makeup the whole.

User

**Level 2: Detailed Level**

**Encryption:**

Encryption

**Decryption:-**

Decryption

**5.2UML DIAGRAMS**

A diagram is the graphical presentation of a set of elements most often rendered as a connected graph of vertices (things) and arcs (relationships).The UML includes some diagrams:

1. **Class diagram** shows a set of classes, interfaces, and collaborations and their relationships. Class diagrams address the static design view of a system. Class diagrams that include active classes address the static process view of a system.



2. **Use case diagram** shows a set of use case and actors (a special kind of class) and their relationships. Use case diagrams address the static use case view of a system. These diagrams are especially important in organizing and modeling the behavior of a system.



3. **Sequence diagram** is an interaction diagram that emphasizes the time-ordering of messages

**Encryption:**



**Decryption: -**



4. **Activity Diagram:** They are a loosely defined diagram technique for showing workflows of stepwise activities and actions, with support for choice, iteration and concurrency.



**5.2 INPUT/OUTPUT DESIGN**

**Input design:** considering the requirements, procedures to collect the necessary input data in most efficiently designed. The input design has been done keeping in view that, the interaction of the user with the system being the most effective and simplified way.

**Output design:** All the screens of the system are designed with a view to provide the user with easy operations in simpler and efficient way, minimum key strokes possible. Instructions and important information is emphasized on the screen. Almost every screen is provided with no error and important messages and option selection facilitates. Emphasis is given for speedy processing and speedy transaction between the screens. Each screen assigned to make it as much user friendly as possible by using interactive procedures. So to say user can operate the system without much help from the operating manual.

**Chapter -6**

**IMPLEMENTATION**

**OVERVIEW OF SOFTWARE DEVELOPMENT TOOLS**

**Java And Its Features**

**J**ava was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank and Mike Sheridan at SUN Microsystems Incorporation in the year 1991.It took 18 months to develop the 1st working version. This language was initially called “OAK”, but was renamed “JAVA” in 1995, many more contributed to the design and evolution of the language.

**JAVA OVERVIEW**

**J**ava is a powerful but lean object-oriented programming language. It has generated a lot of excitement because it makes it possible to program for Internet by creating Applets. Programs that can be embedded in web page. The context of an applet can be an animation with sound, an interactive game or a ticker tape. With constantly updated stock prices. Applets can be just little decorations to liven up web page, or they can be serious applications like Word processor or Spreadsheet.

But Java is more than a programming language for writing Applets. It is being used more and more for writing standalone applications as well. It is becoming so popular that many people believe it will become standard language for both general purpose and Internet programming.

There are many buzzwords associated with Java, but because of its spectacular growth in popularity, a new buzzword has appeared ubiquitous. Indeed, all indications are that it will soon be everywhere.

Java builds on the strength of C++. It has taken the best features of C++ and discarded the more problematic and error prone parts. To this lean core, it has added garbage collection (automatic memory management), multithreading (the capacity for one program to do more than one thing at a time), security capabilities. This result is that Java is simple, elegant, and powerful and easy-to-use.

Java is actually a platform consisting of 3 components:

Java Programming Language.

Java Library of Classes and Interfaces.

Java Virtual Machine

The following sections will say more about these components.

**JAVA IS PORTABLE:**

One of the biggest advantages Java offers is that it is portable. An application written in Java will run on all the major platforms. Any computer with a Java-based browser can run the applications or Applets written in the Java-Programming-Language. A programmer no longer has to write one program to run on a Macintosh, another program to run on a Windows-machine still another to run on a UNIX-machine and so on. In other words, with Java developers write their programs only once.

The Virtual Machine is what gives Java is cross platform capabilities. Rather being compiled into machine language, which is different for each OS’s and computer architecture, Java code is compiled into Byte codes. With other languages, the program code is compiled into a language that the computer can understand. The problem is that other computers with different machine instruction set cannot understand that language. Java code on the other hand is compiled into Byte-Code rather than a machine language. These byte codes go to the JVM, which executes them directly or translates them into the language that is understood by the machine running it.

In summary, these means that with the JDBC API extending Java, a programmer writing Java code can access all the major RDBMS on any platform that supports the JVM.

JAVA IS OBJECT-ORIENTED

The Java programming language is OBJECT-ORIENTED, which makes program design focus on what you are dealing with, rather than on how your are going to do something. This makes it more useful for programming in sophisticated projects, because one can break the things into understandable components. A big benefit is that these components can then be reused.

Object-Oriented Languages use the paradigm of classes. In simplest term, a class includes both the data and the functions to operate on data. You can create an instance of a class, also called an object, which will have all the data members and functionality of its class. Because of this, you can think of a class as being like template, with each object being a specific instance of a particular type of class.

The class paradigm allows one to encapsulate data so that specific data values are those using the data cannot see the function implementation. Encapsulation makes it possible to make the changes in code without breaking other programs that use that code.

If for example, the implementation of a function is changed, the change is invisible to any programmer who invokes that function, and does not affect his/her program, except hopefully to improve it.

Java includes inheritance, or the ability to derive new classes from existing classes. The derived class, is also called as Sub-Class, inherits all the data in the functions of the existing class.

**JAVA DEVOLPMENT EVNIRONMENT**

To code, edit, debug and test the java programs, one needs to have a java development environment. At the minimum this will consists of a java compiler interpreter and applet viewer where applets can be tested. Sun’s java development kit (JDK) latest version is 2.2 can be freely downloaded from the Internet. Java compiler is available on DOS, Win95, WIN’NT, Solaris and MAC etc.

**SERVER:**

ServerSocket serverSocket = null;

try {

serverSocket = new ServerSocket(4445);

} catch (IOException e) {

System.err.println("Could not listen on port: 4445.");

System.exit(1);

}

Socket clientSocket = null;

try {

clientSocket = serverSocket.accept();

} catch (IOException e) {

System.err.println("Accept failed.");

System.exit(1);

}

**CLIENT:**

Socket kkSocket = null;

PrintWriter out = null;

BufferedReader in = null;

System.out.println(args[0]);

try {

kkSocket = new Socket(args[0], 4445);

out = new PrintWriter(kkSocket.getOutputStream(), true);

in = new BufferedReader(new InputStreamReader(kkSocket.getInputStream()));

} catch (UnknownHostException e) {

System.err.println("Exception " + e);

System.exit(1);

} catch (IOException e) {

System.err.println("Couldn't get I/O for the connection.");

System.exit(1);

}

BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in));

String fromServer;

String fromUser;

**PROTOCOL:**

Runtime rt=Runtime.getRuntime();

Process p=rt.exec("cmd /c "+ args);

InputStream is = p.getInputStream();

BufferedReader br = new BufferedReader(new InputStreamReader(is));

while ((sb = br.readLine()) != null)

{

sa += sb ;

}

**Chapter -7**

**TESTING**

**7 .SOFTWARE TESTING**

Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

#### 7.1 TESTING OBJECTIVES

* + To ensure that during operation the system will perform as per specification.
  + TO make sure that system meets the user requirements during operation
  + To make sure that during the operation, incorrect input, processing and output will be detected
  + To see that when correct inputs are fed to the system the outputs are correct
  + To verify that the controls incorporated in the same system as intended
  + Testing is a process of executing a program with the intent of finding an error
  + A good test case is one that has a high probability of finding an as yet undiscovered error

The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers an as yet undiscovered error.

Note that the result of the system testing will prove that the system is working correctly. It will give confidence to system designer, users of the system, prevent frustration during implementation process etc.,

## 7.2 TEST CASE DESIGN:

## White box testing

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independents path in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exits the current menu.

## 

## Black Box Testing

Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

The following are the different tests at various levels:

**Unit Testing:**

Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the module/program. In the Generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

**Integration Testing:**

All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

## Validation Testing

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

## System Testing

This testing is a series of different tests whose primary is to fully exercise the computer-based system. This involves:

* Implementing the system in a simulated production environment and testing it.
* Introducing errors and testing for error handling.

**TEST CASES**

|  |  |  |
| --- | --- | --- |
| **Test case 1:** Verifying Encryption**.** | | **Priority (H, L):** High |
| **Test Objective:** For Verifying Encryption**.** | | |
| **Test Description:** “User browses for a master file, enter message, encryption key and presses Encryption button”, Encryption Program will be called and takes the browsed master file and performs encryption and embeds into output file. | | |
| **Requirements Verified:** Yes | | |
| **Test Environment:**  Java software should be installed; | | |
| **Test Setup/Pre-Conditions:**  Java software should be installed; path and CLASSPATH should be set. Audio, Video & Image file should be of .wav format. | | |
| Actions | Expected Results | |
| User browses master file, enter message, encryption key and presses Encryption button. | Displays Message. | |
| **Pass: yes Conditions pass: yes**  **Fail**: No | | |
| **Problems / Issues:** NIL | | |
| **Notes**: Successfully Executed | | |

**TEST CASE 1:**

**Test case for verifying encryption:**

|  |  |  |
| --- | --- | --- |
| **Test case 1:** Verifying Decryption**.** | | **Priority (H, L):** High |
| **Test Objective:** For Verifying Decryption**.** | | |
| **Test Description:** “User browses the output file, enter password and presses Retrieve now button”, Decryption Program will be called and takes the browsed file and performs Decryption displays Original information. | | |
| **Requirements Verified:** Yes | | |
| **Test Environment:**  Java software should be installed; | | |
| **Test Setup/Pre-Conditions:**  Java software should be installed; path and CLASSPATH should be set. | | |
| Actions | Expected Results | |
| User browses the file, enter decryption key and presses Retrieve Now button | Displays original Message. | |
| **Pass: yes Conditions pass: yes**  **Fail**: No | | |
| **Problems / Issues:** NIL | | |
| **Notes**: Successfully Executed | | |

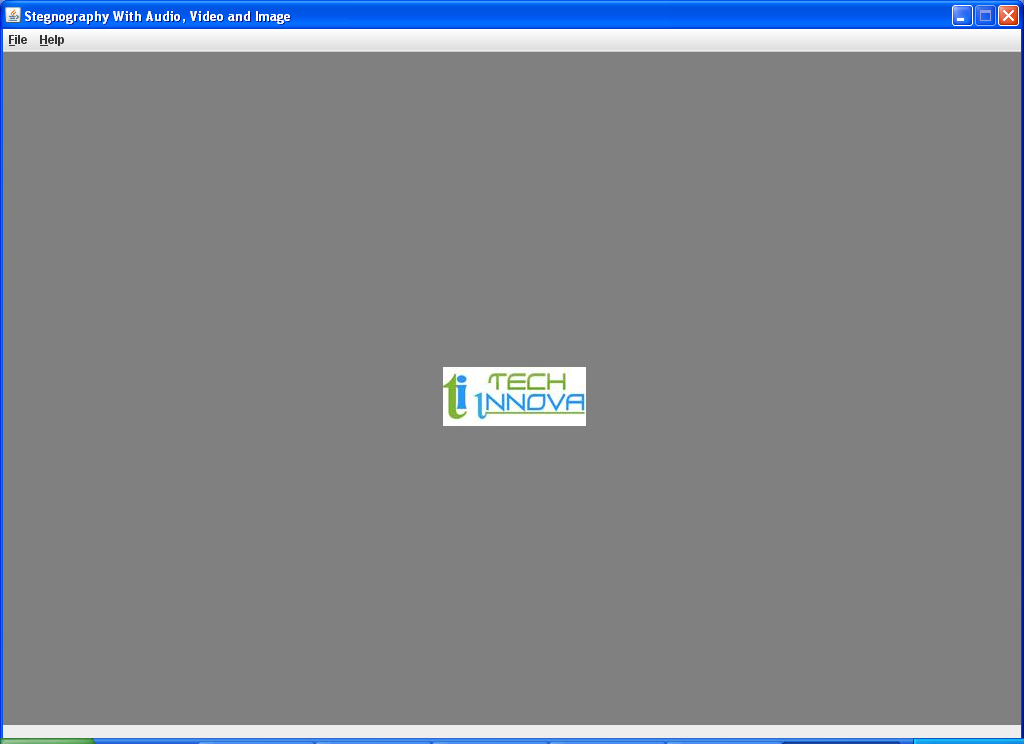
**Test case for verifying decryption:**

**Chapter -8**

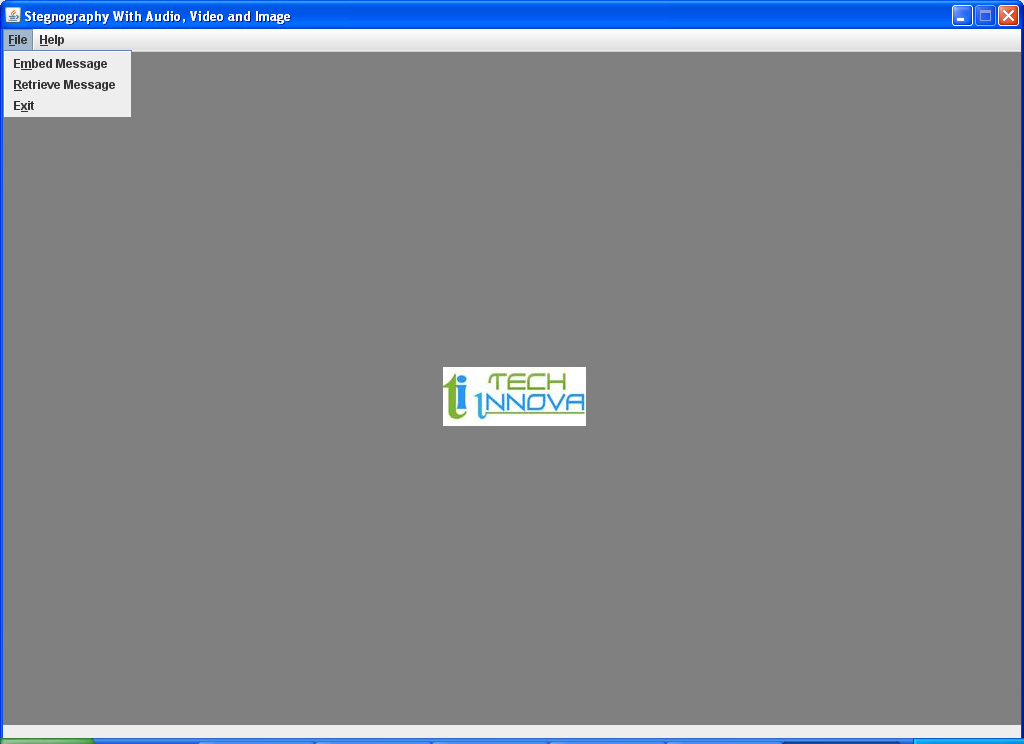
**OUTPUT SCREENS**

8. OUTPUT SCREENS

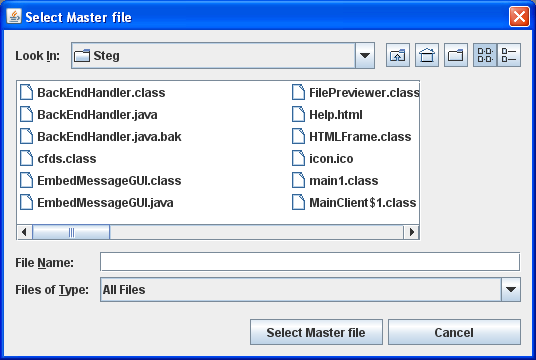
1. Main Screen of the application



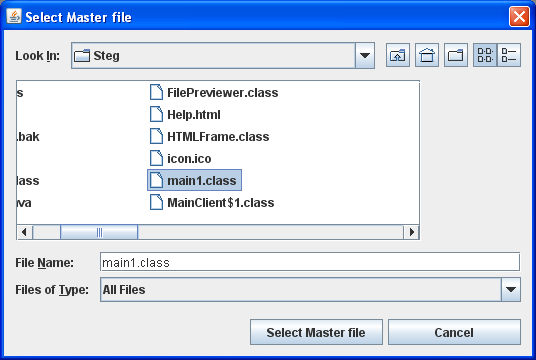
1. When the user has selected the file menu. A dropdown menu is displayed with Embed Message, Retrive Message & Exit options.



3. & 4. When the user has selected the Embed Message option has selected then this window is opened here we have to select the Master File .

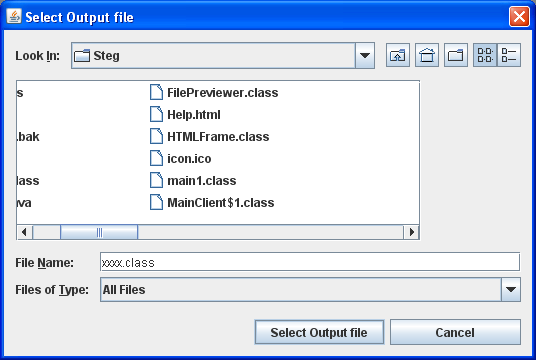


4.

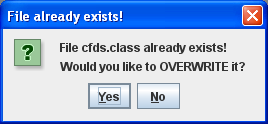


5. After selecting the Main File the user has to select the output file.

(i) If the output file is not there in the present directory then a file will be created.

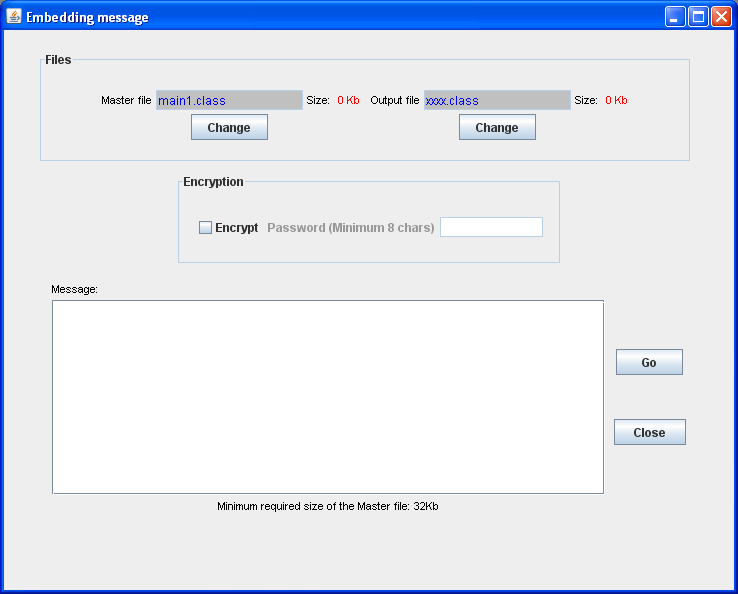


(ii) If the output file exists then the below dialog box will be displayed. If **“yes”** is clicked then the file will be overwritten. If **“No”** is clicked then again we have to select other file name.

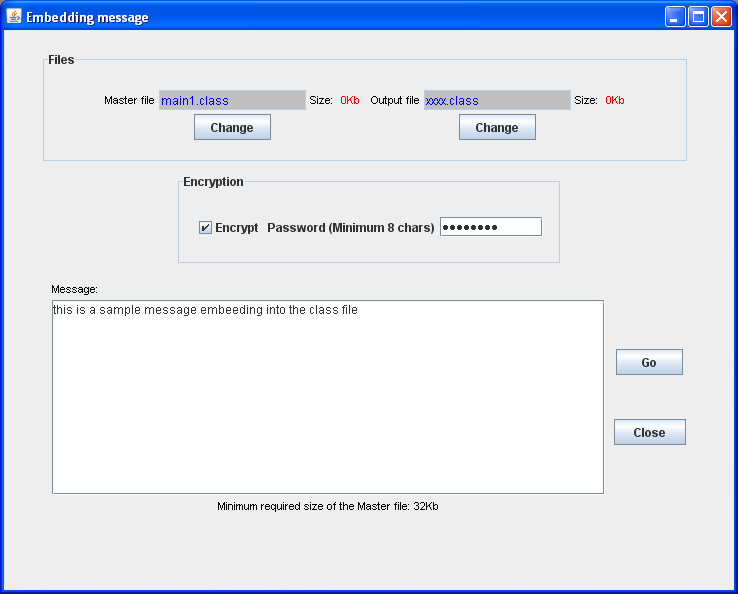


6 (i) (ii) (iii). After selecting the output file the below window will be opened. In this window if user wants to create a password then user has to check the check box before **“Encrypt”** label & he has to set a password length of minimum 8 characters. In the message user has to enter message what he want to send. After entering the message user has to click the **“GO”**  button.

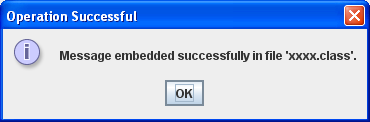
6 (i)



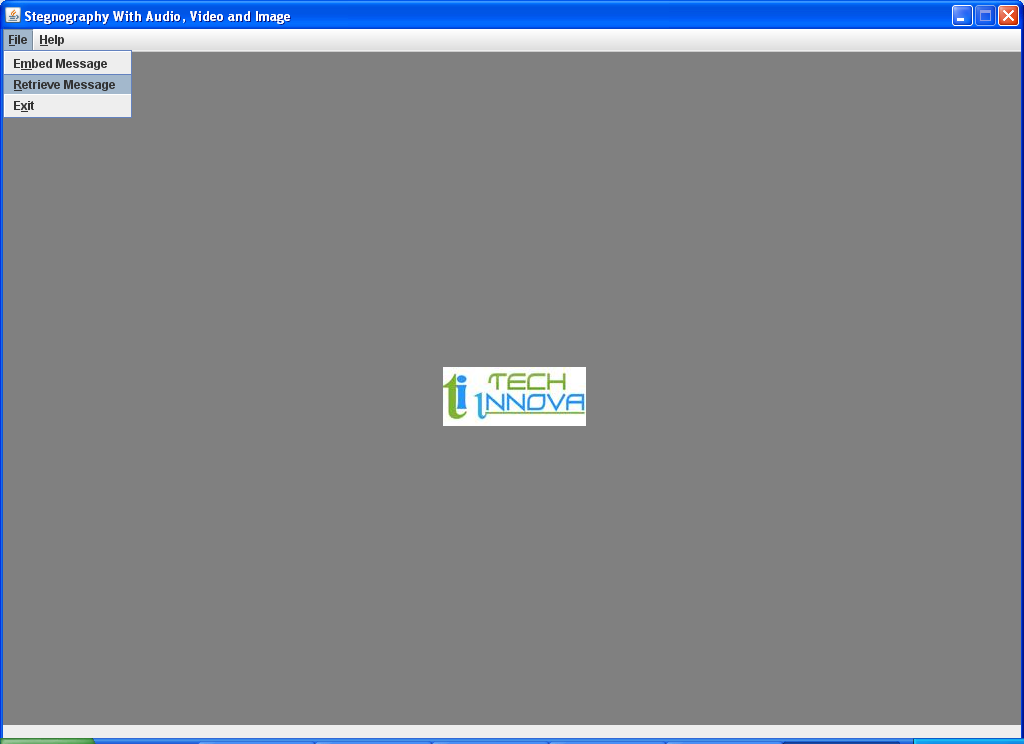
6 (ii)



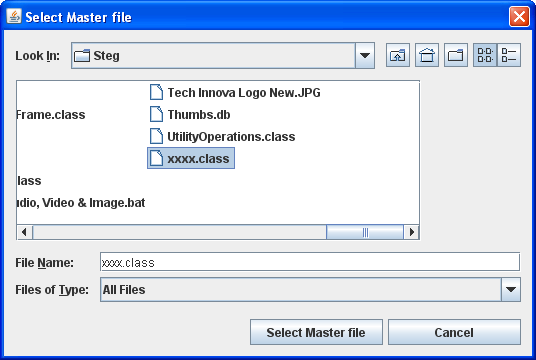
6 (iii)



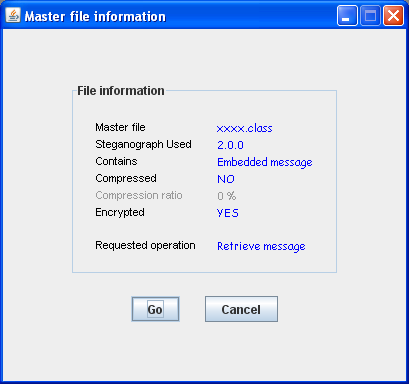
7. If the user clicks the **“Retrive Message”.**



8. After selecting the retrive message the below window will be opened. Here we have to select the Master file which we have given as output file in the **screen 5(i).**

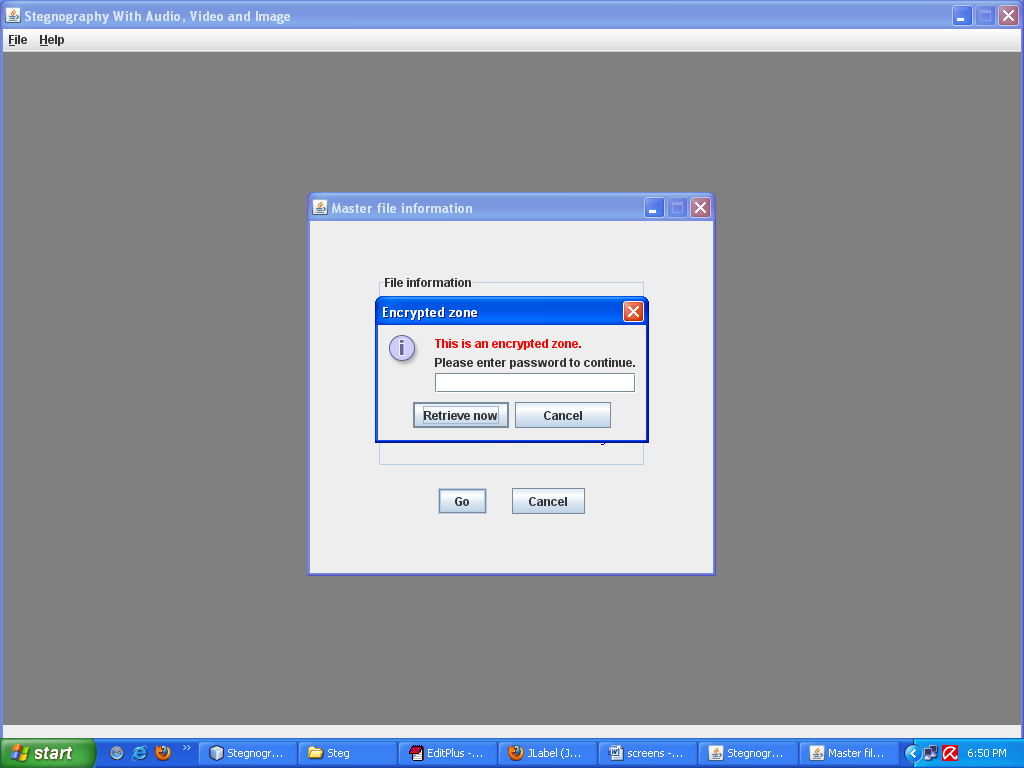


9. After selecting the master file the below window will be opened.



10. If the user clicks on the **“GO”** button in the above screen then the below screen will be displayed & here the user has to enter the password given while embed the message .

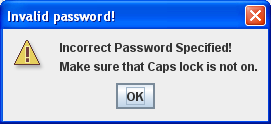
(i)



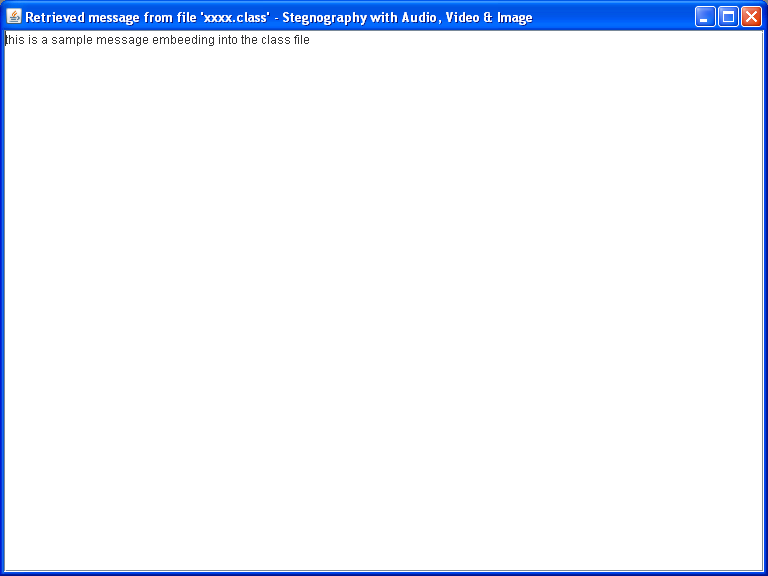
(ii)



(iii) If the user enters wrong password then the below alert will be displayed.

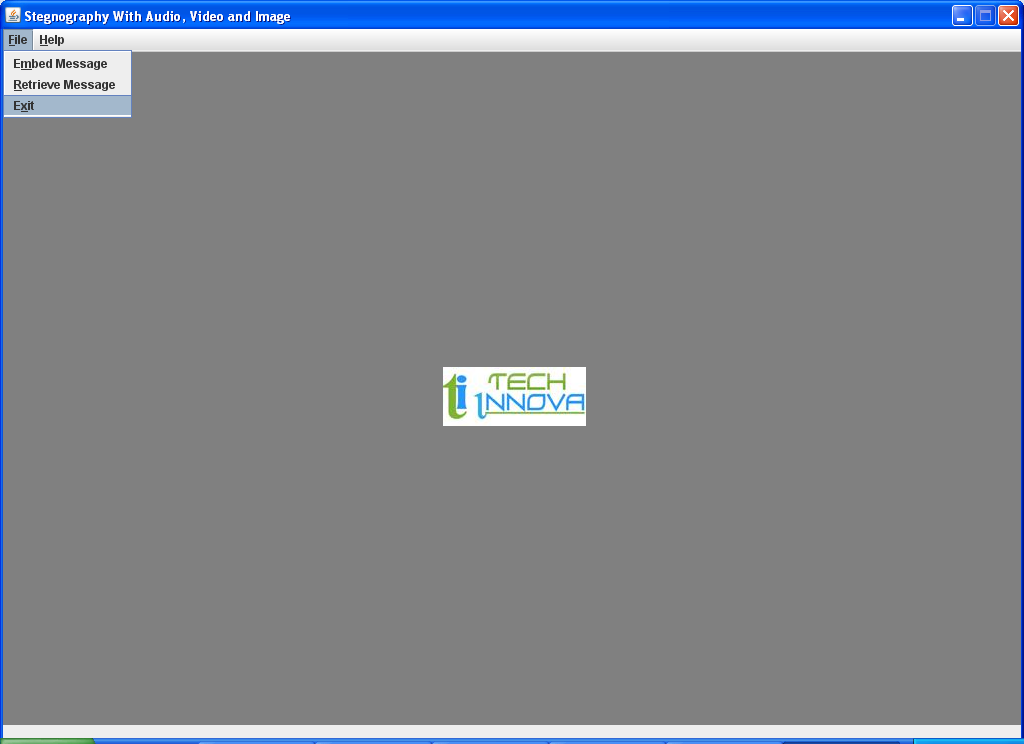


11. If the user enters the correct password then the retrived message will be displayed.

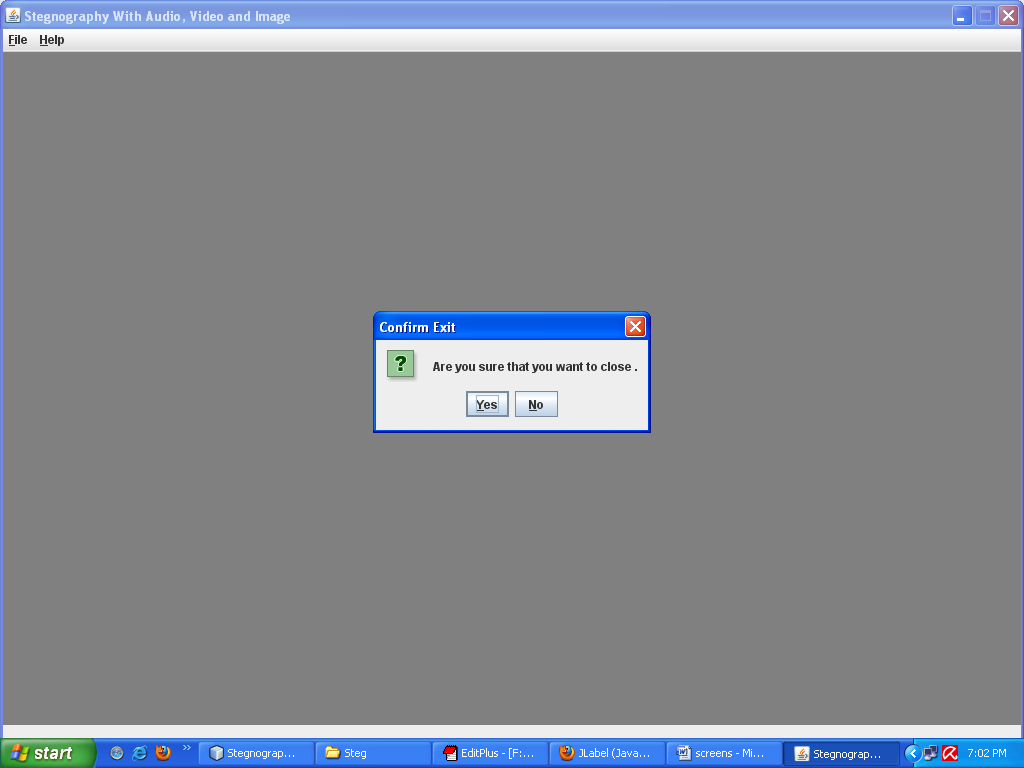


12(i)(ii). If the user wants to **“EXIT”** the application then he has to select the exit option & he has to select the **“YES”** option in the alert box.

(i)



(ii)



**CHAPTER --- 9 CONCLUSIONS**.

Steganography with Audio, Video & Image is a desktop application. The purpose of this application is to provide the security for the confidential information. This application doest allow the hackers to view the data, can view only Audio, Video & Image file when it is being passed over the internet. Then at the recipient side the original information i.e., plain text will be extracted from the Audio, Video & Image by performing decryption operations. This Project has been developed successfully.

I learned java 2 standard edition and swings which are very use full to develop this application.

**APPENDIX**

**ABBREVATIONS:**

**UML : Unified Modeling Language**

# DFD: Data Flow Diagrams

**FAQ’S**

**What is java GUI?**

It is an interface which gives the look and feel of a project.

**BIBILIOGRAPHY**

**Advanced Java Programming - Dietel and Dietel**

**Mastering JAVA 2 - John Zukowski**

**Java Server Programming - Apress**

**Software Engineering - Roger S Pressman**

**Análysis & Design of InformationSystems – Senn**

Websites

[www.eci.gov.in](http://www.eci.gov.in)

[www.google.com](http://www.google.com)

[www.apeci.com](http://www.apeci.com)

[www.askjeeves.com](http://www.askjeeves.com)