

**PROJECT PROPOSAL : Group 9**

**Project Name** : Steganography

**Group Members** :

* Haris Younas

**(UW-BS-AI-22-004)**

* Syed Abdul Rehman Nasir

**(UW-BS-AI-22-003)**

* Hamza Sajjad

**(UW-BS-AI-22-042)**

**Introduction**

Steganography is the practice of concealing information within other seemingly innocent or unrelated data, such as images, audio files, text documents, or even network protocols. The hidden information is typically embedded in such a way that it is difficult for unintended recipients or adversaries to detect its presence.

Steganography techniques aim to hide the existence of the hidden data, making it appear as normal and unaltered as possible. The primary objective is to prevent the unintended user from suspecting or even realizing that hidden information is present within the carrier file or communication.

Steganography should not be confused with cryptography. While both practices involve protecting information, cryptography focuses on encrypting data to render it unintelligible to unauthorized users. Steganography, on the other hand, focuses on hiding the existence of the data, making it a complementary technique to encryption for enhanced security.

It's important to note that steganography alone does not provide strong security against dedicated adversaries or advanced analysis techniques. Thus, steganography is often used in conjunction with encryption and other security measures to provide a layered approach to protecting sensitive information.

Maintaining secrecy is very important in a large corporation and because of the intelligent of the hackers it becomes tedious. Already we have crypto graphy for transmitting secret information. Even though crypto graphy successfully transmitting secret information, it will give a suspicion to the hackers and it affects unintended users.

Our project, **DIGITAL STEGANO GRAPHY** overcomes this factor and it gives a solution for transmitting secret formation with out affecting unintended users. Stegano graphy uses multimedia data as a covering medium (Covering secret information). By using stegano graphy data (secret information) can hided with in data (multimedia data, here multimedia data is an image) and it can be sent anywhere to transfer the message easily without giving any suspicion to others.

**Background study:**

Steganography is an ancient practice that dates back to ancient Greece, where secret messages were hidden on wax tablets covered in beeswax. The term "steganography" itself comes from the Greek words "steganos," meaning covered or concealed, and "graphein," meaning writing.

Throughout history, various methods of steganography have been used to hide information. One early example is invisible ink, where hidden messages are written using substances that only become visible when exposed to heat, chemicals, or special lighting. Another technique involved hiding messages within the quills of feathers or inside hollow objects.

With the advent of digital technology, steganography has evolved to exploit the properties of digital files such as images, audio, and video. The first digital steganographic technique was introduced in 1985 by Dr. Andreas Westfeld and Andreas Pfitzmann, who developed a method to embed hidden data within the least significant bits of a digital image.

There are various techniques employed in steganography to achieve this:

1. Least Significant Bit (LSB) Insertion: This technique involves embedding the hidden data within the least significant bits of the cover file. Since altering the least significant bits does not significantly affect the cover file's appearance or quality, unintended users are unlikely to detect any changes.
2. Spread Spectrum: This technique distributes the hidden data across the entire cover file, typically by modifying the pixel values of images or the audio samples of audio files. The changes made are often imperceptible to human senses but can be extracted by those who know how to interpret them.
3. Text Steganography: In this method, hidden data is concealed within the text itself. Techniques such as Word-Level Techniques , or using Text formatting techniques, such as modifying the font, color, or size of specific characters or words, can be used to encode the hidden data. These methods can be employed to hide information within a written document.

The field of steganalysis has also emerged, focusing on detecting the presence of steganography and attempting to extract the hidden information. Steganalysis techniques involve statistical analysis(examining the distribution of pixel values, frequencies, correlations), pattern recognition algorithms to identify patterns or anomalies in digital files that indicate the presence of hidden data.

Steganography has both legitimate and malicious applications. It can be used for secure communication or as an additional layer of security in combination with encryption. However, it can also be exploited by individuals involved in illegal activities, such as hiding malware or concealing sensitive information for malicious purposes.

Overall, the study of steganography encompasses a wide range of techniques, algorithms, and analysis methods, aiming to either conceal information within digital files or detect the presence of hidden data

**Tools and Technologies:**

Some of the tools and technologies that can be used in a Java-based stygnography project:

1. **Java Programming Language:** Java is a popular object-oriented programming language that is widely used for developing secure applications. Java offers features such as platform independence, security, and robustness, making it an ideal choice for developing stygnography projects.
2. **IDE:** IntelliJ IDEA is an integrated development environment (IDE) used for developing software in Java.
3. **Graphical User Interface (GUI):** The project utilizes the Java Swing framework for creating the graphical user interface (GUI) components and managing the application's windows, labels, panels, progress bar, and other visual elements.
4. **Resource Loading:** The project uses resource loading mechanisms provided by Java, such as **getClass().getResource()**, to load external resources like images.
5. **Multithreading:** The project includes a separate thread, SplashThread, to update the progress bar and load the main menu frame concurrently while displaying the splash screen. Multithreading is used to ensure responsiveness and avoid freezing the GUI during time-consuming tasks.
6. **Image processing:** Steganography involves hiding data within an image while maintaining the visual integrity of the image. This typically requires manipulating the pixel values of the image to embed or extract the hidden data. So we will use some image processing techniques used in steganography.

**Key Features:**

Some key features/functionality requirements that could be included in our project:

1. **Open Image:** Allows the user to open a steganographed image file for decoding.
2. **Decode Message:** Extracts the hidden message from the steganographed image.
3. **Reset Interface:** Resets the user interface by clearing the displayed message and removing the loaded image.
4. **GUI Components:** The project defines a GUI frame using Java's Swing library. It includes buttons (jButton1 and jButton2) for encoding and decoding operations, labels (jLabel1 and jLabel2) for displaying text, and panels (jPanel1, jPanel2, and jPanel3) for organizing the components.
5. **Event handling:** It is a fundamental aspect of graphical user interface (GUI) programming. It allows you to respond to user interactions or system events and perform actions accordingly. In our project , event handling is used to handle the progress of the splash screen and transition to the menu frame when the progress is complete.
6. **Window Settings:** The code sets various properties of the GUI frame, such as the title, dimensions, and default close operation.

**Algorithms:**

1. **LSB (Least Significant Bit) Steganography:** This algorithm works by replacing the least significant bits of pixel values in the image with the bits of the hidden message. The hidden message is encoded by modifying the color values of the pixels without significantly altering the visual appearance of the image.
2. **Image I/O:** The project uses Java's built-in javax.imageio package to read and manipulate image files. This package provides the necessary functionality to load and save images in various formats, such as PNG and BMP.

**Class Diagram:**

|  |
| --- |
| **Stegno module** |
| **+**secret text  **+**cover image |
| **+**select secret text()  **+**secret cover image() |

|  |
| --- |
| **Encryption** |
| **+**secret text  **+**cover image  **+**Stegno image |
| **+**take text()  **+**take cover image()  **+**encrypt()  **+**create encrypted image() |

|  |
| --- |
| **Unstego module** |
| **+**stego image |
| **+**take stego image() |

|  |
| --- |
| **Decryption** |
| **+**decrypt  **+**stego image |
| **+**decrypt()  **+**display secret text() |

**ARCHITECTURE OF THE SYSTEM:**

**Sender**

(Hiding Data)

( Secret Information )

Stegano Medium

**Receiver**

(Extracting Secret Information)

Stegano Medium

Secret Information

Stegano Medium

**Flow Chart:**

Select Operation

Start

End

Encode

Decode

Message to be encode

Load Image/Stegano Media

Open the image

Decode

Embed message into image

Extract Secret Information

Encoding Complete

End

Save the embedded image

OR Stegano Media

Start Stegano Application

**Gantt Chart:**

|  |  |  |
| --- | --- | --- |
| Start Date | End Date | Work |
| 27-May | 28-May | Background Study |
| 29-May | 30-May | Choose a algorithm |
| 31-May | 1-June | Implement encryption process |
| 2-June | 3-June | Implement decryption process |
| 4-June | 5-June | Test encryption/decryption functionality |
| 6-June | 7-Jun | Least Significant Bit Alogrithm |
| 8-Jun | 9-Jun | Embed Encrypted Message in Image |
| 10-Jun | 10-Jun | Graphical User Interface |
| 11-Jun | 12-Jun | Document project and prepare for submission |

**Distribution:**

**Encryption and Decryption:**

* Muhammad Haris
* Muhammad Hamza Sajjad
* Syed Abdul Rehman Nasir

**Graphical User Interface:**

* Muhammad Hamza Sajjad
* Syed Abdul Rehman Nasir
* Muhammad Haris

**Least Significant Bit Alogrithm:**

* Muhammad Haris

**Document project and prepare for submission:**

* Muhammad Hamza Sajjad
* Syed Abdul Rehman Nasir