# PROJECT AND TEAM INFORMATION

## Project Title

|  |
| --- |
| Image-based and Audio-based Steganography for Secure Data Transmission |

## Student / Team Information

|  |  |
| --- | --- |
| Team Name: | Team SilentCipher |
| **Team member 1 (Team Lead)**  Sharma Aayushi  230121926  [sharmaaayushi953@gmail.com](mailto:sharmaaayushi953@gmail.com) |  |
| **Team member 2**  Akansha Rawat  230122459  [akansharaw0101@gmail.com](mailto:akansharaw0101@gmail.com) |  |

|  |  |
| --- | --- |
| **Team member 3**  Tanuja  230124746  [tanujachuphal7@gmail.com](mailto:tanujachuphal7@gmail.com) |  |
| **Team member 4**  Rawat Singh Gaurav  230121457  [singhrawatg121@gmail.com](mailto:singhrawatg121@gmail.com) |  |

# PROPOSAL DESCRIPTION

## Motivation

|  |
| --- |
| In today’s digital world, securing sensitive information during transmission is crucial. Traditional encryption methods, though effective, can be detected by adversaries, making it possible to intercept and analyse encrypted data. Steganography, the art of concealing data within other files, provides an additional layer of security by hiding information in plain sight. However, current solutions mainly focus on text encryption or basic image steganography, leaving room for more advanced methods that could work with different media types like images and audio.  The problem we aim to solve is the challenge of securely transmitting data by embedding it within image and audio files without detection. Existing methods are often vulnerable to attacks, such as steganalysis, where adversaries analyse media files for hidden messages. Additionally, embedding large files or sensitive information in media files may distort the file’s appearance or sound, making the presence of hidden data detectable.  **Important:**  The importance of this problem lies in the need for secure and undetectable data transmission methods. In scenarios like confidential government communications, intellectual property protection, or covert military operations, it is essential to send sensitive data securely without raising suspicion. By enhancing image-based and audio-based steganography techniques, we can offer an additional layer of protection beyond traditional encryption methods. This project will enable more sophisticated and resilient data transmission solutions that can be applied to various fields, including cybersecurity, digital forensics, and secure communications, without compromising the integrity or quality of the carrier media. |

## State of the Art / Current solution

|  |
| --- |
| Today, steganography is commonly implemented using methods such as **Least Significant Bit (LSB)** manipulation for both images and audio files. In **image-based steganography**, the secret data is embedded by altering the least significant bits of pixel values, typically in lossless formats like PNG or BMP to avoid data loss. The modified image appears visually unchanged, making it difficult for an observer to detect the presence of hidden data. This technique is often used to conceal short text messages or small files.  For **audio-based steganography**, the approach is similar, where the least significant bits of audio samples (in WAV or PCM formats) are altered to encode hidden data. This can be applied to both short messages and files. Techniques like **Phase Coding** and **Frequency Domain Embedding** can also be used for more sophisticated hiding of data within the audio’s frequency spectrum. |

## 

## Project Goals and Milestones

|  |
| --- |
| The main goal of this project is to develop a robust steganography system capable of securely hiding data within both **image** and **audio** files for covert communication. The system should provide a secure, undetectable method for embedding and extracting messages or files in digital media, while ensuring minimal distortion to the carrier media (images and audio). By incorporating encryption, error correction, and advanced embedding techniques, the project will aim to enhance both the security and capacity of steganographic data transmission.  **Initial Milestones:**   1. **Research & Analysis** :    * Investigate existing image and audio-based steganography techniques.    * Study popular algorithms like LSB and advanced methods like frequency domain embedding.    * Review current steganalysis techniques to understand potential attack vectors. 2. **Image-based Steganography Prototype** :    * Develop a basic image encoding and decoding system using LSB for hiding and extracting data.    * Test encoding small text messages and files in lossless image formats (e.g., PNG).    * Ensure the system preserves image quality and detectability is minimized.   **Intermediate Milestones:**   1. **Audio-based Steganography Prototype** :    * Implement LSB or frequency domain techniques for encoding data in audio files (e.g., WAV).    * Test the encoding and decoding process, ensuring minimal audio distortion.    * Handle both small messages and file embedding. 2. **Encryption & Compression Integration** :    * Introduce encryption methods to secure the hidden data before embedding.    * Implement compression to reduce the size of large files before encoding.   **Final Milestones:**   1. **Error Detection and Resistance to Steganalysis** :    * Add error detection techniques like checksums or hashes to verify data integrity.    * Enhance the system’s resistance to steganalysis by integrating multi-layer embedding and data masking. 2. **Testing & Optimization** :    * Perform extensive testing on both image and audio files for capacity, quality, and resistance to detection.    * Optimize for speed, efficiency, and scalability. |

## Project Approach

|  |
| --- |
| To design a secure and efficient solution for image-based and audio-based steganography, the approach will focus on modular development, with separate components for encoding, decoding, encryption, and error detection. The overall system will ensure data security, minimal distortion of the carrier media, and resistance to steganalysis.   1. **Platform Selection:**    * **Programming Language:** Python will be the primary programming language due to its extensive libraries for image and audio processing, such as **Pillow** (for image manipulation), **Wave** (for handling audio), and **NumPy** (for efficient numerical operations). Python also offers robust libraries for encryption like **PyCryptodome**.    * **IDE/Tools:** Jupyter Notebook or PyCharm will be used for the development and testing of the solution, as they provide efficient debugging and support for Python libraries. 2. **Design of Image-based Steganography:**    * **Data Encoding:** Use the **Least Significant Bit (LSB)** method to hide data within the pixel values of an image. The Pillow library will be used to load, manipulate, and save images.    * **Data Decoding:** Extract the hidden message by reading the least significant bits of the pixels and converting them back into binary data.    * **Enhancements:** Use encryption techniques (e.g., AES) to secure the data before embedding and error correction codes (e.g., CRC32) to ensure data integrity. 3. **Design of Audio-based Steganography:**    * **Data Encoding:** Implement the LSB method or alternative techniques like **Phase Coding** using the **wave** module to handle WAV files.    * **Data Decoding:** Extract the hidden message from the audio file by reading the LSBs of the samples and reconstructing the original message.    * **Encryption and Compression:** Integrate encryption (e.g., RSA or AES) and compression (e.g., using **zlib**) before embedding large files. 4. **Security Considerations:**    * Implement methods to prevent common attacks like **steganalysis** by applying techniques like **multi-layer embedding** and **frequency domain manipulation**. |

## 

## System Architecture (High Level Diagram)

|  |
| --- |
| +-----------------------------------+  | User Interface |  | (Data Input and Output) |  +-----------------------------------+  |  v  +-----------------------------------+  | Data Processing |  | (Data Validation and Preparation) |  +-----------------------------------+  |  v  +-----------------------------------+ +----------------------------------+  | Image-based Steganography | | Audio-based Steganography |  | (Encoding and Decoding Module) | | (Encoding and Decoding Module) |  +-----------------------------------+ +----------------------------------+  | |  v v  +-----------------------------------+ +----------------------------------+  | LSB Encoding | | LSB/Phase Coding Encoding |  | (Embedding Data in Images) | | (Embedding Data in Audio Files)|  +-----------------------------------+ +----------------------------------+  | |  v v  +-----------------------------------+ +----------------------------------+  | Encryption Module | | Encryption Module |  | (Secure Data Handling) | | (Secure Data Handling) |  +-----------------------------------+ +----------------------------------+  | |  v v  +-----------------------------------+ +----------------------------------+  | Error Detection Module | | Error Detection Module |  | (Ensuring Data Integrity) | | (Ensuring Data Integrity) |  +-----------------------------------+ +----------------------------------+  | |  v v  +-----------------------------------+ +----------------------------------+  | Output File Generation | | Output File Generation |  | (Generating Stego Images) | | (Generating Stego Audio) |  +-----------------------------------+ +----------------------------------+  |  v  +-----------------------------------+  | Security Layer |  | (Preventing Steganalysis) |  +-----------------------------------+ |

## Project Outcome / Deliverables

|  |
| --- |
| The outcomes and deliverables of this project include the following:   1. **Image-based Steganography System**: A fully functional module that allows users to securely hide and retrieve messages or files within image files. This system will use techniques like Least Significant Bit (LSB) manipulation, with options for encryption and error detection to enhance security. 2. **Audio-based Steganography System**: A module for encoding and decoding data within audio files (WAV format). This will also utilize LSB manipulation or alternative techniques like Phase Coding to embed data in audio while preserving the quality of the sound. 3. **Encryption and Compression Integration**: Both image and audio modules will incorporate encryption methods (e.g., AES or RSA) to secure the data before embedding. Compression techniques (e.g., ZIP or GZIP) will be used to reduce the size of larger files to be hidden. 4. **Error Detection**: Implementation of error-checking mechanisms such as checksums or CRC to verify the integrity of the hidden data and ensure reliable transmission. 5. **Steganalysis Resistance**: Advanced techniques will be incorporated to make the system resistant to common steganalysis methods, ensuring that the hidden data is not easily detectable. 6. **Documentation**: Complete technical documentation outlining the system architecture, design decisions, and usage instructions. |

# Assumptions

|  |
| --- |
| The assumptions made for solving this problem include:   1. **Lossless Media**: The carrier media (image and audio files) will be in lossless formats (e.g., PNG, BMP for images, WAV for audio) to avoid any degradation that could reveal hidden data. 2. **Data Size Limitations**: The size of the secret data will be small enough to fit within the carrier media without causing noticeable distortion, especially in audio or high-resolution images. 3. **User Access to Media Files**: Users will have access to both the carrier media and the tools required for encoding and decoding the data. 4. **No Compression on Carrier Files**: No lossy compression (e.g., JPEG or MP3) will be applied, which could interfere with the steganography process. |

## References

.

|  |
| --- |
| 1. **Python Libraries**:  * **Pillow (PIL)** for image manipulation: <https://pillow.readthedocs.io/> * **NumPy** for efficient numerical operations: <https://numpy.org/> * **PyCryptodome** for encryption algorithms (AES, RSA): <https://www.pycryptodome.org/> * **Wave Module** (built-in Python) for audio file manipulation: <https://docs.python.org/3/library/wave.html>  1. **Steganography Techniques**:  * “Digital Image Steganography” by S. B. S. Rajput, et al., for methods like LSB and frequency-domain embedding. * "Audio Steganography: A Survey" by M. C. R. N. P. Gupta, et al., which provides an overview of techniques in audio-based steganography. * **Introduction to Steganography** by Thomas H. Hearn: https://www.scholarpedia.org/article/Steganography  1. **Encryption & Security**:  * **Cryptography and Network Security** by William Stallings, for understanding the basics of encryption (AES, RSA) and their implementation. * **Advanced Cryptography**: <https://crypto.stackexchange.com/>  1. **Error Detection**:  * **Python Checksums** using libraries like **zlib** for error detection: <https://docs.python.org/3/library/zlib.html> * **Error Detection in Data Transmission**: <https://en.wikipedia.org/wiki/Error_detection_and_error_correction>  1. **Steganalysis**:  * **Steganalysis Techniques**: <https://www.springer.com/gp/book/9783319787747> for advanced methods of detecting steganographic data. |