A

PROJECT REPORT
on

"StegX"

for

Mini Project 2-A (REV- 2019 'C' Scheme) of Third Year,
(TE Sem-V)

in

Computer Science and Engineering (IOT and Cyber Security including Blockchain Technology)

by

KETKI DIGHE – 122AX027
MAYUR MOHITE – 122AX030
PRAVANSHU MAJI – 122AX036

UNDER THE GUIDANCE OF

PROF. PRANITA PINGALE

UNIVERSITY OF MUMBAI

SIES Graduate School of Technology
Sri Chandrasekarendra Saraswati Vidyapuram Sector-V,
Nerul, Navi Mumbai,
Maharashtra 400706
Academic Year 2024-25

CERTIFICATE

This is to certify that the project entitled "**StegX"** is a bonafide work of

1. KETKI DIGHE 122AX027
2. MAYUR MOHITE 122AX030
3. PRAVANSHU MAJI 122AX036

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of **Mini Project 2-A (REV- 2019 'C' Scheme) of Third Year, (TE Sem-V)** in **Computer Science and Engineering (IOT and Cyber Security including Blockchain Technology) as** laid down by **University of Mumbai** during academic year **2024-25.**

Internal Guide **Head of Department** **Principal**

We have examined this report as per university requirements at SIES Graduate School of Technology, Nerul (E), Navi Mumbai on _____.

Name of External Examiner: _____
Signature with Date: _____

Name of Internal Examiner: _____
Signature with Date: _____

ACKNOWLEDGEMENT

We wish to express our deep sense of gratitude and thank to our Internal Guide, Prof. Pranita Pingale for her guidance, help and useful suggestions, which helped in completing our miniproject work in time. We are also extremely grateful to our Miniproject coordinator Prof. Smruthy C.S for her guidance provided whenever required. We also thank our Hod , Dr. Sulochana Madachane for her support in completing the project. We also thank to our Principal Dr. K Lakshmisudha for extending her support to carry out this project.

Also, we would like to thank the entire faculty of Computer Science and Engineering (IOT and Cyber Security including Blockchain Technology) department for their valuable ideas and timely assistance in this work, last but not the least, we would like to thank our teaching and non-teaching staff members of our college for their support, in facilitating timely completion of this project.

Mini Project Team

Ketki Dighe 122AX027

Mayur Mohite 122AX030

Pravanshu Maji 122AX036

ABSTRACT

This project presents a versatile steganography tool designed to securely embed and extract hidden information within images, audio, and text files. Built using PyQt5 for a user-friendly graphical interface, the tool offers an intuitive platform for encoding and decoding hidden data. Each media type is managed through separate modules, providing a seamless experience for users to perform steganographic operations while ensuring the perceptual quality of the original media remains intact. By hiding data in commonly used media formats, the tool helps enhance privacy and security in digital communication. The system leverages robust multimedia processing libraries such as OpenCV for image steganography and pydub for audio handling. These libraries enable efficient data embedding techniques while maintaining the media's original appearance and sound quality. The tool supports a range of common file formats, ensuring compatibility with a variety of input sources and offering flexibility in the type of data concealed, including text and images. The interface is designed to simplify the steganographic process for users of all technical backgrounds. The core aim of this project is to provide an accessible yet powerful solution for protecting sensitive information. By embedding data within media files that appear innocuous, users can ensure their private information is safeguarded from unauthorized access. This tool contributes to the growing field of steganography, offering an innovative approach to data security in a world where privacy protection is of increasing concern.

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

In today's highly connected and digital world, data security and privacy have become paramount concerns for individuals, businesses, and governments alike. With the increasing reliance on digital platforms for communication, commerce, and information exchange, vast amounts of sensitive information are transmitted across networks every day. This includes personal details, financial transactions, confidential business communications, and proprietary data. As the volume of digital data grows, so does the threat landscape, with cyber-attacks, data breaches, and unauthorized access becoming more frequent and sophisticated.

Traditional data protection methods such as encryption play a crucial role in safeguarding sensitive information by converting it into an unreadable format that can only be decoded with the proper key. However, encryption has a limitation: it inherently draws attention. Encrypted data, by its very nature, signals that its content is valuable or confidential, making it a potential target for adversaries. While encryption ensures that the content remains secure, it does not hide the fact that sensitive information exists. This is where steganography—the art and science of hiding information within seemingly innocuous media—provides a powerful complementary approach to data security. This project aims to develop a comprehensive, user-friendly steganography tool that supports three major forms of media: image, audio, and text. By leveraging widely used file formats such as PNG, BMP, WAV, MP3, and TXT, the tool ensures compatibility with a variety of media types while maintaining the integrity and usability of the original files. The goal is to create a platform where users can securely embed and extract hidden data with minimal effort, making steganography accessible to non-technical users.

Steganography conceals the presence of secret data by embedding it within common media files such as images, audio, and text. Unlike encryption, which transforms data into an unreadable format, steganography hides data in plain sight, making it invisible to casual observers and potential attackers. For example, a simple image may carry a hidden message without altering its visual appearance, or an audio file may contain concealed information without affecting its sound quality. The primary advantage of steganography lies in its ability to mask the existence of hidden information, thus adding an extra layer of security.

1.1) Motivation

The motivation for this project arises from the growing need to enhance digital privacy and security in the face of increasingly sophisticated cyber threats. As society becomes more digitized, with personal, financial, and business information constantly being shared online, the risks of unauthorized access and data exploitation have multiplied. Sensitive data can be exposed through hacking, surveillance, or interception during transmission. While encryption technologies offer robust protection, they also serve as a signal that the encrypted data is important, which could potentially attract unwanted attention from malicious actors.

The primary motivation behind this project is to provide an alternative, subtler method of securing information. Unlike encryption, which focuses on transforming data into a coded format, steganography hides the very existence of data. In today's digital world, this added layer of secrecy is particularly valuable. By embedding sensitive information into benign-looking media files, steganography offers a way to keep communications and data safe while minimizing the risk of detection.

Moreover, the need for multi-media steganography tools has become increasingly evident as users interact with various types of media on a daily basis. While existing steganography tools typically specialize in one type of media, such as image steganography or audio steganography, modern users often work with different formats for different purposes—whether it's sharing images on social media, sending audio clips over messaging apps, or using text-based communication platforms. Thus, there is a clear need for a unified steganography tool that can handle a variety of media formats, providing users with flexibility and ease of use. A second significant motivation stems from the lack of accessible steganography tools for non-technical users. Many existing steganography tools are complex and require a deep understanding of how steganographic algorithms work, making them impractical for average users who simply want to ensure the privacy of their communications. This project seeks to bridge this gap by developing a **user-friendly interface** that abstracts the technical complexities of steganography, allowing anyone to embed or extract hidden data without specialized knowledge.

1.2) Problem Statement and Objectives

Problem Statement

The primary challenge this project addresses is the lack of versatile, user-friendly steganography tools that support multiple media types—specifically images, audio, and text—while maintaining data security and media integrity. Most existing tools are either too complex for everyday users or are limited to a single media type, making them impractical for broader use cases. Moreover, tools that do exist often result in noticeable degradation of the original media, which can raise suspicion and reduce the effectiveness of the hidden data. This project also addresses the need for an accessible interface, as many steganography applications require extensive technical knowledge, making them inaccessible to the general public. The problem lies in developing a system that not only handles diverse media types but also provides an easy-to-navigate interface for both embedding and extracting hidden data, all while preserving the quality of the original media.

Objectives

The main objective of this project is to design and implement a steganography tool that allows users to securely embed and extract hidden information within images, audio, and text files. The tool should achieve the following specific objectives:

1. Support Multiple Media Types:
    - Provide steganographic capabilities for images, audio, and text. This will involve embedding data in media formats like PNG and BMP for images, WAV and MP3 for audio, and TXT for text files.

2. Ensure Media Integrity:
    - Use techniques like Least Significant Bit (LSB) manipulation to ensure that the hidden data is embedded in a way that does not significantly degrade the quality or appearance of the original media. The goal is to make the embedded data undetectable to the naked eye or casual listener.

3. Provide a User-Friendly Interface:
    - Develop a graphical user interface (GUI) using PyQt5 that simplifies the steganographic process. Users should be able to upload files, input hidden data, and select whether to embed or extract data through an intuitive interface that does not require technical expertise.

4. Ensure Security and Data Integrity:
    - Implement robust error checking to ensure that the hidden data can be successfully extracted without corruption. Additionally, the tool should handle different file sizes and automatically calculate whether a given media file can accommodate the hidden data.

5. Support Common File Formats:
    - The tool should handle popular media file formats to ensure its usability across different platforms. For example, images will use lossless formats like PNG, audio will use uncompressed formats like WAV, and text will support plain text (TXT) files.

6. Allow Scalability:
    - Design the system in a modular way so that future extensions can be made to include other forms of media or improve existing steganographic algorithms.

7. Ensure Cross-Platform Compatibility
    - In an effort to reach a wide audience, the tool will aim to be **cross-platform**, ensuring that it can be used on different operating systems such as Windows, macOS, and Linux. This objective is crucial for making the tool accessible to users in various environments and industries.

By achieving these objectives, the project will deliver a powerful and flexible steganography tool that can meet the needs of individuals and organizations looking to enhance privacy in their digital communications. Whether for personal use, professional applications, or secure document sharing, the tool will provide a discreet and reliable method of concealing sensitive data across multiple media formats.

2. LITERATURE REVIEW

2.1) Survey of Existing System

Author	Project	Year	Content
Enas Wahab Abood, Abdulhssein M. Abdullah, Mustafa A. Al Sibahee, Zaid Ameen Abduljabbar, Vincent Omollo Nyangaresi, Saad Ahmad Ali Kalafy, Mudhafar Jalil Jassim Ghrabta	Audio Steganography with Enhanced LSB Method for Securing Encrypted Text with Bit Cycling	2022	This paper presents a hybrid system combining cryptography and audio steganography for secure data transmission. It uses a bit cycling encryption technique to generate cipher text, which is hidden in audio files using an enhanced Least Significant Bit (LSB) method with random bit distribution. The system is evaluated using PSNR, MSE, and SSIM metrics, demonstrating its effectiveness with high data security, minimal distortion, and low time consumption. The method is suitable for use in lightweight devices due to its efficiency(3279-8289-1-PB).
Nandhini Subramanian, Omar Elharrouss, Somaya Al-Maadeed, Ahmed Bouridane	Image Steganography: A Review of Recent Advances	2021	This review highlights modern approaches to image steganography using deep learning, especially traditional methods like LSB, CNN-based techniques, and GAN-based methods. The paper classifies methods, discusses datasets, performance metrics, and addresses challenges in steganography, along with countermeasures such as steganalysis.

Author	Project	Year	Content
Manisha Verma, Hardeep Singh Saini	Analysis of Various Techniques for Audio Steganography in Data Security	2019	This paper reviews audio steganography techniques to enhance data security. It discusses hiding data within audio files, using methods like LSB coding, parity coding, phase coding, spread spectrum, and echo hiding. Each method's strengths and weaknesses are analyzed, focusing on how these techniques improve the security and privacy of transmitted data. The study emphasizes the need for more secure methods to prevent unauthorized access to hidden information (1-IJSRNSC-0489-22).
Parmar Ajit Kumar Maganbhai, Prof. Krishna Chouhan	A Study and Literature Review on Image Steganography	2015	This review discusses various image steganography techniques, including classical methods like LSB, wavelet transforms, and Huffman encoding. It explores the applications of steganography in secure communication, copyright protection, and introduces methods to detect hidden data through steganalysis (Stego).

2.2) Research Gap

Despite the wide range of techniques available for steganography, there remain several research gaps and challenges that need to be addressed, particularly when it comes to balancing security, data capacity, and computational efficiency. Based on the survey of existing systems, the following gaps have been identified:

1. Limited Cross-Media Solutions

   Many existing steganographic systems focus on a single type of media, such as images or audio, without providing a unified solution that can handle multiple forms of media (image, audio, and text). There is a need for a comprehensive tool that can seamlessly integrate these different media types, offering a more versatile solution for secure communication.
2. Detection Vulnerabilities

   Methods like LSB, while simple and effective for small amounts of hidden data, are vulnerable to detection by statistical and analytical attacks. Even more advanced techniques, such as phase coding or frequency-domain methods, can suffer from perceptible degradation of the media quality, especially when large amounts of data are embedded. There is a need for more robust algorithms that offer higher security without compromising media quality.
3. Hybrid Systems Complexity

   While combining cryptography with steganography offers enhanced security, it often introduces significant computational overhead. Many hybrid systems, such as those involving complex encryption algorithms like AES or RSA, increase processing time and resource consumption, making them less suitable for real-time or resource-constrained environments(3279-8289-1-PB). Simplifying these hybrid approaches while maintaining security remains a challenge.
4. Scalability and Usability

   Current tools often lack user-friendly interfaces and scalability. Most steganographic tools are developed for specific use cases and are not designed with the flexibility to handle large-scale applications or diverse user needs. An ideal solution would be scalable, easy to use, and adaptable to different file types and sizes.

3.METHODLOGY

3.1) Design and Architecture

1) Modular Design:

The tool follows a modular design, where each form of media (image, audio, and text) is treated as a separate module. This modularity ensures flexibility, maintainability, and scalability. Each module uses specialized libraries tailored to handle the steganographic processes for that specific media type, such as OpenCV and NumPy for image processing, and pydub for audio. This separation also allows for easy upgrades or the addition of new functionalities in the future.

2) Core Components:

Graphical User Interface (GUI): The GUI is the primary user interaction layer, designed using PyQt5, which is a Python binding of the popular Qt framework. The GUI provides an easy-to-use, intuitive platform that abstracts the complexities of steganography for end-users. Users interact with the software through clearly defined tabs representing different media types, each containing controls for embedding or extracting data.
The main components of the GUI are:
Tabs: The GUI includes different tabs for each type of steganography: Image, Audio, and Text. Users can switch between these tabs to select the desired steganographic operation.
File Selection & Input Fields: Each tab has options for users to upload their cover files (image, audio, text), as well as an input field for the secret message or data to be hidden.
Process Control: The user can start the embedding or extraction process through dedicated buttons. Progress bars or status messages keep the user informed about long-running tasks.
Data Embedding & Extraction Engines: Each media module has its own data embedding and extraction engine, which implements the steganographic algorithms. These engines ensure that hidden data is securely and efficiently embedded into the media files, while keeping the cover media as unchanged as possible.

3) Media-Specific Steganography Engines:

Image Steganography Engine: The image steganography engine uses the Least Significant Bit (LSB) manipulation technique. In an image, each pixel is represented by 8 bits per color channel (e.g., RGB). The LSB technique works by modifying the least significant bit of these pixel values, which does not significantly affect the visual quality of the image but allows data to be embedded.

Key features:

Supports lossless image formats like PNG and BMP to avoid compression artifacts that could destroy the hidden data.

Uses OpenCV and NumPy libraries for pixel manipulation and image handling.

Embeds data into either a single color channel (e.g., blue) or across all channels (RGB), depending on the required data capacity.

Audio Steganography Engine: The audio steganography engine also uses the Least Significant Bit (LSB) technique but applies it to the digital samples of the audio file. Audio data is stored as a series of amplitude values, and the LSB of these values is modified to hide secret data.

Key features:

Supports popular audio formats like WAV and MP3, with WAV being preferred due to its uncompressed nature.

Uses pydub and wave libraries for audio file processing and manipulation.

Embeds data by modifying the LSBs of audio samples, ensuring minimal impact on the perceived quality of the audio file.

Text Steganography Engine: Text steganography is more challenging due to the limited amount of redundancy available in text files. The tool uses formatting-based techniques, where hidden information is encoded through subtle changes in the formatting or whitespace of the text. This method ensures that the appearance of the text file remains unchanged to the human eye, but the hidden data can be extracted programmatically.

Key features:

Supports plain text files (TXT).

Uses methods such as modifying the spacing between words or lines, or embedding invisible characters (like zero-width spaces).

The data embedding is designed to be undetectable while preserving the original formatting of the text.

4) File Handling and Format Support:

The steganography tool is designed to handle a variety of file formats, ensuring flexibility and compatibility with the user's needs. Each media type undergoes format checks and conversions where necessary to ensure that the hidden data remains intact and retrievable after saving and reloading the files. The system handles each media type as follows:

- **Images:** The tool supports BMP and PNG formats for embedding data. These lossless image formats preserve the quality of the image and the integrity of the hidden data, preventing it from being corrupted by lossy compression algorithms found in formats like JPEG.

- **Audio:** The preferred format for audio steganography is WAV, an uncompressed format that maintains high audio quality and ensures that the hidden data remains intact. However, the tool also supports compressed formats like MP3, though the steganography process is carefully adjusted to minimize data loss due to compression.

- **Text:** The tool supports plain text (TXT) files, which are widely used and simple to manipulate for formatting-based steganography. The system checks the file's format before embedding data and ensures that the hidden data is preserved even after multiple rounds of saving and reloading.

3.2) User Workflow

Step 1: Media Selection:

The user selects the media type (image, audio, or text) and uploads the desired cover file. The GUI checks the file format to ensure it is supported.

Step 2: Input Data:

The user inputs the secret data they wish to hide within the selected media file. The interface ensures that the data fits within the capacity of the chosen media.

Step 3: Embedding Process:

The steganographic engine for the selected media processes the file and embeds the secret data. For longer processes (especially with large audio files), a progress bar or message notifies the user about the ongoing operation.

Step 4: Save the Steganographed File:

Once the embedding process is complete, the user can save the steganographed file in the same or a new format, ensuring that the hidden data is securely stored within the file.

Step 5: Extraction Process:

The user can upload a previously steganographed file and select the "extract" function to retrieve the hidden data. The tool then applies the reverse process to extract the embedded information, displaying it in a readable form.

4. EXPERIMENT AND RESULTS

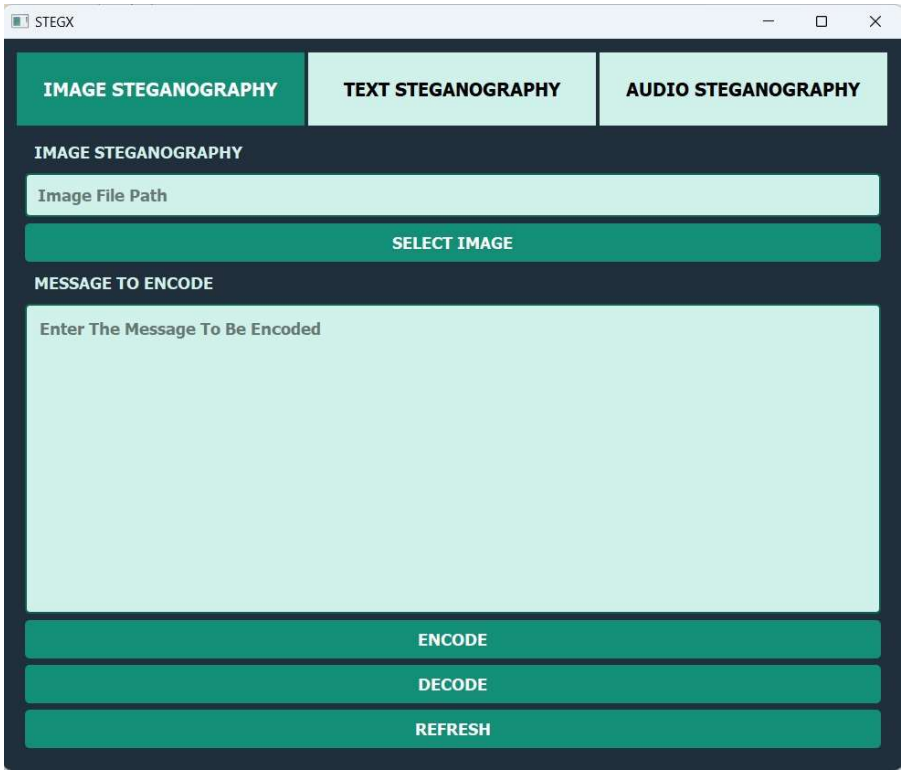

Fig 4.1 : Image Steganography GUI

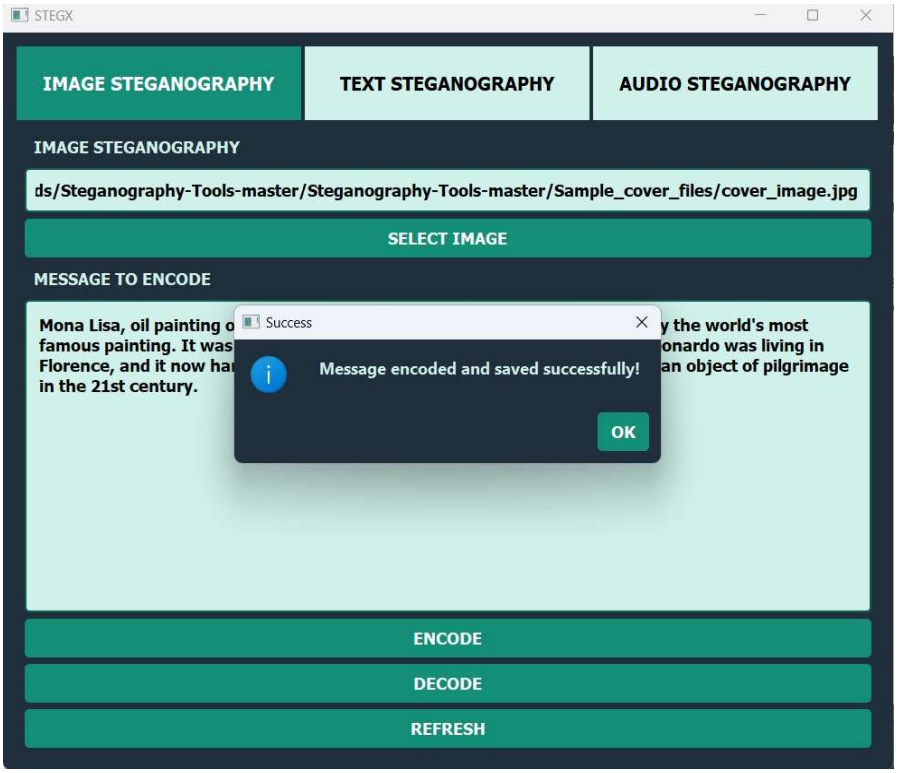

Fig 4.2 : Message Encoding in Image Steganography

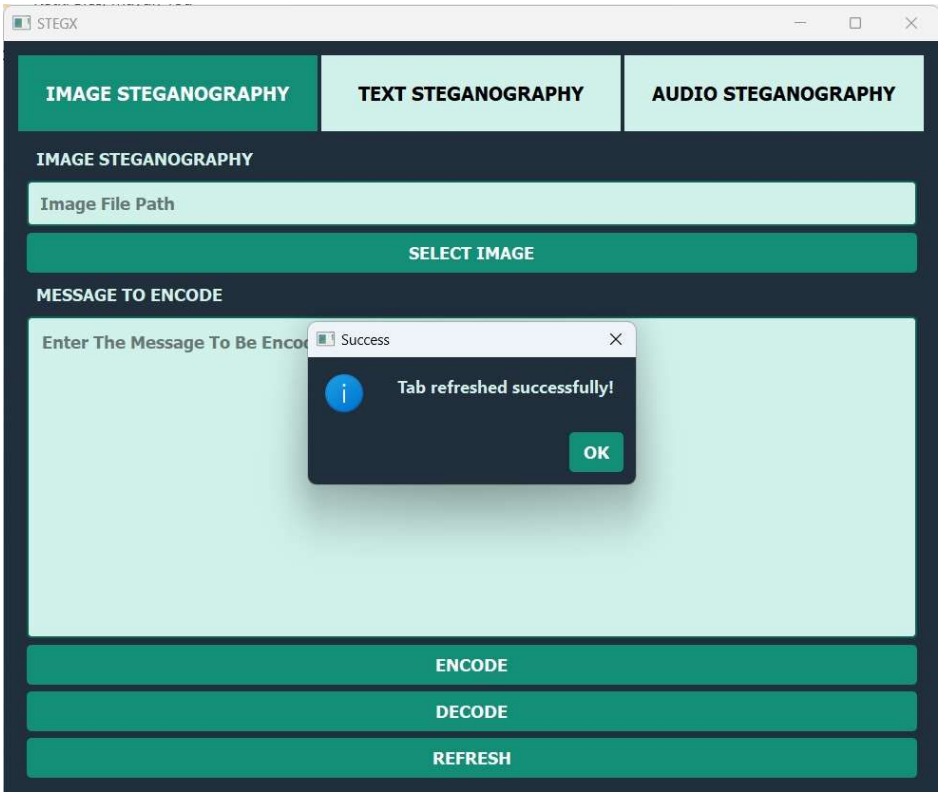

Fig 4.3 : Refresh Function in Image Steganography

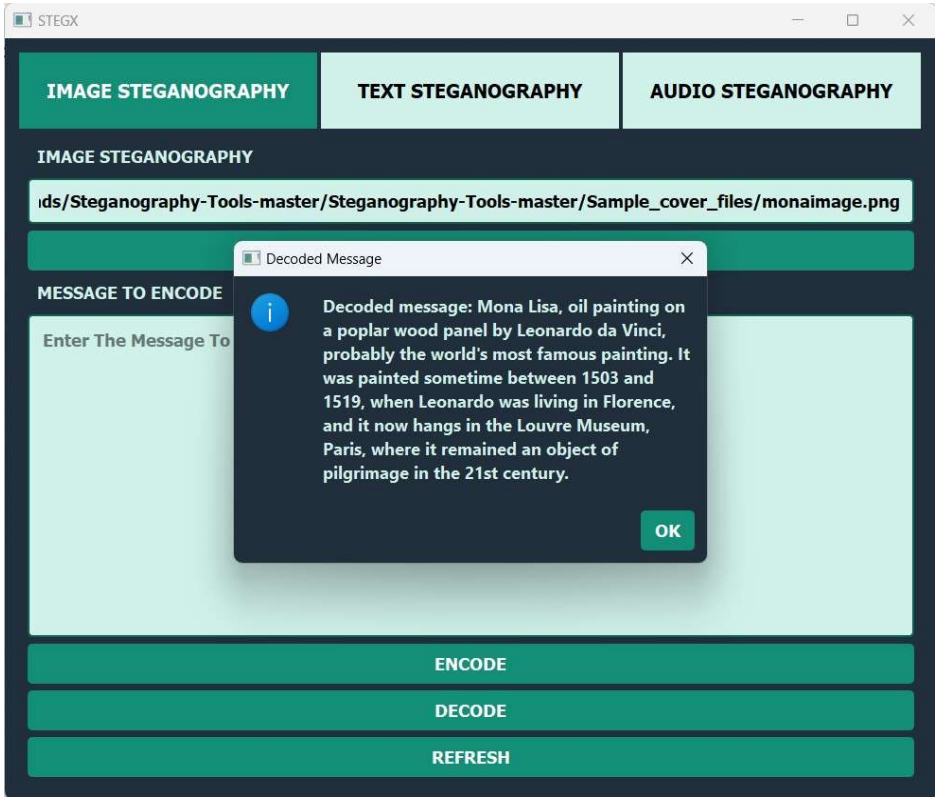

Fig 4.4 : Message Decoding in Image Steganography

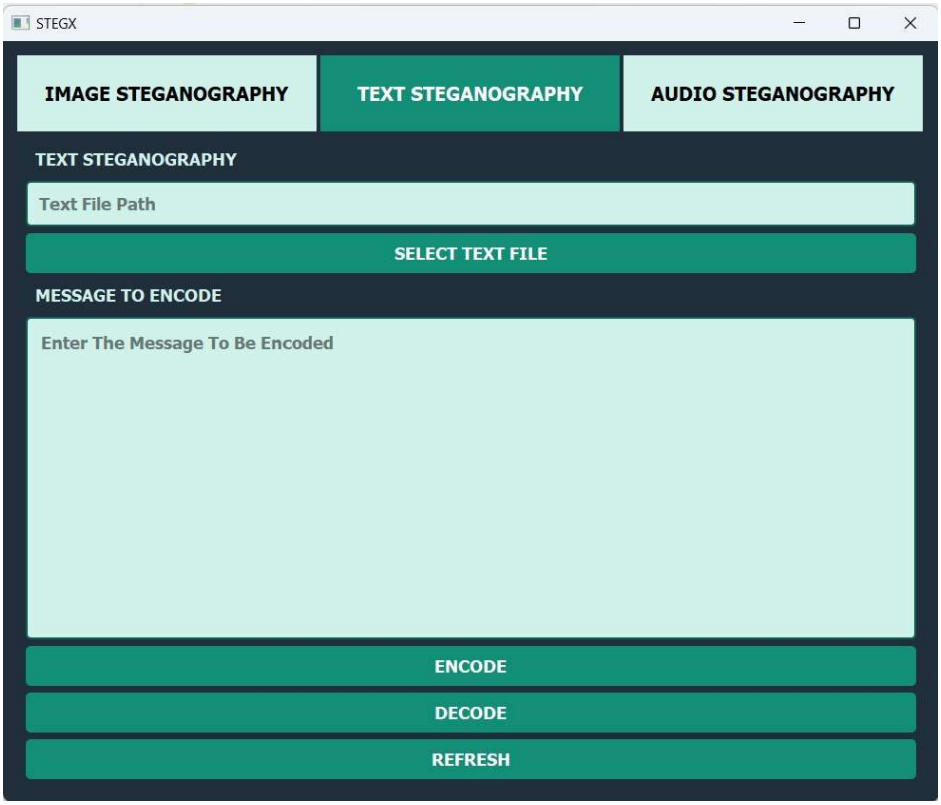

Fig 4.5 : Text Steganography GUI

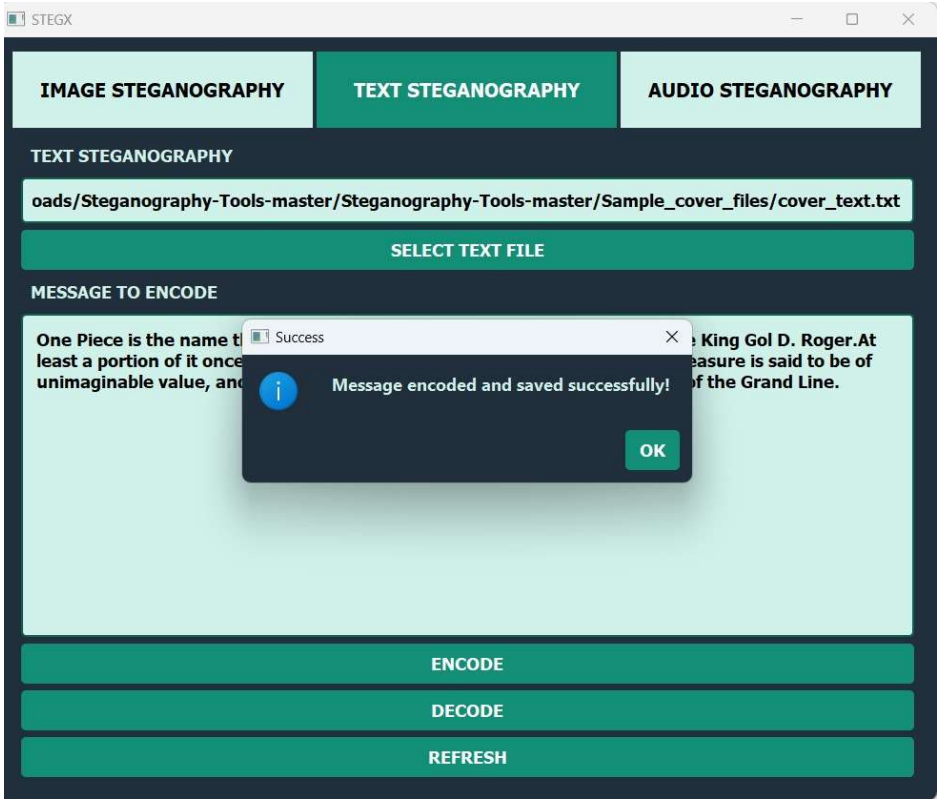

Fig 4.6 : Message Encoding in Text Steganography

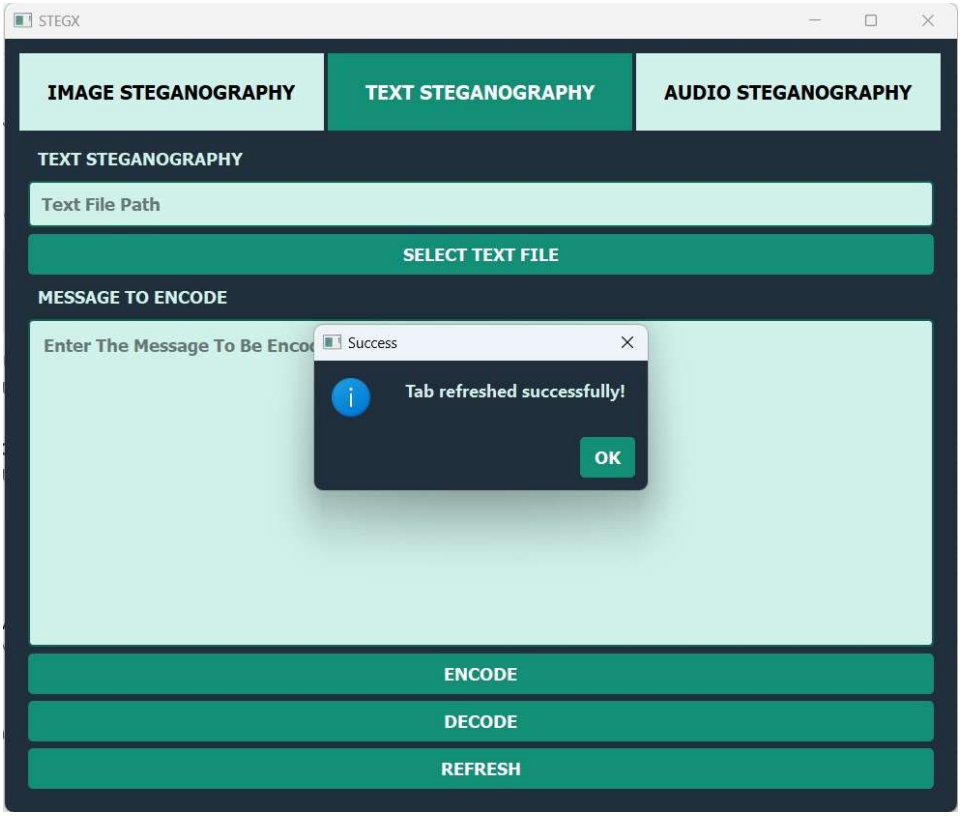

Fig 4.7 : Refresh Function in Text Steganography

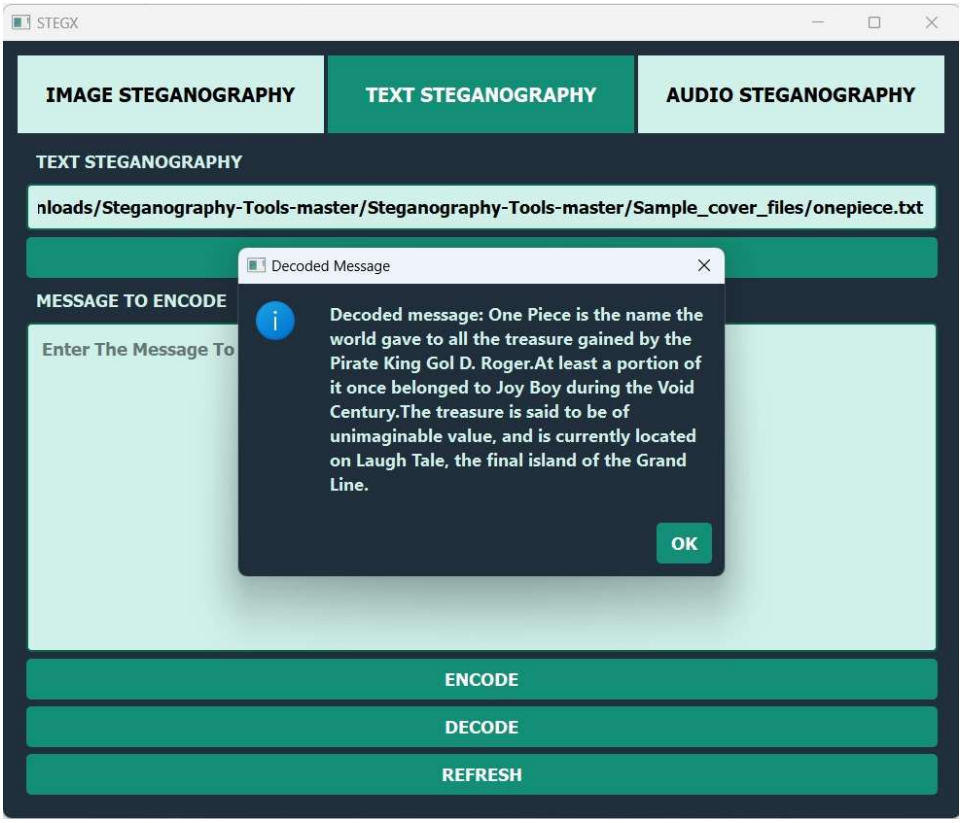

Fig 4.8 : Message Decoding in Text Steganography

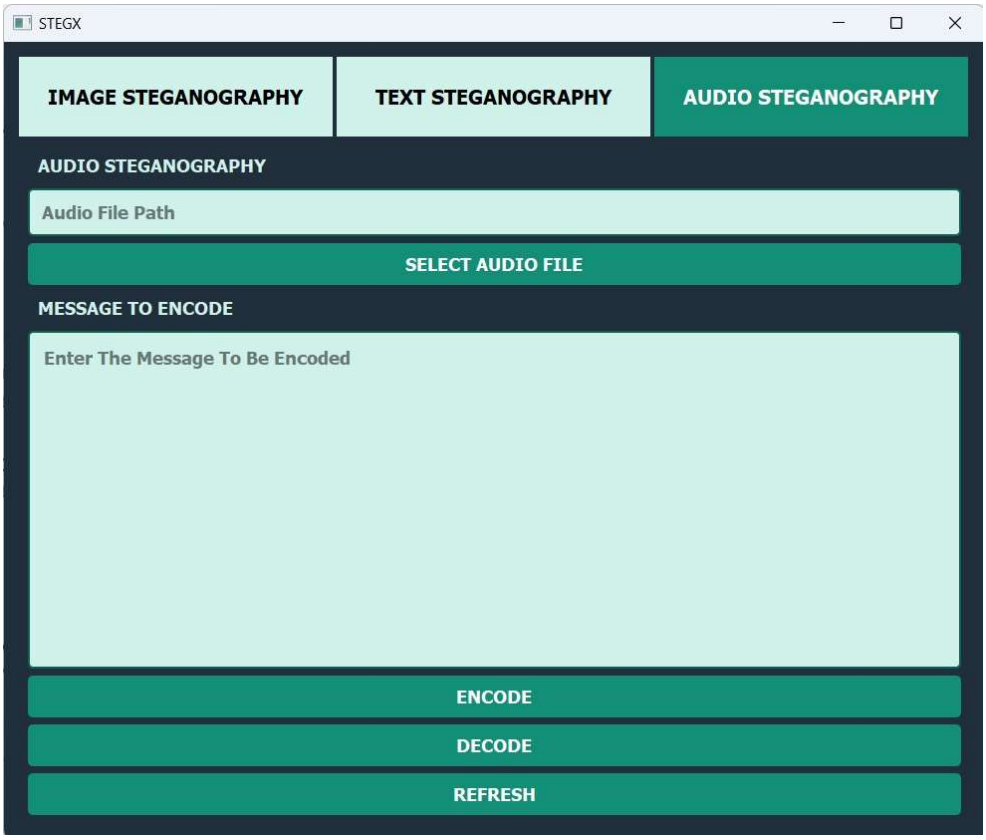

Fig 4.9 : Audio Steganography GUI

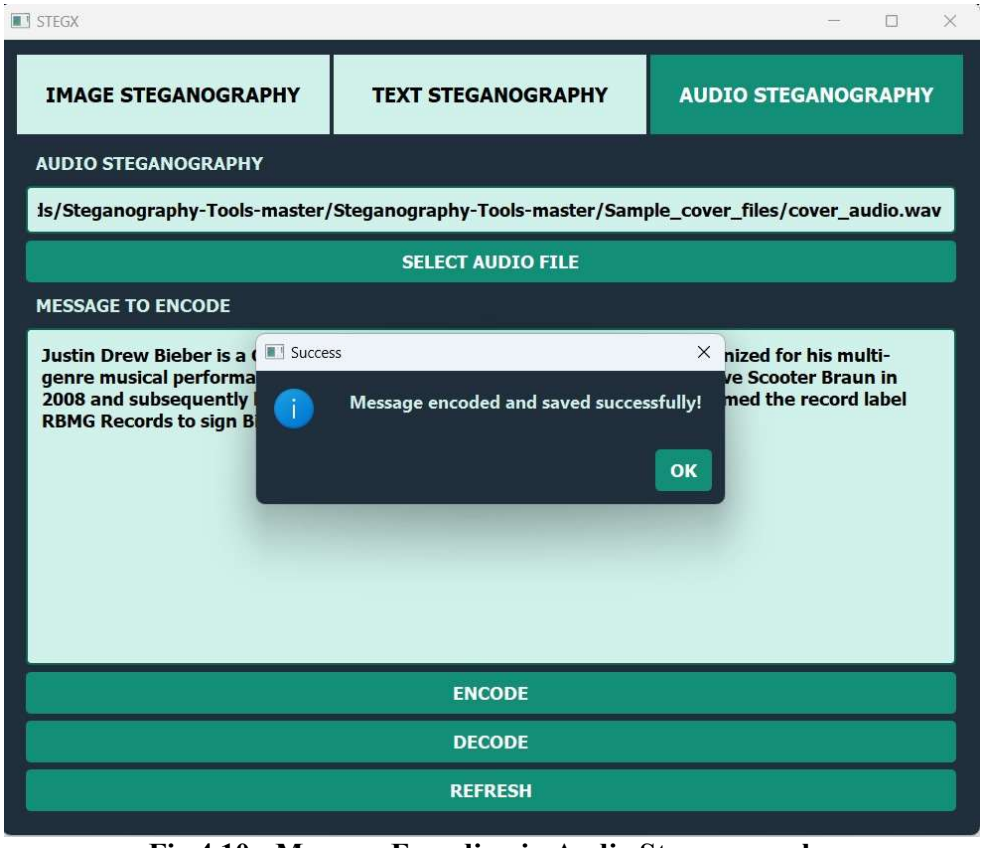

Fig 4.10 : Message Encoding in Audio Steganography

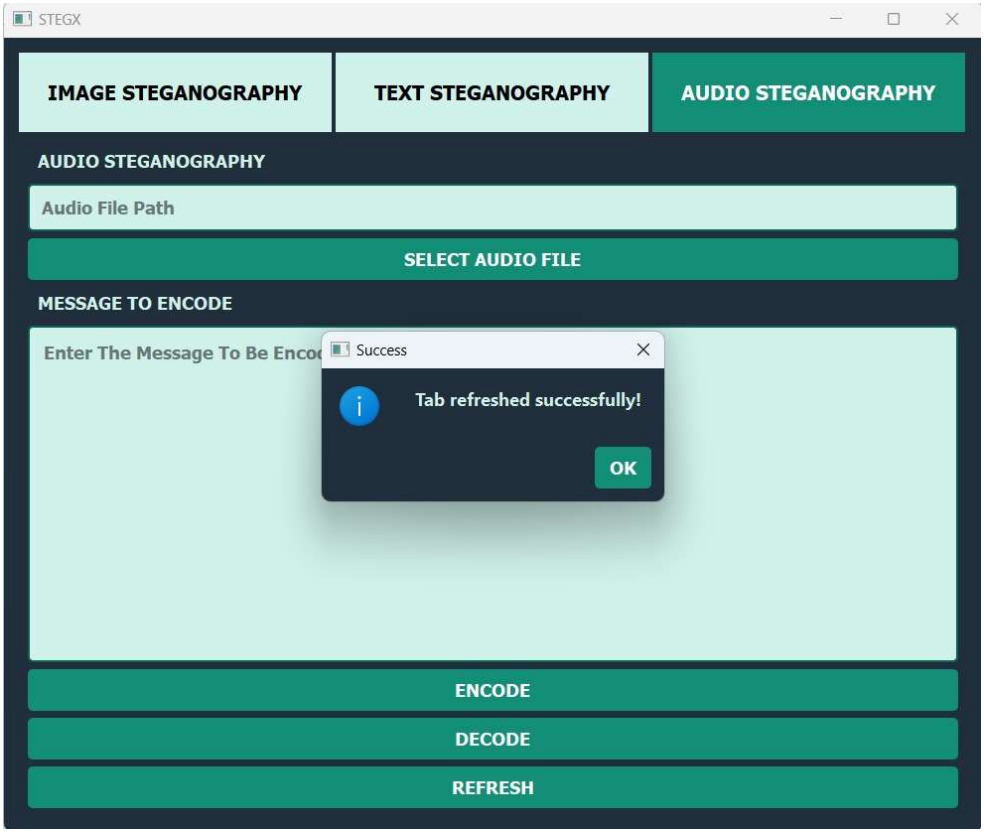

Fig 4.11 : Refresh Function in Audio Steganography

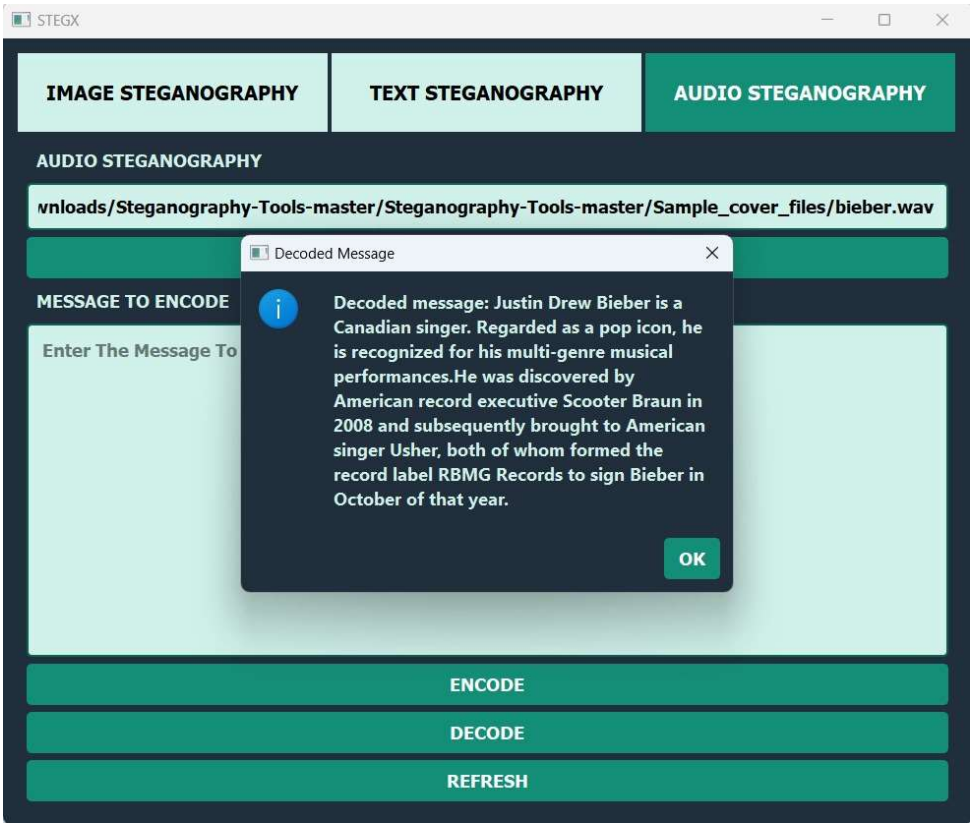

Fig 4.12 : Message Decoding in Audio Steganography

5. CONCLUSION

The successful development of this multi-media steganography tool showcases its ability to securely embed and extract hidden data within images, audio, and text files, providing a robust solution for enhancing digital privacy. Using the Least Significant Bit (LSB) manipulation technique, the tool effectively hides data in a way that remains imperceptible to the human eye or ear, while ensuring that the quality and integrity of the original media are preserved. The project focuses on delivering a highly reliable tool that supports key media types, making it versatile for various secure communication needs.

The graphical user interface (GUI), built using PyQt5, plays a critical role in making the tool accessible to users of all technical backgrounds. The intuitive design allows users to quickly select their preferred media type, upload files, and embed or extract data without needing in-depth knowledge of the underlying steganographic algorithms. This ease of use expands the tool's potential audience, making it suitable for both individuals seeking personal privacy and organizations looking for more discreet communication methods.

In conclusion, this project demonstrates the practicality and effectiveness of steganography as a tool for secure data transmission. By focusing on image, audio, and text steganography, the tool addresses a variety of use cases while maintaining simplicity and efficiency. It enables users to hide data discreetly, avoiding detection and ensuring that sensitive information can be shared securely across multiple media types. The successful implementation of this tool highlights steganography's potential to enhance privacy in the modern digital landscape, offering a critical layer of security for individuals and organizations alike.

6. REFERENCES

[1] M. Verma and H. S. Saini, "Analysis of Various Techniques for Audio Steganography in Data Security," Int. J. Sci. Res. in Network Security and Communication, vol. 7, no. 2, pp. 1-5, Apr. 2019. [Online]. Available: www.ijsrnsc.org. Accessed: Oct. 24, 2024.

[2] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and A. Bouridane, "Image Steganography: A Review of Recent Advances," IEEE Access, vol. 9, pp. 106051-106073, 2021. doi: 10.1109/ACCESS.2021.3100593.

[3] P. A. K. Maganbhai and K. Chouhan, "A Study and Literature Review on Image Steganography," Int. J. Sci. Res., vol. 6, no. 7, pp. 1-4, Jul. 2015.

[4] E. W. Abood, A. M. Abdullah, M. A. Al Sibahee, Z. A. Abduljabbar, V. O. Nyangaresi, S. A. A. Kalafy, and M. J. J. Ghrabta, "Audio Steganography with Enhanced LSB Method for Securing Encrypted Text with Bit Cycling," Bulletin of Electrical Engineering and Informatics, vol. 11, no. 1, pp. 185-194, Feb. 2022. doi: 10.11591/eei.v11i1.3279.

[5] J. Li, Z. Huang, and D. Huang, "Text Steganography in Social Media," IEEE Transactions on Multimedia, vol. 21, no. 3, pp. 709-719, Mar. 2019. doi: 10.1109/TMM.2018.2877764.

[6] X. Zhang, X. Luo, and Y. Cao, "Covert Communications Based on Text Steganography in Social Networks," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1957-1972, Sept. 2021. doi: 10.1109/TIFS.2021.3065625.

[7] L. Wu, R. Wang, Q. Hu, and X. Gao, "Efficient Deep Learning Model for Audio Steganalysis," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 3910-3925, Dec. 2021. doi: 10.1109/TIFS.2021.3108234.

[8] M. Islam, M. S. Amin, and M. B. Majumder, "A Secure and Imperceptible Text Steganography Approach Using Unicode of Bengali Characters," IEEE Access, vol. 8, pp. 183459-183468, 2020. doi: 10.1109/ACCESS.2020.3029356.

[9] H. Wang, W. Luo, S. Z. Li, and J. Huang, "Steganalysis of Text-Based Steganography in Open Domain Social Networks," IEEE Transactions on Information Forensics and Security, vol. 14, no. 11, pp. 2895-2908, Nov. 2019. doi: 10.1109/TIFS.2019.2907725.