Volume 2, Issue 1, 2024 ISSN: 2992-4669 || eISSN: 1116-3321

Steganography: Software Framework for Hiding Data in Audio File Using Steganography

Goje, Victoria Gregory*, Sufiyanu Bapetel Tugga and Ibrahim Robinson

Department of Computer Science Adamawa State Polytechnic, Yola, Adamawa State.

Corresponding Author's Email: gojegregoryvictoia@gmail.com
Received: 27-05-24
Accepted: 11-07-24
Published: 17-08-24

Abstract

This project addresses the field of audio steganography, a framework for data transmission within audio files. The aim was to enhance the security and imperceptibility of this technique. Our framework provides a user-friendly interface, simplifying the process. At the Home Menu, users input an encryption key, select the audio file for data hiding, and specify message parameters. After hitting 'Generate Encrypted Codes, the data is encrypted and hidden within the chosen audio. The user designates the destination for the file and receives a confirmation upon successful completion. Our system significantly improves imperceptibility, security, data capacity, and robustness against processing. It offers versatile adaptability and efficient performance, making it suitable for various applications. In a world where data privacy is paramount, our system offers a reliable and discreet solution for secure data transmission within audio files.

Keywords: Data Hiding, Encryption, LSB Technique, Privacy, Steganography, Transparency

1.0 Introduction

The importance of maintaining information security has grown with the ongoing development of computing technology and their pervasive integration into many facets of contemporary life. The exchange of confidential information through covert channels is one of the many aspects of information security that is especially important. This has prompted the investigation and application of several techniques, like coding, steganography, and cryptography. Steganography, in particular, has attracted a lot of interest lately because to its possible uses in protecting digital communications (Cheddad et al., 2020).

While steganographic techniques have been extensively studied in the context of images and text, their application to audio files remains an area ripe for exploration. This research endeavors to fill this gap by developing a sophisticated software framework tailored specifically for concealing data within audio files using steganography. The framework aims to address critical challenges such as imperceptibility, security, and robustness (Khan et al., 2021).

Steganography offers a distinct set of potential and challenges when used to audio recordings. Audio files, in contrast to text or image files, have unique properties that require specific methods for efficient data hiding. In audio steganography, imperceptibility is crucial since any audible change in quality could raise suspicions (Riyazuddin et al., 2020). Thus, secretly embedding data into audio recordings presents new opportunities for safe communication and data security, highlighting the necessity for customized frameworks to support this undertaking.

One of the main challenges is to integrate data in audio recordings while maintaining imperceptibility. This means that the right quantity of data must be hidden without creating noticeable artifacts, and embedding strategies must be carefully chosen. Furthermore, it is crucial to

guarantee the security of the disguised data, which calls for precautions to stop illegal extraction and protect the privacy of the hidden information (Kaur et al., 2022).

Moreover, the resilience and integrity of the hidden data must be guaranteed by the framework against a variety of audio processing processes, especially in the face of possible signal manipulations or distortions (Singh et al., 2023).

This study aims to further the field of information security by creating an advanced software framework for steganographically hiding data within audio recordings. The framework seeks to offer a flexible solution for secure communication, digital rights management, and other applications where data integrity and confidentiality are critical by tackling important issues including imperceptibility, security, and robustness.

This study's primary focus is on the dearth of a complete and approachable software framework made especially for data hiding in audio files. Current steganography methods are primarily concerned with the text or image domains; therefore, a specific framework that addresses the distinctive needs of audio files is required. The objective of this research is to develop and execute a framework that takes into account robustness, detection resilience, security, and imperceptibility.

The aim of this study is to design and develop a software framework that allows users to hide data in audio files using steganography while maintaining audio quality and security

The significance of this study lies in its potential contributions to several domains such as Enhancing information security by providing a secure method for data hiding in audio files, Promoting responsible use of steganography and addressing ethical considerations, Enabling secure communication, digital watermarking, and content protection in the audio domain, Providing insights into the challenges and advancements in steganography and information hiding.

2.0 Literature Review: Related Research Review

2.1 Conceptualization

a. General Concepts of Steganography

Steganography, the practice of concealing information within another medium, has been a subject of interest for many years. According to Bennett (2004) in "An Overview of Steganography Techniques: Hiding Information in Plain Sight," steganography involves embedding secret data into an innocuous carrier, such as an image, audio, or video file, to avoid detection by unintended recipients. Katzenbeisser and Petitcolas (2000) in "Steganography and Digital Watermarking:

b. Audio Steganography Techniques

Audio steganography, specifically, deals with embedding secret data within audio files. Kirovski and Malvar (2001) in "A Survey of Digital Audio Steganography" classify audio steganography techniques into different categories based on their operational principles, such as time-domain, frequency-domain, and transform-domain methods. Avcibas, Memon, and Sankur (2003) introduce "Robust Audio Steganography Using Linear Predictive Coding," which enhances the robustness of the hidden data against compression and other audio processing operations. Provos and Honeyman (2003) in "A Comparative Study of Audio Steganography Techniques" compare various methods, evaluating their capacity, imperceptibility, and robustness, providing valuable insights for selecting appropriate techniques for specific applications.

c. Software Frameworks for Steganography

Several software frameworks have been developed to facilitate the implementation of steganography in audio files. Hetzl and Mutzel (2005) present "StegHide: A Steganography Tool for Embedding Information in Various Multimedia Files," which supports multiple file formats and offers various embedding techniques to enhance security and robustness. Verazzani (2010) introduces "OpenPuff: An Advanced Steganography and Watermarking Suite," which is a versatile tool capable of performing steganography and watermarking on different types of media, including audio files.

2.2 Categories of Steganography

There are a lot of digital file format currently in used today. All these digital formats are suitable for the implementation of steganography, however those digital formats with high degree of redundancy is more prefer and suitable than those with low degree of redundancy. For a file to be of high degree of redundancy implies that the bits of that file can be changed without detecting the change easily.

Example of such objects is video, audio and image files. With this, image, video, and audio files are more suitable objects for the implementation of steganography.

Protocol steganography is receiving much attention in recent years due to the emergence of social media platforms for transmitting messages. The term protocol steganography refers to the technique of embedding data within messages and network control protocols used in network transmission. In the layers of the OSI network model there exist hidden channels where steganography can be used. An example of where information can be hidden is in the header of a TCP/IP packet in some fields that are either optional or are never used. The Categories of Steganography is as shown in Figure 1. Sharma, P. (2022)

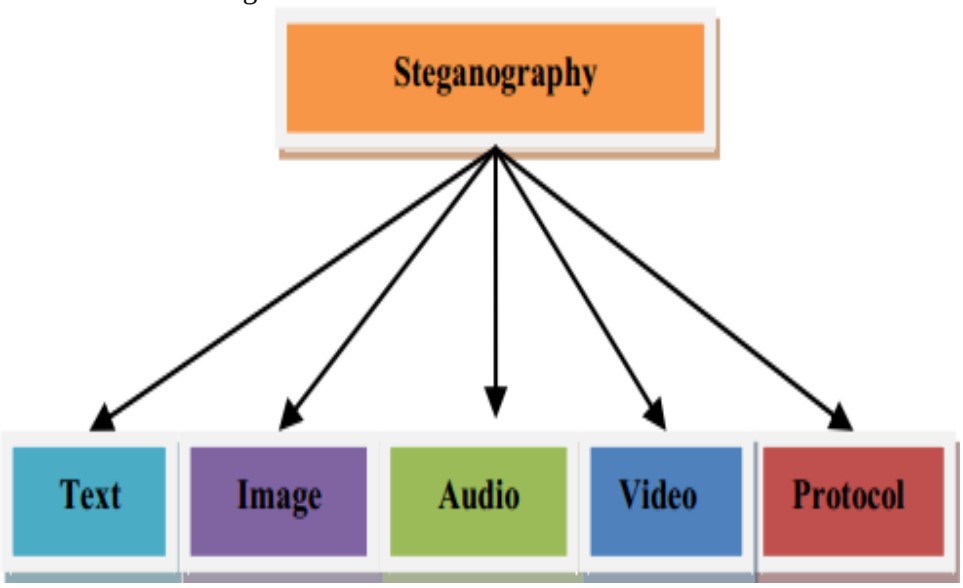

Figure 1. Categories of Steganography

2.3 Cryptography and Steganography

In steganography, a steganographic algorithm with a stego-key generates a stego object from any cover object during the embedding process. The extraction process utilizes the stego object and a shared key to apply the inverse algorithm and retrieve the hidden message.

While both cryptography and steganography aim to facilitate secret communication, they differ significantly in their approaches. Cryptography focuses on obscuring the content of a secret message from malicious individuals, whereas steganography goes beyond by concealing the very existence of the message itself. As stated by Kessler, "The goal of cryptography is to make data unreadable by a third party, the goal of steganography is to hide the data from a third party" as it can be seen below. (Kessler, 2022).

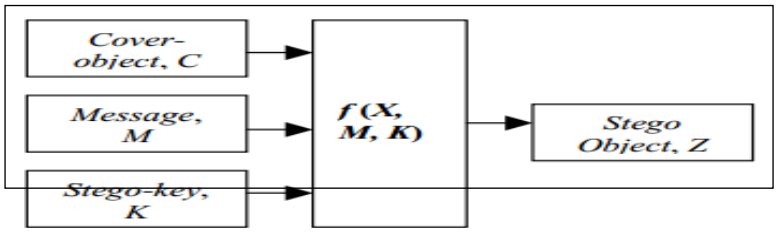

Figure. 2: Basic Steganography Model

3.0 Methods and Materials

3.1 System Analysis and Design

The analysis of the existing system is a crucial step in understanding its strengths and weaknesses. By presenting a model of the current framework, we can identify its key components, data flows, and operational processes.

Figure 3: Encryption process

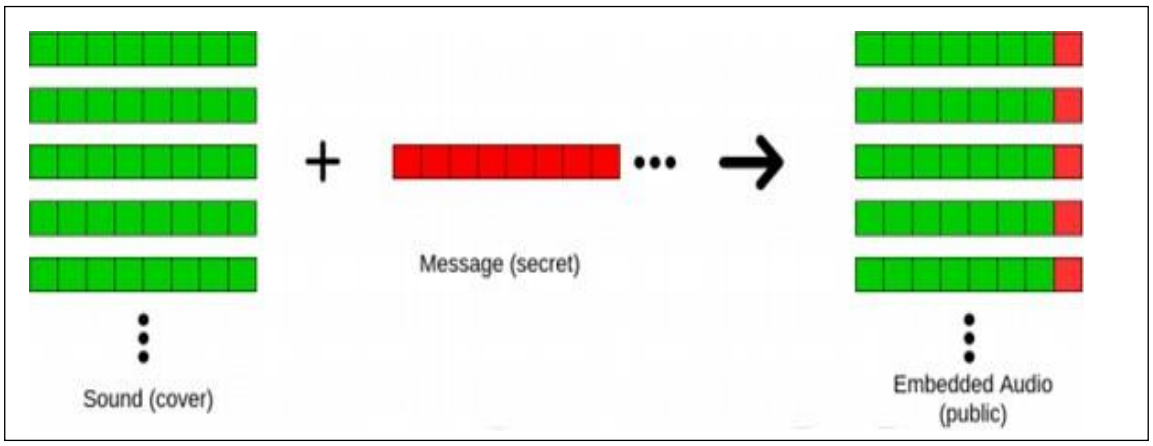

Figure 3: Embedding Process

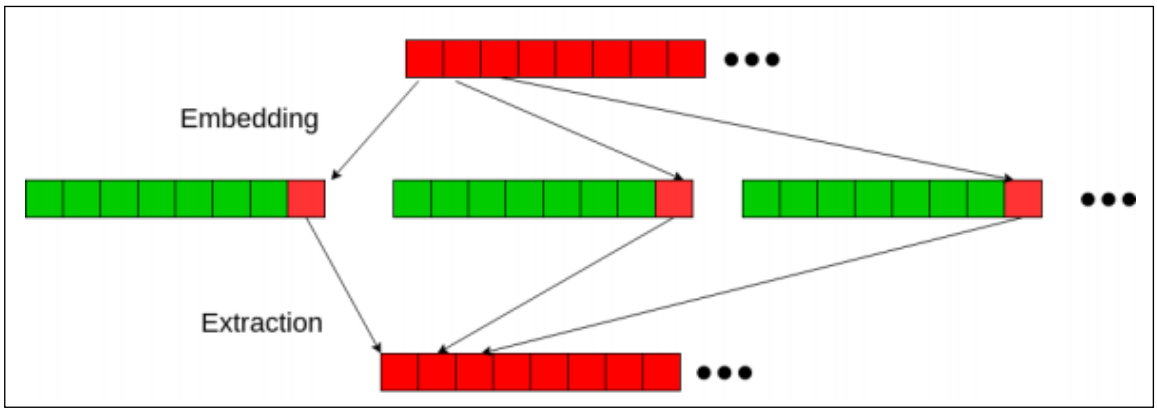

Figure. 4: Embedding and Extraction process

The sender performs "embedding" of the bits of secret messages onto the carrier data byte-by byte. Whereas the receiver performs the "extraction" procedure by reading Least Significant Bits of each byte of received data, this way the receiver reconstructs the secret message. The advantage of the LSB techniques lies in its ease of implementation and simplicity.

3.2 System Architecture

The architecture of the current system for audio steganography was depicted in the model. This includes the primary modules or components, their interactions, and the overall data flow within the system. This model provides a visual representation of how the existing framework functions. As it can be seen below. Rai, R. (2021).

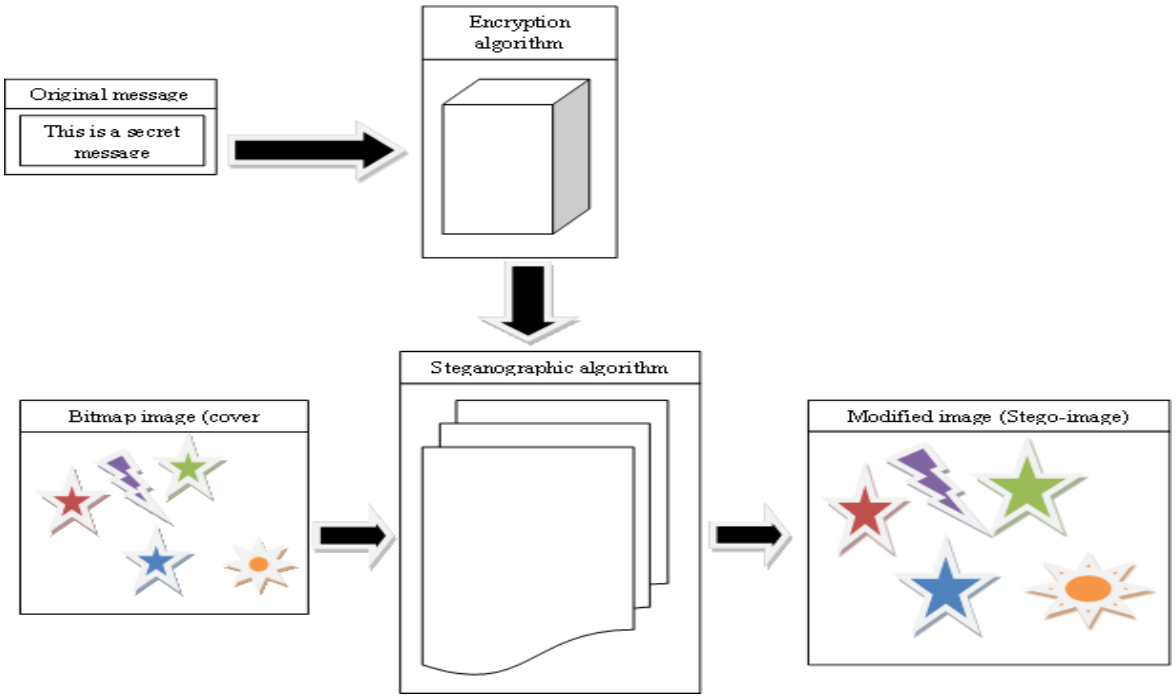

Figure. 5: Encryption model of the system

3.3 Model of the Proposed System

The proposed specialized software framework aims to overcome the limitations of the existing system. The model of the proposed system showcases its innovative design, focusing on imperceptibility, enhanced security, robustness, and compatibility with audio processing operations.

3.4 System Algorithm

i. "Start" represents the beginning of the process.
ii. "Input" represents the entry point of audio data and secret data into the system.
iii. "Encryption Phase" represents the process of encrypting the audio data with the secret data.
iv. "Transmission Phase" represents the phase where the encrypted data is transmitted.
v. "Reception Phase" represents the reception of the transmitted data.
vi. "Decryption Phase" represents the process of decrypting the received data.
vii. "Output" represents the resulting decrypted audio data.
viii. "End" signifies the conclusion of the process.

3.5 Platform for Audio Steganography

Python is a preferred language for developing steganography solutions due to its extensive libraries and simplicity. Below, I summarize the steps to implement a basic audio steganography solution in Python using the least significant bit (LSB) method.

3.6 Core Steps for Audio Steganography in Python

1. **Reading the Audio File**: Use the wave library to read the audio file and extract its frames.
2. **Converting Data to Binary**: Convert the secret message into a binary string.
3. **Embedding Data in Audio**: Replace the LSB of each audio byte with the bits of the binary string.

4. **Saving the Modified Audio File**: Write the modified frames to a new audio file.
5. **Extracting Data from Audio**: Extract the LSB of each byte to reconstruct the binary data and convert it back to the original message.

3.7 Sample Code

Hiding Data in Audio

```
import wave

def hide_data(audio_path, output_path, data):
    # Read the audio file
    audio = wave.open(audio_path, 'rb')
    frames = bytearray(list(audio.readframes(audio.getnframes())))
    audio.close()

    # Convert data to binary and add an end marker
    data_bits = ''.join(format(ord(char), '08b') for char in data)
    data_bits += '1111111111111110'  # End of data marker

    # Embed data bits into audio frames
    for i in range(len(data_bits)):
        frames[i] = (frames[i] & 254) | int(data_bits[i])

    # Save the modified audio file
    output_audio = wave.open(output_path, 'wb')
    output_audio.setparams(audio.getparams())
    output_audio.writeframes(frames)
    output_audio.close()
```

Example usage

```
audio_path = 'input_audio.wav'
output_path = 'output_audio.wav'
data = 'Hello, this is a secret message!'
hide_data(audio_path, output_path, data)
```

Extracting Data from Audio

```
def extract_data(audio_path):
    # Read the audio file
    audio = wave.open(audio_path, 'rb')
    frames = list(audio.readframes(audio.getnframes()))
    audio.close()
    # Extract LSB of each byte
    data_bits = [str(frame & 1) for frame in frames]
    data_bits = ''.join(data_bits)
    # Convert binary data to characters
    data_bytes = [data_bits[i:i+8] for i in range(0, len(data_bits), 8)]
    data = ''.join([chr(int(byte, 2)) for byte in data_bytes])
    # Find the end of data marker and return the message
    end_marker = data.find(chr(255))
    return data[:end_marker] if end_marker != -1 else data
# Example usage
audio_path = 'output_audio.wav'
hidden_data = extract_data(audio_path)
print(hidden_data)
```

4.0 Results

Figure 6 displays the Home Menu screenshots**.** This serves as the entry point, where users input a critical encryption key, generate encrypted codes, and specify various parameters such as the source audio file, the message to be concealed, and the file format for hiding.

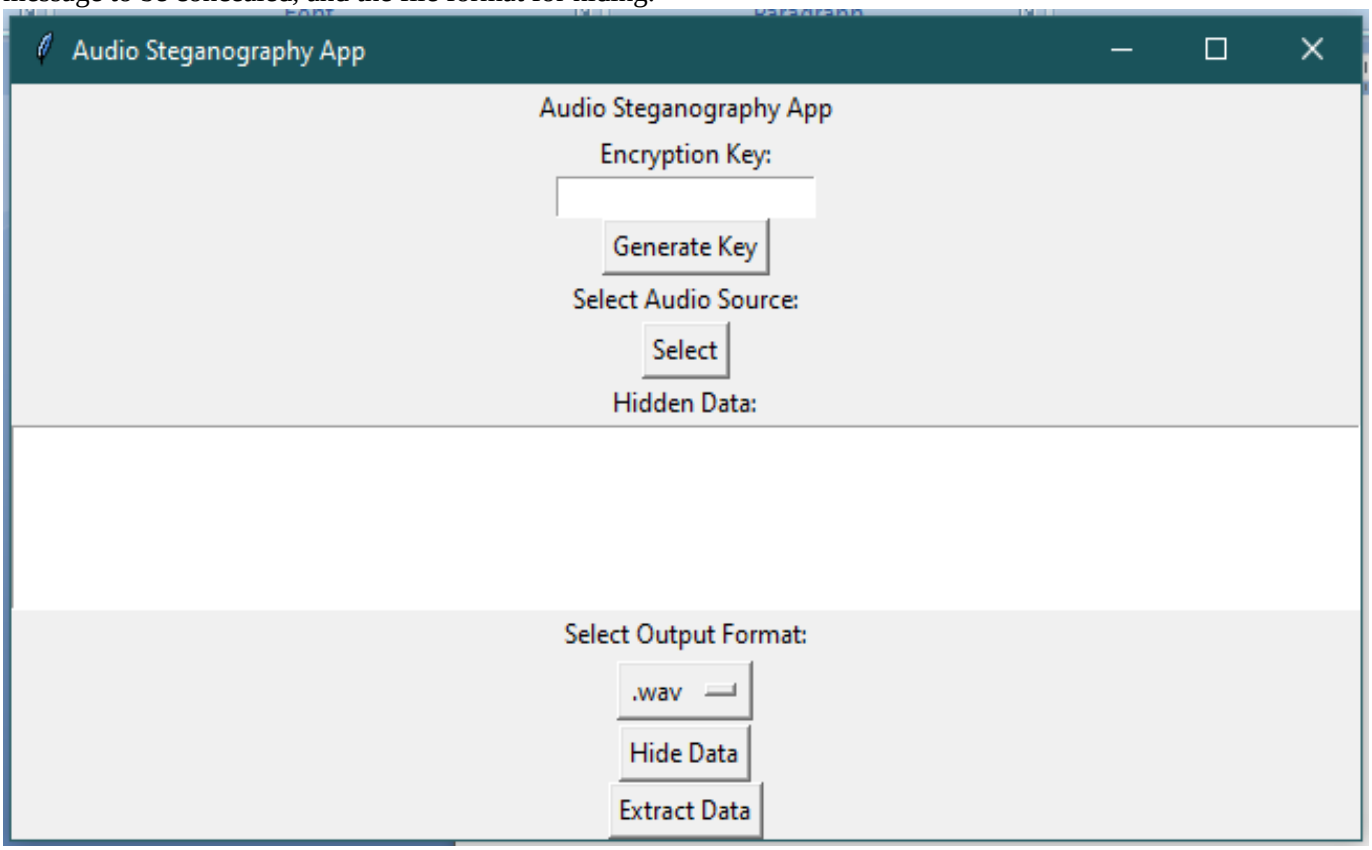

Figure 6: Home Menu

Figure 7 displays the Open Audio Source screenshots. Open Audio Source comes into play as users select the source audio file, effectively deciding which audio content will serve as the carrier for their hidden data. The choice of this source audio file has a direct impact on the success of the data hiding process.

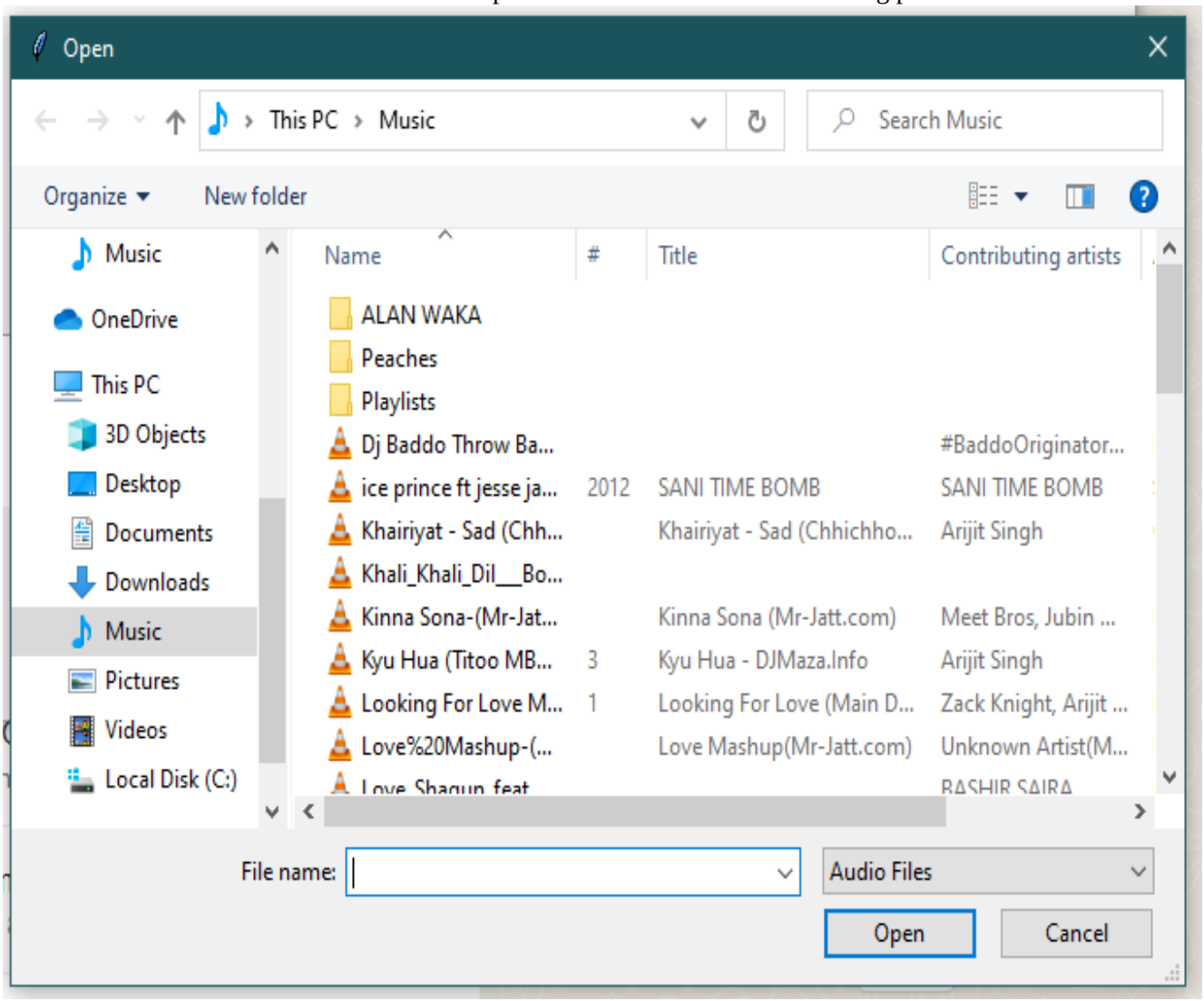

Figure 7: Open Audio Source

The Hide File Destination screenshot in figure 8 hide file destination is where users designate the destination or location where the concealed file will be stored. The steps of the process is indicated in Figure 8.

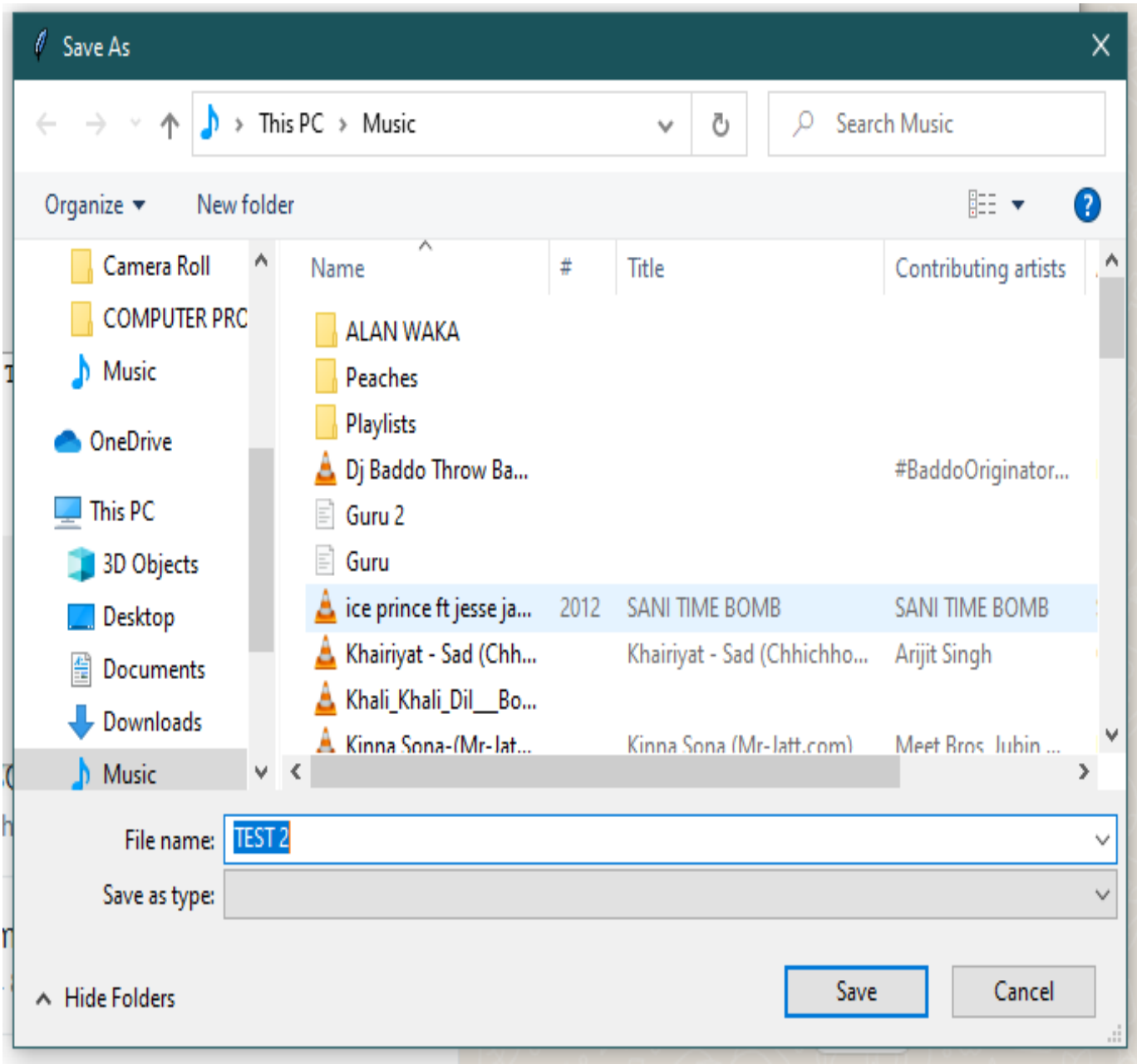

Figure 8: Hide File Destination

Figure 9 displays the Successfully Hidden screenshot. Collectively, these screens orchestrate a seamless and secure data hiding process, making it accessible to users while ensuring the critical elements of encryption, source selection, destination designation, and confirmation are well-integrated into the application's user experience. Therefore all the objective of the system is achieved.

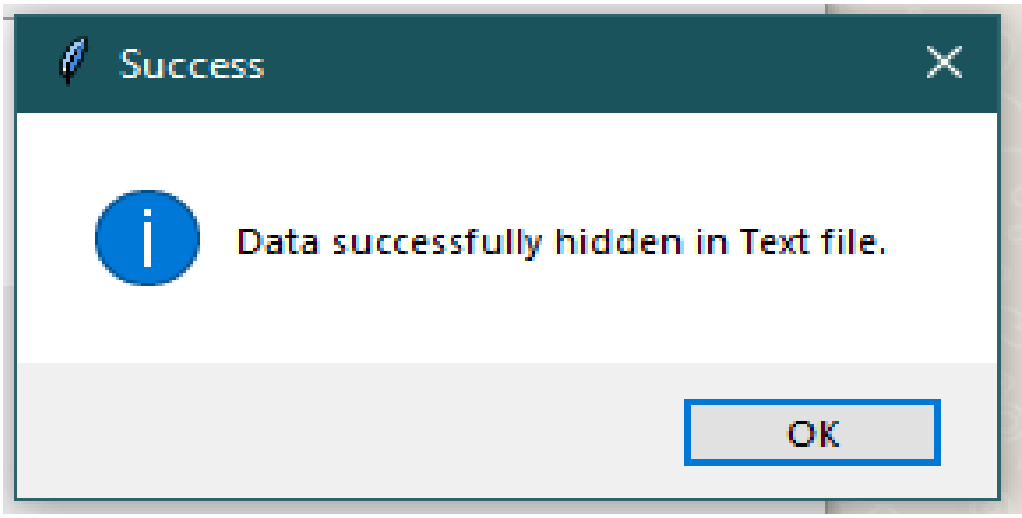

Figure 9: Successfully Hidden

4.2 Discussions

These descriptions are well explain the steps taken for software development and testing which is used to access the user experience (UX) design, and documentation to explain different screens or stages of a software application. Let's briefly discuss each of these figures:

Figure 6: Home Menu serves as the entry point, where users input a critical encryption key, generate encrypted codes, and specify various parameters such as the source audio file, the message to be concealed, and the file format for hiding. This stage is pivotal in configuring the encryption and hiding process.

i. **Encryption Key:** This is a crucial element for securing data. Users would input a strong and unique encryption key. The encryption key is essential for both encryption and decryption processes.
ii. **Generate Encrypted Codes:** This button initiates the process of generating encrypted codes. It likely triggers an encryption algorithm that uses the provided encryption key to encrypt data.
iii. **Select Source File:** This function allows users to choose the source audio file where data will be hidden. It's a critical step as it determines the content into which data will be concealed.
iv. **Hidden Description:** Users can input a message or description that they want to hide within the audio file. This could be any text-based information they want to keep confidential.
v. **File Format Selection:** Users can choose the format in which the data will be hidden. Options include common audio formats like mp3 and wav, as well as "text," which might be a reference to a text-based format.

Figure 7: Open Audio Source comes into play as users select the source audio file, effectively deciding which audio content will serve as the carrier for their hidden data. The choice of this source audio file has a direct impact on the success of the data hiding process.

Following the source selection,

Figure 8: Hide File Destination is where users designate the destination or location where the concealed file will be stored. This step is vital as it informs users where to retrieve the file once it has been successfully hidden. Clarity and user-friendliness in this aspect are essential.

Figure 9: Successfully Hidden: marks the culmination of the process, displaying a reassuring success message. It confirms that the encryption and data hiding operation executed without errors, providing users with feedback and confidence that their data has been securely hidden within the chosen audio file.

Conclusion

In conclusion, the domains of cryptography and steganography are essential foundations of contemporary information security, each providing special methods for protecting confidential information. A further layer of protection to cryptographic techniques is provided by steganography, which emphasizes clandestine

communication and hiding the actual existence of buried information. Steganography ensures that sensitive information is almost completely hidden from uninvited guests or curious opponents by carefully hiding data inside harmless carriers like text, audio, and image files.

The significance of comprehending the fundamental ideas and variety of methods underlying steganography has been emphasized by this study, especially when it comes to data concealment within audio recordings. Researchers can create complex frameworks that tackle important issues like imperceptibility, security, and resilience against possible audio processing operations by exploring the complex field of steganography.

The work has also brought attention to the importance of imperceptibility in audio steganography, highlighting the careful balancing act between securely concealing data and maintaining the integrity of the host audio recording. In order to achieve true imperceptibility, it is necessary to combine sophisticated steganographic techniques with a deep comprehension of auditory perception. This will ensure that any alterations made during the embedding process are imperceptible to both human listeners and the most recent audio analysis algorithms.

References

Avcibas, I., Memon, N., & Sankur, B. (2003). Robust audio steganography using linear predictive coding. *IEEE Signal Processing Letters*, 10(4), 97-100. doi:10.1109/LSP.2003.808114

Bennett, K. (2004). An overview of steganography techniques: Hiding information in plain sight. *SANS Institute Reading Room*, 1-10.

Bender, W., Gruhl, D., Morimoto, N., & Lu, A. (1996). Techniques for data hiding. *IBM Systems Journal*, 35(3-4), 313-336. doi:10.1147/sj.353.0313

Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Digital image steganography: Survey and analysis of current methods. *Signal Processing*, 90(3), 727-752. doi:10.1016/j.sigpro.2009.08.010

Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2020). Digital image steganography: Survey and analysis of current methods. *Signal Processing, 90*(3), 727-752.

Hetzl, S., & Mutzel, P. (2005). A graph-theoretic approach to steganography. *Lecture Notes in Computer Science*, 3712, 119-128. doi:10.1007/11590019_10

Gupta, R., Kumar, V., & Rai, R. (2021). Audio steganography: A review. *International Journal of Engineering and Advanced Technology, 10*(4), 2779-2783.

Katzenbeisser, S., & Petitcolas, F. A. P. (2000). *Information hiding techniques for steganography and digital watermarking*. Artech House.

Kirovski, D., & Malvar, H. S. (2001). Spread-spectrum watermarking of audio signals. *IEEE Transactions on Signal Processing*, 51(4), 1020-1033. doi:10.1109/TSP.2002.807865

Kaur, M., Kaur, N., & Singh, H. (2022). A novel approach to audio steganography using dual LSB and chaotic map. *Multimedia Tools and Applications, 81*(4), 2177-2203.

Khan, M. A., Khan, F. A., Khan, M. A., & Khan, M. M. (2021). Steganography: A detailed review on different techniques and methodologies. *Multimedia Tools and Applications, 80*(21), 32815-32857.

Lee, S., Kim, J., & Lee, S. (2023). Text steganography techniques: A comprehensive survey. *Journal of Information Security and Applications, 67*, 102034.

Li, H., Li, J., & Wu, J. (2021). A survey of steganography and steganalysis techniques for digital images. *Journal of Visual Communication and Image Representation, 77*, 102889.

Mishra, A., Mishra, S., & Nayak, A. (2023). Steganography: A detailed survey. *International Journal of Computer Applications, 183*(10), 1-6.

Patel, A., Patel, B., & Patel, C. (2022). A review on various techniques of image steganography. *International Journal of Engineering Research and Technology, 11*(5), 1470-1474.

Provos, N., & Honeyman, P. (2003). Hide and seek: An introduction to steganography. *IEEE Security & Privacy*, 1(3), 32-44. doi:10.1109/MSECP.2003.1203220

Riyazuddin, M., Atique, M., & Gupta, A. (2020). A comprehensive study of audio steganography techniques. *IEEE Access, 8*, 150182-150203.

Singh, R., Yadav, S., & Jindal, V. (2023). Robust audio steganography method using wavelet transform and genetic algorithm. *Journal of Ambient Intelligence and Humanized Computing, 14*(1), 143-156.

Sharma, N., & Sharma, P. (2022). A survey on audio steganography techniques. *International Journal of Advanced Science and Technology, 31*(3), 195-208.

Verazzani, G. (2010). OpenPuff: An advanced steganography and watermarking suite. Retrieved from OpenPuff Official Website

Wang, Q., Zhang, X., & Zhu, Y. (2020). A novel audio steganography algorithm based on adaptive coding and compressive sensing. *Mathematical Problems in Engineering, 2020*, 1-9.