Journal of Science, Engineering & Technology Project Reports

Advanced Data Security using Modulo Operator and LSB Method

R. H. Basavaraj,* A. Srikanth, R. Prajwal Praneeth Kumar, U. Vinay and C. Rohini

Department of Computer Science and Engineering,
Impact College of Engineering and Applied Sciences,
Sahakarnagar, Bengaluru-560092, INDIA

R. H. Basavaraj: **Corresponding Author*: email: basavarajappu2001@gmail.com;
Cell: +917624828236
A. Srikanth: email: srikanthtony71@gmail.com
R. Prajwal Praneeth Kumar: email: prajwal18861@gmail.com
U. Vinay: email: vinayudaya2001@gmail.com
C. Rohini: Guide: email: rohinichandrappa5@gmail.com

PROJECT REPORT INFO Received 20th May 2023, Accepted 27th June 2023,
www.esrapublications.com

ABSTRACT

Steganography is a practice of hiding information within another piece of information such as: a digital image, an audio or a video file in a way that, it is undetectable to the human sense or standard analysis techniques. The method involves replacing the least significant bit of each pixel or audio/video sample with the bits of the secret message to be hidden. The changes made to the LSBs are usually imperceptible to the human eye or ear. The proposed technique for image steganography involves embedding the secret message in the LSBs of the image pixels, while maintaining the image quality and size. For audio steganography, the secret message is hidden in the least significant bits of the audio samples, without affecting the audio quality. Finally, for video steganography, the technique involves hiding the message in the LSBs of the video frames, without causing any visual degradation or noticeable changes in the video quality.

*Address for correspondence: Department of Computer Science and Engineering, Impact College of Engineering and Applied Sciences, Sahakarnagar, Bengaluru-560092, INDIA

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgaum-590014, Karnataka, INDIA

A Project Work Phase-2 Report on

"ADVANCED DATA SECURITY USING MODULO OPERATOR AND LSB METHOD"

Submitted in partial fulfilment of the requirements for the award of the degree

Bachelor of Engineering
in
Computer Science Engineering

SUBMITTED BY

R. H. BASAVARAJ*	**[1IC19CS024]**
A. SRIKANTH	**[1IC19CS001]**
R. PRAJWAL PRANEETH KUMAR	**[1IC19CS018]**
U. VINAY	**[1IC19CS028]**

UNDER THE GUIDANCE
OF
Mrs. ROHINI C
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
IMPACT COLLEGE OF ENGINEERING AND APPLIED SCIENCES
Sahakarnagar, Bengaluru-560092, INDIA

2022-2023

IMPACT COLLEGE OF ENGINEERING AND APPLIED SCIENCES

Sahakarnagar, Bengaluru-560079, INDIA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that, the Project Work Phase-2 entitled **“ADVANCED DATA SECURITY USING MODULO OPERATOR AND LSB METHOD”** carried out by **BASAVARAJ, R. H [1IC19CS024], A. SRIKANTH [1IC19CS001], PRAJWAL PRANEETH KUMAR, R [1IC19CS018]** and **U. VINAY [1IC19CS028]** bonafide students of the Impact College of Engineering and Applied Sciences has been submitted in partial fulfilment of requirements of Eight Semester **Bachelor of Engineering in Computer Science and Engineering** as prescribed by the **Visvesvaraya Technological University**, Belagavi, Karnataka State, INDIA during the academic year 2022–2023.

Name of the Guide	**Name of the HOD**	**Name of the Principal**
Mrs. ROHINI, C	**Dr. DHANANJAYA, V**	**Dr. JALUMEDI BABU**

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the Project Work Phase-2 would be incomplete without the mention of the people who made it possible, without whose constant guidance and encouragement would have made our efforts go in vain.

We consider ourselves privileged to express our gratitude and respect towards all those who guided us throughout the completion of this Project Work Phase-2 on "**ADVANCED DATA SECURITY USING MODULO OPERATOR AND LSB METHOD**".

We consider ourselves proud to be a part of the **Impact College of Engineering and Applied Sciences, Bengaluru, INDIA**, the institute which stood by us throughout our endeavors.

We would like to thank our guide **Mrs. ROHINI C**, Assistant Professor, Department of Computer Science and Engineering, for the support and encouragement provided throughout the completion of the Project Work Phase-2.

We are grateful to **Dr. DHANANJAYA V**, Professor and Head, Department of Computer Science and Engineering, for his support and encouragement provided throughout the process.

We would like to express our gratitude to the Management and the Principal **Dr. JALUMEDI BABU**, Impact College of Engineering and Applied Sciences, for providing us the infrastructure and congenial environment to nourish our knowledge.

We also express our gratitude and sincere thanks to all the teaching and non-teaching Staff of the Department of Computer Science and Engineering.

Finally, we would like to express our heartfelt thanks to our Parents and Friends for their unconditional support and help during the course of the Project Work Phase-2.

A. SRIKANTH	[1IC19CS001]
PRAJWAL PRANEETH KUMAR, R	[1IC19CS018]
BASAVARAJ, R. H	[1IC19CS024]
U. VINAY	[1IC19CS028]

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 STEGANOGRAPHY

Steganography is the practice of concealing a message within another message or a file such as: an image or an audio file in such a way that, the existence of the hidden message or the file is concealed. The goal of the steganography is to hide the existence of the message or a file, rather than to encrypt it so that, it can be communicated in plain sight without anyone suspecting that there is a hidden message or a file.

Steganography can be used for various purposes, such as: covert communication or digital watermarking. One example of steganography is to hide a message within the pixels of an image by slightly altering the color values of the pixels. This alteration is usually imperceptible to the human eye but can be detected by a computer program.

Steganography is often used in combination with encryption to provide an additional layer of security. By encrypting a message and then hiding it within another file, it becomes much more difficult for an attacker to discover the message, as they would need to both decrypt the message and find it within the cover file.

Steganography techniques can also be used to embed information in audio files, video files, and other types of digital media. For example, audio files can be modified by slightly altering the amplitude of certain frequencies to embed a message, while video files can be modified by slightly altering the color values of certain pixels.

One of the challenges of steganography is to balance the concealment of the message with the quality of the cover file. If too much data is hidden in the cover file, it may be noticeable and raise suspicion. On the other hand, if too little data is hidden, the message may be too weak to be detected.

Steganography has been used throughout history for various purposes, including espionage, military communications, and even in literature. Today, steganography has been found to be useful in a wide range of applications such as: digital watermarking, copyright protection and secure communication.

While steganography can be a powerful tool for hiding information, it is not foolproof and can be detected with the right tools and techniques. As such, it should not be relied upon as the sole means of securing sensitive information.

1.2 PYTHON

Python is a very popular and a high-level, programming language which was released in 1991 by Guido van Rossum. It is used widely in a variety of applications such as: web design, artificial intelligence, computing scientific data, machine learning, data-analysis and in many more programs.

Python is simple syntax and easy to learn, it is found to be a very popular choice for the beginners. Python supports functional, procedural and object-oriented programming. It has a very large and highly active community, which contributes to an extensive library of packages and modules available for the various tasks.

Some notable features of Python include: auto memory management, its dynamic typing and an ability to run on diverse platforms. It also has a robust standard library for the developers and provides a wide range of functionalities.

Python is often used in conjunction with other technologies, such as: web frameworks like Django and Flask, scientific computing libraries like: NumPy and Pandas; and machine learning frameworks like: TensorFlow and PyTorch.

1.2.1 CHARACTERISTICS

- Easy to Use and easy to Learn: Python is simple and easy-to-read syntax, hence, is easy to use and easy to learn. It is a popular choice for beginners, as it has a relatively low learning curve.
- Interpreted Language: Python is an interpreted language; means, at run-time the code is executed line-by-line. The development process thus, becomes faster as the developers can see the result of their code immediately.
- Object-Oriented: Python also supports object-oriented programming (OOP). This permits developers to write reusable code and create the complex data structures.
- Dynamic Typing: Python is dynamically typed, which means, variables need not be declared with a specific type before use. This makes coding faster as the developers don't have to worry about the data types of variables.
- Large Standard Library: Python has a large standard library, and provides a broad range of modules and built-in functions. This reduces the amount of code that developers need to write from scratch.
- Cross-Platform: Python serves as a cross-platform language, that means, it has a capacity to run on multiple platforms such as: Windows, macOS and Linux.
- Third-Party Libraries: Python is a vast ecosystem of modules and third-party libraries which extends the language's functionality. These libraries cover a multiple range of use cases; from the web development function to the scientific computing.
- Scalable: Python is scalable, which means that it can be used for both small and large-scale projects. This makes it a popular choice for startups and large companies alike.

1.2.2 APPLICATIONS

- Web Development: Python is used for the web development, thanks to its robust web frameworks like: Django, Flask, Pyramid and more.
- Scientific Computing: Python is very useful in the scientific computing and for the data analysis, thanks to libraries like: NumPy, Pandas and SciPy.
- Artificial Intelligence and Machine Learning: Python is found to be useful in the machine learning applications and artificial intelligence, because of popular libraries like: TensorFlow, Keras, PyTorch and more.
- Gaming: Python is used in the gaming industry for developing game engines and frameworks, thanks to libraries like: Pygame, PyOpenGL and more.
- Education: Python is widely used in the education field for teaching programming and computer science concepts to the students.
- Finance: Python is used in finance for quantitative analysis, risk management, and trading automation, thanks to libraries like: Pandas, NumPy and more.
- Automation: Python is used in automation for tasks like web scraping, test automation and more.
- Desktop Applications: Python is used for developing desktop applications, thanks to libraries like: Tkinter, PyQT and more.

CHAPTER 2

LITERATURE SURVEY

1. *Video steganography based on embedding the video using PCF technique* by K. Rajalakshmi.; K. Mahesh in '2017 International Conference on Information Communication and Embedded Systems (ICICES)'.

Video steganography is considered as a method of hiding information and secret communication of the most significant problems occurred to secure the data transmission in the electronic era. The main purpose of this paper is important for the efficient transfer of the data and to maintain the secrecy of the data that is to be transmitted. From the past time to present time security of confidential information is always an important issue. In this paper, a novel technique is proposed by the authors to conceal the existence of the message so that it becomes tricky for attacker to notice it. This paper deals with video steganography algorithms to hide video file within another video using Patch Wise Code Formation technique. This technique provides robustness, better video quality and authentication ability. The performance of the process is measured in terms of the compression ratio using the metrics such as: PSNR, MSE, CR and BPP. [1]

Advantages

- High capacity: The technique can embed a significant amount of secret data in the video while maintaining good video quality.
- Good video quality: The technique does not significantly alter the video quality, which makes it suitable for applications that require secure communication of video data, such as video surveillance and military applications.
- Pixel complexity-based selection: The use of the PCF algorithm to select pixels for data embedding ensures that the changes made to the video are minimal, making it difficult to find the presence of hidden message.

Disadvantages

- Susceptibility to attacks: The technique is vulnerable to attacks by adversaries who are aware of the use of steganography and can use various methods to detect the presence of a hidden message.
- Limited applicability: The technique may not be suitable for videos with high levels of motion or rapidly changing scenes, as it relies on selecting pixels based on their complexity, which may not be appropriate in such scenarios.
- Complexity: The technique involves several steps, including the selection of pixels, message embedding, and extraction, which may make it difficult for users with limited technical knowledge to implement and use effectively.

2. *Enhanced LSB Steganography with people detection as stego key generator* by Pande Gede Pradnya Jaya, S.T.; Bambang, H.; Fiky Y. Suratman in '2019 International Conference on Intelligent Computing and Control Systems (ICCS)'.

Technology used to hide secret message in a communication is called Steganography. Secret message can be text, image or any file that can be converted into binary form. This secret message inserted into cover file which can be in the form of image, sound or video; basically cover file, must be bigger than the secret message in size. Many methods have been proposed on how to hide secret messages in a cover file. These methods include spatial domain which work on bit instead of statistically on cover file. Least Significant Bit have been long known as the simplest steganography embedding method. In this research the authors have proposed a method of LSB with enhanced technique to increase security. To extract secret message from cover file one should have particular stego key that describes the location of the messages and how to reconstruct them. They have utilized a people detection method as a stego key since most of the video footages involve human figure; from the personal video footage, movie video to the security camera recording. While maintaining the appearance of the cover video from arousing suspiciousness, the size of the secret message that can be embedded; which should be a point of consideration in this work. [2]

Advantages

- The paper proposes an enhanced method of LSB steganography which uses detection of people as a stego key generator that can provide an additional layer of security to the steganographic process.
- The developed method is designed to be resistant to attacks that are commonly used to detect steganography, such as: statistical analysis and visual inspection.
- The paper presents result which demonstrates the effectiveness of the proposed method in terms of steganographic capacity, image quality and detection resistance.
- The proposed method can be applied to various types of images, including grayscale and colour images as well as videos.

Disadvantages

- The proposed method relies on the accuracy of the people detection algorithm, which may be affected by various factors such as: lighting conditions, camera angles and occlusions.
- The paper does not provide a detailed analysis of the computational complexity of the proposed method, which may be a concern for real-time applications or large-scale steganography.
- The paper does not compare the developed methodology with other state-of-the-art steganographic methods, which makes it difficult to assess the relative performance of the proposed method.
- The paper also does not discuss the potential limitations of the proposed method, such as: its vulnerability to attacks which can specifically target the people detection algorithm.

3. *RSA Secured Web Based Steganography Employing HTML Space Codes and Compression Technique* by I. Bajaj.; R. K. Aggarwal in ‘2019 International Conference on Intelligent Computing and Control Systems (ICCS)’.

Web development has given its way to develop the strategies to secure the information available on the web. The data can be secured in numerous ways and one of the common way is to obnubilate the data through HTML web pages where the data is hidden in the source code of the web document. The web contains innumerable amount of personal data which needs to be hidden. The authors have proposed an algorithm which combines the techniques of cryptography and steganography which aids in securing the data by hiding it behind the source code of the HTML web pages. RSA encryption is used for encrypting the secret message. The encrypted data is compressed, mapped into Space codes and embedded under the HTML web pages by making mere changes in the code which cannot be noticed by the intruder as they are not suspicious. The robustness and security are enhanced in this method. [3]

Advantages

- The paper proposes a new steganography technique that employs RSA encryption, HTML space codes and compression techniques to ensure secure data transmission.
- The proposed technique is designed to resist attacks from hackers, data sniffers and unauthorized users.
- The use of compression techniques reduces the size of the data, making it easier and faster to transmit over the Internet.
- The use of RSA encryption ensures that the transmitted data is secure and can only be decrypted by authorized users who possess the private key.
- The proposed technique is web-based, easy to use and accessible from all types of devices with internet connectivity.

Disadvantages

- The paper does not provide enough details on the implementation of the proposed technique.
- The paper does not provide enough experimental results to determine the effectiveness of the developed method.
- The proposed technique may increase the processing time and computational power required to encrypt and decrypt the data, which may limit its practicality in some situations.
- The proposed technique may require a high level of technical expertise to implement its use effectively.
- The paper does not address the issue of key management, which is critical in ensuring the security of the transmitted data.

4. *A 3DES Double–Layer Based Message Security Scheme* by N. Adam.; M. Mashaly.; W. Alexan in ‘2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)’.

The paper "A 3DES Double-Layer Based Message Security Scheme" proposes a new message security scheme that uses a double-layered approach based on Triple Data Encryption Standard (3DES) to provide enhanced security for the transmitted messages. The proposed scheme employs two layers of encryption and decryption using 3DES to ensure the confidentiality and integrity of the message. The first layer of encryption uses the sender's private key and a randomly generated key to encrypt the message, which is then transmitted to the receiver. The receiver then decrypts the message using the sender's public key and the randomly generated key. The second layer of encryption uses 3DES to further encrypt the message using a session key, which is shared between the sender and the receiver. The proposed scheme is evaluated using various metrics, such as: encryption and decryption time and key space. The experimental results show that, the proposed scheme provides higher security than traditional message security schemes, while maintaining reasonable computation time and memory usage. The results suggest that, the proposed scheme can be used as a practical and effective approach for the message security in various applications, particularly those which require high security levels.[4]

Advantages

- The proposed scheme provides high security for message transmission by using a double-layer encryption approach with the 3DES algorithm.
- The scheme ensures data confidentiality, integrity and authentication by using symmetric key encryption and digital signatures.
- The proposed scheme can be effectively implemented in various network environments and applications.
- The performance evaluation results indicate that, the proposed scheme is efficient and practical for secure message transmission.

Disadvantages

- The proposed scheme does not consider the impact of network latency on the message transmission time.
- The scheme relies on a centralized key management system, which may not be suitable for some distributed network environments.
- The security of the scheme depends on the strength of the symmetric key used in the encryption process.
- The scheme does not provide forward secrecy, which means, compromise of a secret key can compromise past and future messages.

5. *Reversible Data Hiding Based Bit-Plan Permutation and Absolute Moment Block Truncation Coding (AMBTC)* by S. F. Hussein.; A. H. Radie in ‘2020 3rd International Conference on Engineering Technology and its Applications (IICETA)’.

Data hiding is one of the branches of information security, and it is used in the field of communications and computer networks which can be a solution when a secure channel is not available. In this paper, an information hiding method into coded images. A bit plan applied on the secret image and reshaped into the vector, while an Absolute Moment Block Truncation Coding (AMBTC) is used for coding the color images (cover images). The coding is applied as a method for size reduction by representing each block with two means called Mb, Ma and Bit-map that represent the location of them in binary form. The embedding of secret information is represented by modifying the bit-map. Additional complexity is added to the method by permutation of the bit plane vector using a chaotic map called the logistic map. The proposed method was examined and proved its efficiency by obtaining a high PSNR and a low RMSE result, as well as, the visual testing through the histogram test of the image before and after embedding.[5]

Advantages

- The proposed method is based on the concept of reversible data hiding (RDH) which ensures that, the original image can be fully restored after the data hiding process.
- The method uses bit-plan permutation (BPP) to further improve the security of the embedded data.
- Absolute moment block truncation coding (AMBTC) is used to compress the image data and make more room for the embedded data.

- Experimental results showed that, the proposed method outperforms other RDH methods in terms of embedding capacity and the image quality.
- The proposed method is applied to various types of images including grayscale and color images.

Disadvantages

- The proposed method may not be suitable for real-time applications as the embedding process may take more time.
- The security of the embedded data may still be vulnerable to attacks such as: brute-force attacks.
- The method may not be efficient for embedding large amounts of data in high-resolution images.
- The paper lacks a detailed comparison with state-of-the-art methods.
- The proposed method requires the original image to be available for extraction of the embedded data, which may not be practical in some scenarios.

The following work, available in the literature was also found to be important and useful for our present study.

6. K. Tiwari.; S. J. Gangurde, "LSB Steganography Using Pixel Locator Sequence with AES," (**2021**) 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC).[6]

7. G. Suchi.; R. Manoj.; D. Deepika, (**2015**). Improved Detection of 1-2-4 LSB Steganography and RSA Cryptography in Color and Grayscale Images.[7]

8. M. Prasad.; K. L. Sudha, (**2011**). Chaos Image Encryption using Pixel Shuffling. Computer Science & Information Technology.[8]

9. M. Ramalingam, (**2011**). Stego machine-Video steganography using modified LSB algorithm.[9]

10. R. K. Basak.; K. Dasgupta.; P. Dutta, "Steganography in Grey Scale Animated GIF using Hash-based Pixel Value Differencing," (**2018**) Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN).[10]

CHAPTER 3

SYSTEM ARCHITECTURE

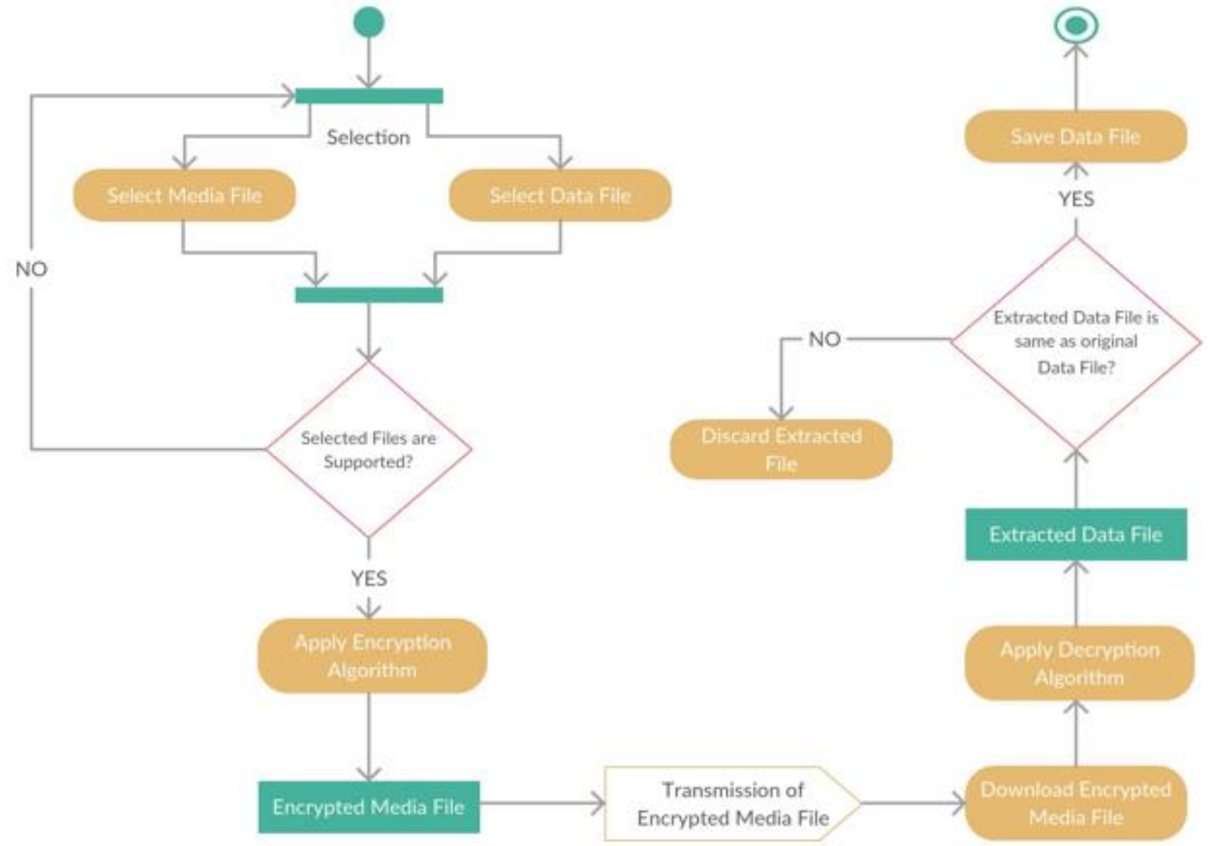

Selection
Select Media File
Select Data File
NO
Selected Files are Supported?
YES
Apply Encryption Algorithm
Encrypted Media File
Transmission of Encrypted Media File
Download Encrypted Media File
Apply Decryption Algorithm
Extracted Data File
Extracted Data File is same as original Data File?
NO
Discard Extracted File
YES
Save Data File

Fig. 3.1: LSB Technique

Fig. 3.1 shows the activity diagram for steganography using LSB method.

As can be seen, the LSB method is a steganography technique which is used to hide secret data within a digital image. The system architecture for the LSB method typically involves the following components:

- Input Image: This is the digital image that is used as the carrier for the secret data. The input image should be of a suitable size and format to accommodate the secret data.
- Secret Data: This is the data that needs to be hidden within the input image. The secret data can be any type of data, such as: a text, an image, an audio or a video.
- Steganography Algorithm: The steganography algorithm is used to embed the secret data within the input image. The LSB method involves replacing the least significant bit of each pixel in the input image with the secret data bits. This is done in a way that the changes to the image are not easily noticeable.
- Output Image: This is the resulting image after the secret data has been embedded using the steganography algorithm. The output image should appear identical to the input image to the naked eye, but it contains the secret data hidden within it.

- Extraction Algorithm: The extraction algorithm is used to extract the secret data from the output image. This involves analyzing the LSBs of each pixel in the output image to retrieve the secret data bits.
- Secret Data Output: This is the resulting secret data after it has been extracted from the output image using the extraction algorithm. The secret data can then be used for its intended purpose.

3.1 HARDWARE REQUIREMENTS

•	Processor	:	I5 8th Generation
•	Hard Disk	:	500 GB
•	Monitor	:	15.6 inch
•	RAM	:	8 GB
•	Mouse	:	Optical
•	Keyboard	:	Multimedia

3.2 SOFTWARE REQUIREMENTS

•	Operating System	:	Windows 10/11
•	Coding language	:	Python
•	IDE	:	Python 3.7 IDLE

CHAPTER 4

METHODOLOGY

Steganography is an art of hiding the secret messages within a medium such as: an image, an audio or a video, without any apparent suspicion of its existence. LSB (Least Significant Bit) method is used in steganography to embed secret data within the digital files. In this method, the LSB of a pixel or a sample in a file is altered to carry the hidden message. Image steganography involves embedding secret data within an image file without visibly altering the image. The LSB technique is also used to embed the secret data within the pixels of the image file. The color values of the image pixels are modified by replacing the LSB of each color component with the message bit. The human eye cannot perceive the difference between the original and the modified image. Audio steganography involves hiding secret messages within an audio file. LSB method can be used to embed the secret message within the audio samples. The least significant bit of each sample is replaced with the message bit to carry the secret message. Audio steganography is a challenging task as the human ear is sensitive to changes in sound quality. Video steganography is another technique of hiding secret messages within a video file. LSB method can be used to embed the secret message within the frames of the video file. The LSB of each color component of each pixel in the frames is replaced with the message bit. The video frames can be modified without affecting the visual quality of the video.

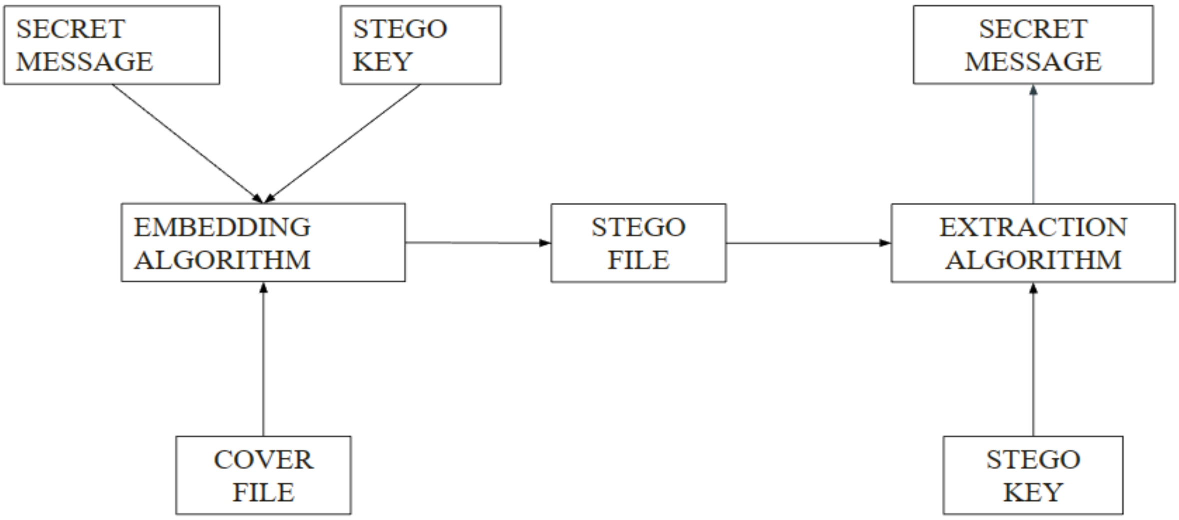


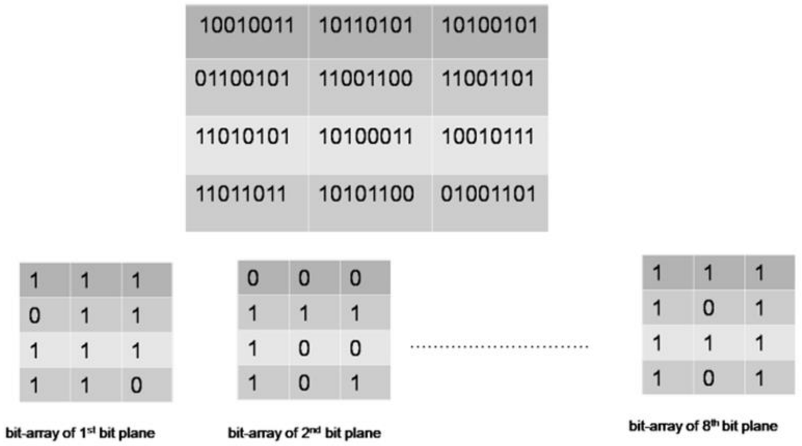
Fig. 4.1: Flowchart

Fig. 4.1 shows the flowchart of the Steganography

4.1 LSB TECHNIQUE

In LSB technique, the video file is converted into frames. In a small video such as: AVI., MPEG., or MP4, 20–25 frames can be generated per second. A frame in the video is actually an image which consists of a collection of pixel values (intensity and color) in a list formation or in a matrix form. A 24-bitmap RGB image will have 24-bits values for each pixel, 8-bits for every 3 color channels. RGB is highly suitable as there are a lots of information where we can hide secret messages, with one-bit change for every byte. Each component is of one byte *i.e.*, 8-bits in which the first one is the most significant bit.

LSB technique is used for hiding the secret information resulting in a change in the last bit of each byte of the component. Substitution of the LSB results in the human imperceptibility. It is known that, the first and most important requirement of the steganography is invisibility of changes in the cover frames. Steganography strength lies in its ability to be unnoticed by the human eye.



10010011	10110101	10100101
01100101	11001100	11001101
11010101	10100011	10010111
11011011	10101100	01001101

1	1	1
0	1	1
1	1	1
1	1	0

bit-array of 1st bit plane

0	0	0
1	1	1
1	0	0
1	0	1

bit-array of 2nd bit plane

....

1	1	1
1	0	1
1	1	1
1	0	1

bit-array of 8th bit plane

Fig. 4.2: Bit Array

In the **Fig. 4.2** the bit-array of 1st to 8th bit is shown.

For hiding three bits of the data in every pixel's color, a 24-bit image is used. Human eye, generally, cannot easily differentiate between the 21-bit color and the 24-bit color.

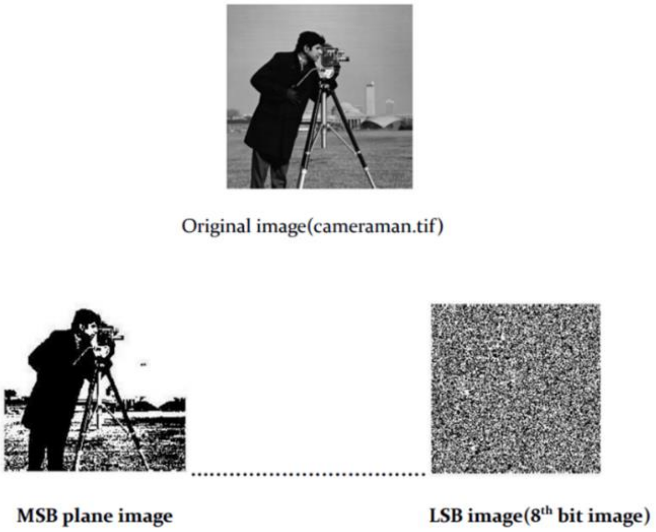


Fig. 4.3: MSB and LSB of Image

Fig. 4.3 shows the difference between the LSB bits and MSB bits of an image

4.2 EMBEDDING

A simple LSB insertion process is used for embedding the image in a cover video file frame. We replaced the least bits of each 24-bit pixel of the selected target frames by each bit plane image generated by the bit slicing method. Here, we embedded every (8-i)th bit plane image into the ith frame of the randomly generated sequence.

4.3 EXTRACTION

In this section we are going to retrieve the hidden image from the embedded video which is called stego-video. We used a very simple extraction algorithm to extract the secret image from the stego-video.

Step 1: First of all, we extracted all the frames from the video.

Step 2: We selected the stego frames using the above-mentioned frame selection method.

Step 3: As the secret image is embedded into the least significant bits of every stego-frames, we simply did the logical and operation between 1 and every byte of the carrier frames to get the bit-array of LSBs which is actually the bit-array of one of the bit-plane images.

At this stage we have 8 bit-plane images of the original image. To recover the original image those 8 bit-plane images were merged according to the order by which they are inserted into the cover frame sequences.

4.4 OBJECT ORIENTED DESIGN

4.4.1 CLASS DIAGRAM

In software engineering, a class diagram is the Unified Modelling Language (UML), which is a kind of static structure diagram that describes the structure of a system by showing the system's classes, and their attributes, operations and the relationship among the objects. The class diagram is a main building block of the object-oriented modelling. It is used for general conceptual modelling of the structure of the application and for the detailed modelling and for translating the models into programming code.

4.4.2 DATA FLOW DIAGRAM

A dataflow outline is a tool for referring to the knowledge progression from one module to the next module as shown in the **Figures 4.4** and **4.5**. This map gives the data of each module's info. The map has no power flow and there are no circles at the same time.

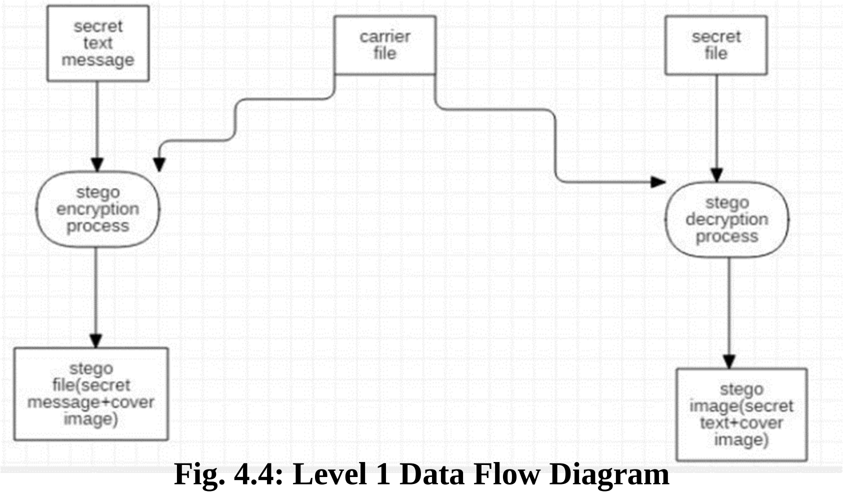


Fig. 4.4: Level 1 Data Flow Diagram

Fig. 4.4 shows the level 1 data flow diagram.

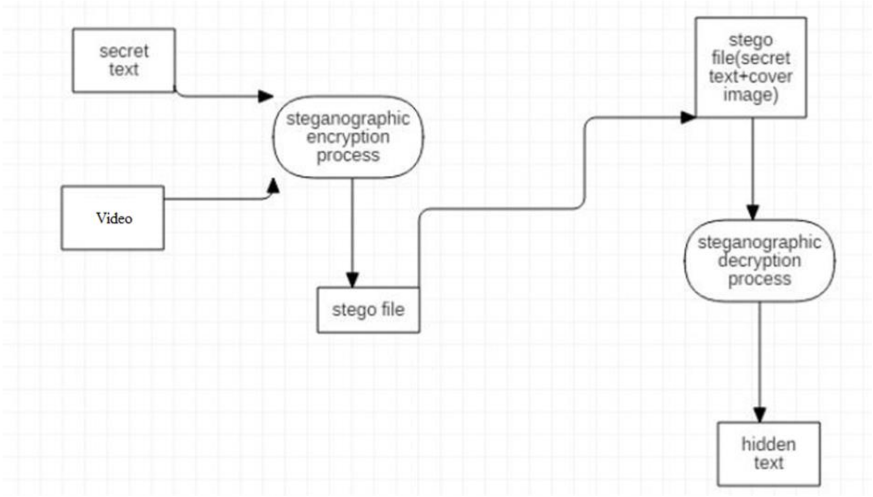

secret text
Video
steganographic encryption process
stego file
stego file(secret text+cover image)
steganographic decryption process
hidden text

Fig. 4.5: Level 2 Data Flow Diagram

Fig. 4.5 shows the knowledge progression from one module to another. The map has no power flow and there are no circles at the same time.

4.4.3 USE CASE DIAGRAM

A Use Case Diagram (UCD) gives a lot of situations that reflect a client-frame relationship. A use case chart shows the entertainer-to-use relationship. Usage cases and on-screen characters are the two most important elements of a usage case diagram. An on-screen character in UCD refers to a user or other person connected with the demonstration process. A use case chart presented in the **Fig. 4.6** is an out of the box perspective which speaks to some activity; each module will perform to complete an errand. A use case is a methodology used in the system analysis to identify, clarify and to organize the system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modelling Language), a standard notation for the modelling of real-world objects and systems. There are a number of benefits having a use case diagram over similar diagrams such as flowcharts.

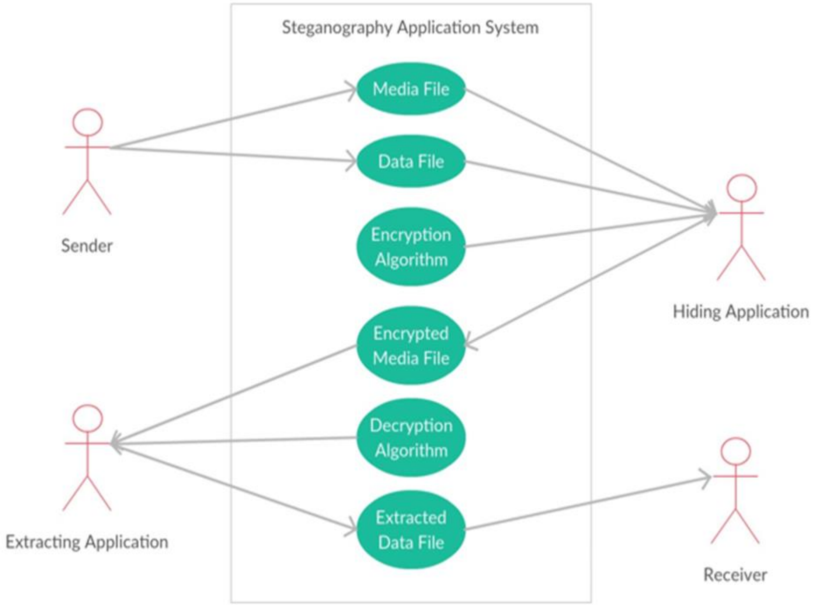


Fig. 4.6: Use case diagram

Fig. 4.6 shows a use case chart in an out of-the box perspective which speaks to some activity. Each module will perform to complete an errand. The module depicts the sender, extracting application, hiding application and the receiver. The sender sends the media and the data file to hiding application and the hiding application encrypts into the encrypted media. The extracting application uses decryption algorithm and gets the extracted data file which is sent to the receiver.

4.4.4 SEQUENCE DIAGRAM

A sequence diagram as shown in the **Fig. 4.7** is a type of interaction diagram because it describes how and in what order a group of objects work together. These diagrams are used by the software developers and the business professionals to understand the requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. A usage scenario is a diagram of how a system could potentially be used. It is a great way to make sure that, we have worked through the logic of every usage scenario for the system. If we consider a service to be a high-level method used by different clients, a sequence diagram is an ideal way to map that out.

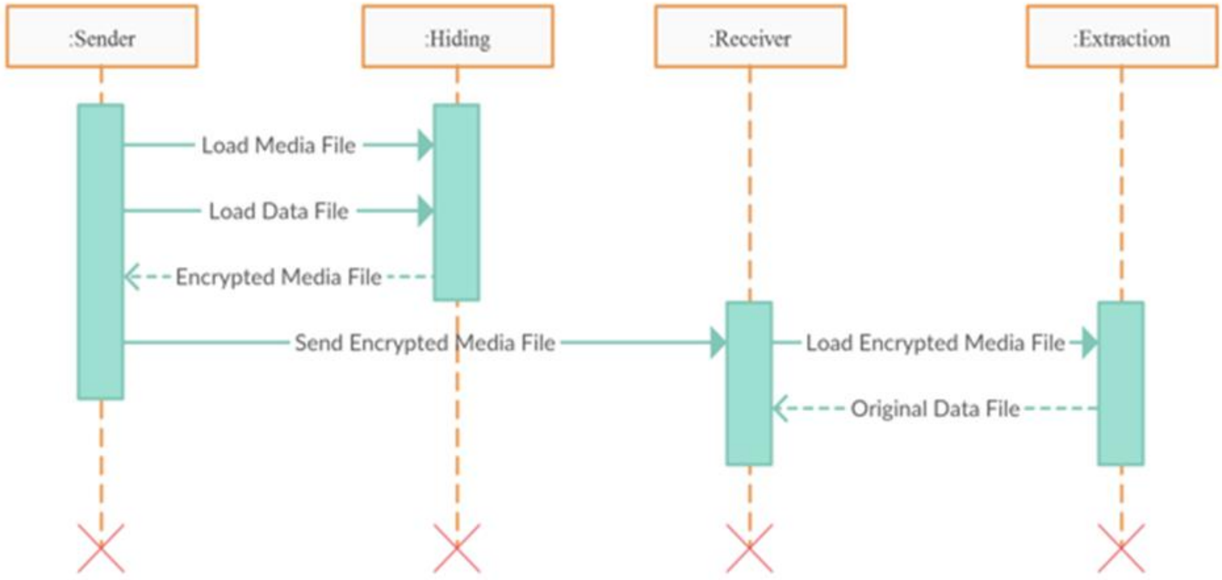


Fig. 4.7: Sequence Diagram

The above Fig. 4.7 shows a usage scenario is a diagram of how your system could potentially be used.

CHAPTER 5

TESTING

5.1 SELECTION

Test Case 1	ST-1
Name of Test	Selection
Items being tested	mp4, wav, jpg
Sample Input	Select a video file, an audio file, and an image file to use for testing
Expected Output	video file, an audio file, and an image file selected
Actual Output	No errors while selecting
Remarks	Passed

Table 5.1: Selection

Table 5.1 shows the testing of selection of images the different types of files for image, audio and video steganography.

5.2 GENERATION

Test Case 2	ST-2
Name of Test	Generation
Items being tested	Secret Message
Sample Input	Generate a secret message of random characters or text
Expected Output	Characters generated
Actual Output	No errors while generating
Remarks	Passed

Table 5.2: Generation

Table 5.2 shows the testing for the generation of secret message of random characters or text.

5.3 LSB IMPLEMETATION

Test Case 3	ST-3
Name of Test	LSB Implementation
Items being tested	Changing of LSB
Sample Input	Embed the secret message within the least significant bits of each file.
Expected Output	Least Significant Bits Changed
Actual Output	No errors while changing
Remarks	Passed

Table 5.3: LSB Implementation

Table 5.3 shows the testing of LSB implementation being done without any errors.

5.4 SAVING

Test Case 4	ST-4
Name of Test	Saving
Items being tested	Modified Files
Sample Input	Save the modified files as new files with different names
Expected Output	Modified Files saved
Actual Output	No errors while saving
Remarks	Passed

Table 5.4: Saving

Table 5.4 shows the testing of the files being properly saved without any errors.

5.5 EXTRACTION

Test Case 5	ST-5
Name of Test	Extraction
Items being tested	Secret Message
Sample Input	extract the hidden message from each modified file
Expected Output	Secret Message Extracted
Actual Output	No errors while Extraction
Remarks	Passed

Table 5.5: Extraction

Table 5.5 shows the testing for secret message being extracted.

5.6 VERIFICATION

Test Case 6	ST-6
Name of Test	Verification
Items being tested	Verify Message
Sample Input	Verify that the extracted message is the same as the original message
Expected Output	Secret Message Verified
Actual Output	No errors while verifying
Remarks	Passed

Table 5.6: Verification

Table 5.6 shows the testing for the message received being the original message which is a success.

CHAPTER 6

RESULTS

6.1 START PAGE

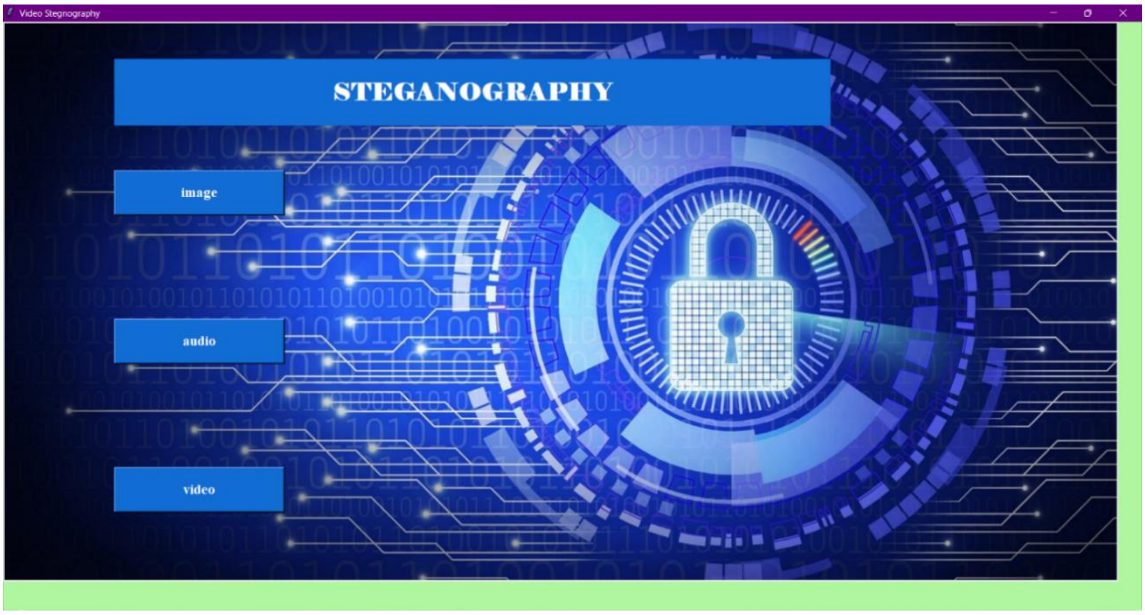


Fig 6.1: Start Page

Fig 6.1 shows the starting page which displays three buttons image, audio and video.

6.2 IMAGE STEGANOGRAPHY

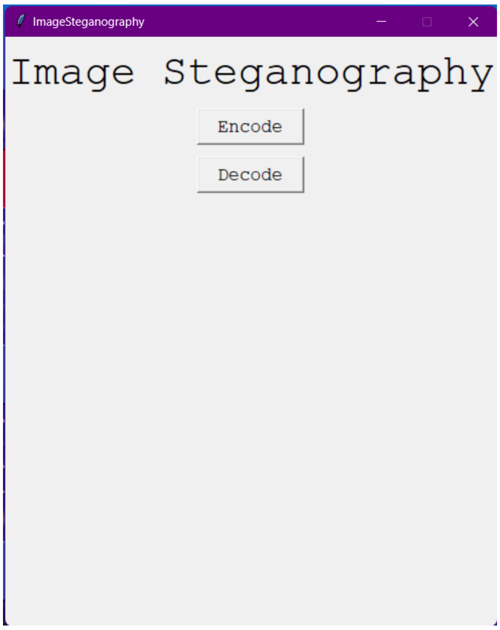
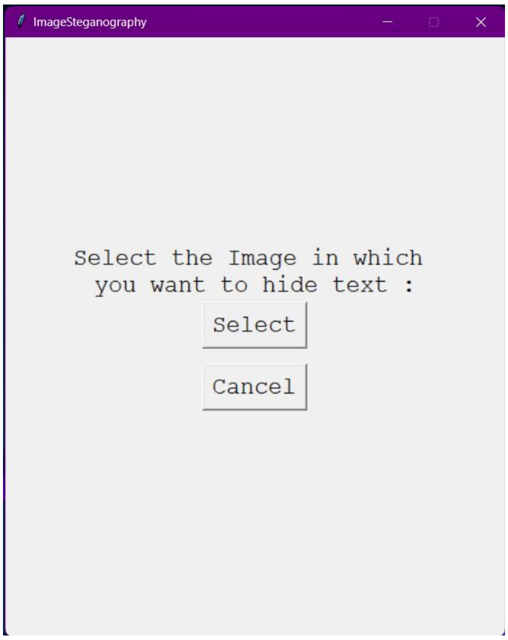


Fig 6.2: Image Steganography

Fig 6.2 shows the button to select encode and decode.

6.3 IMAGE SELECTION

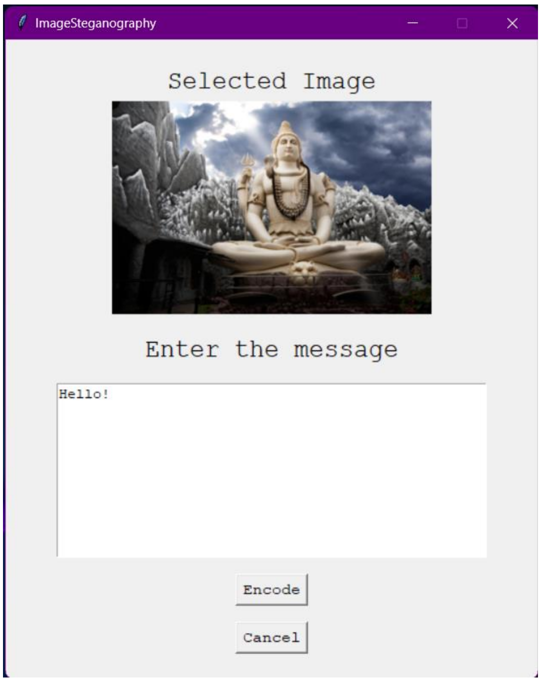


Select the Image in which you want to hide text :

Fig 6.3: Image Selection

Fig 6.3 shows the windows to select the image.

6.4 SECRET MESSAGE EMBEDDING

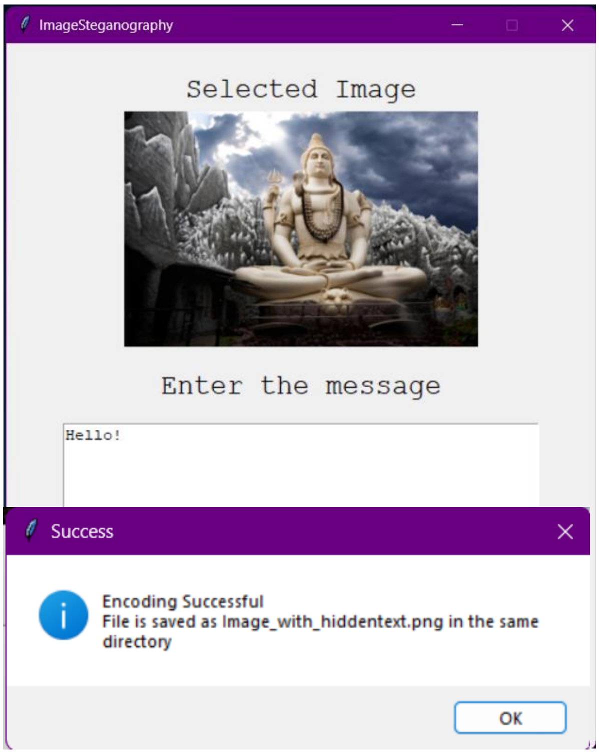


Hello!

Fig 6.4: Secret Message Embedding

Fig 6.4 shows the secret message being written in the text box field to be embedded in the image.

6.5 IMAGE SAVED



Hello!

Success

Encoding Successful
File is saved as Image_with_hiddentext.png in the same directory

Fig 6.5: Image Saved

Fig 6.5 shows the message that the text has been successfully encoded in the image.

6.6 EMBEDDED IMAGE SELECTION

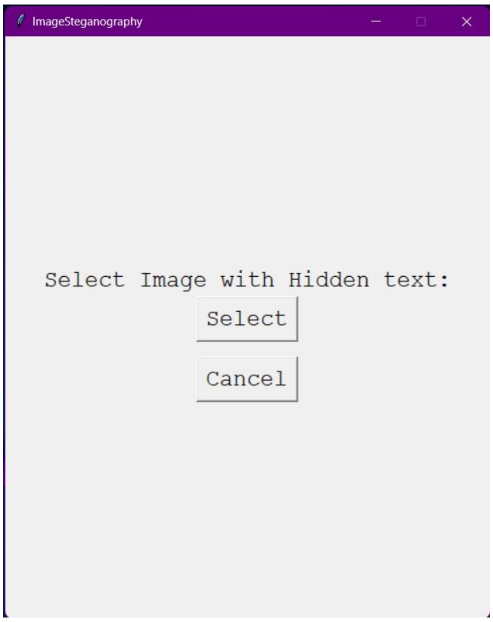
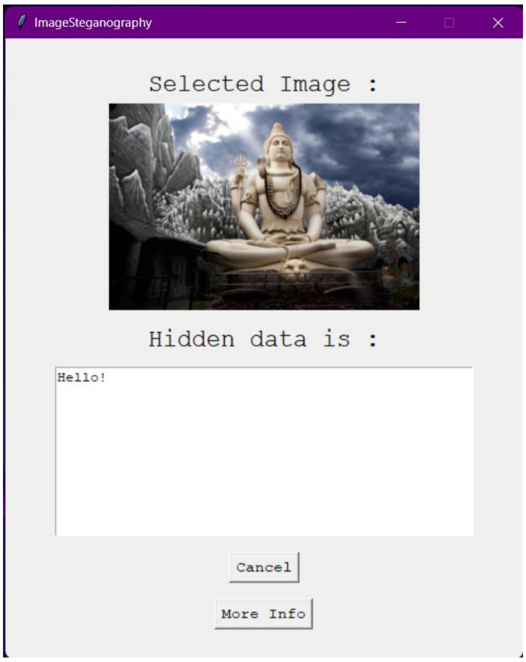


Fig 6.6: Embedded Image Selection

Fig 6.6 shows the window to select the embedded image to extract the text.

6.7 SECRET MESSAGE EXTRACTION

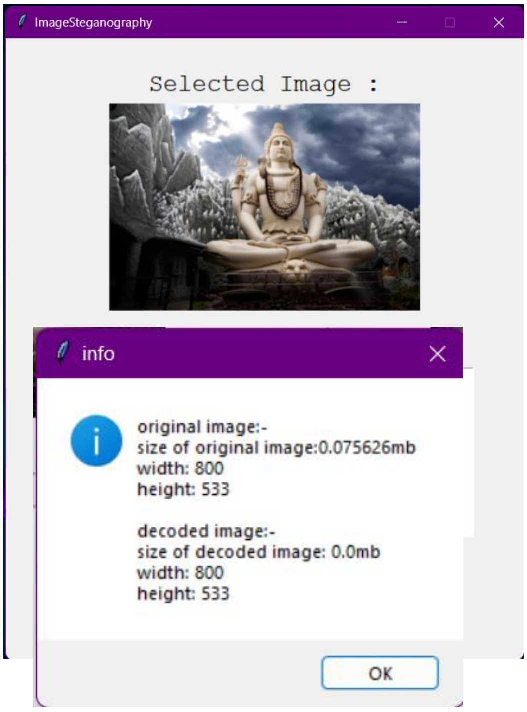


Hello!

Fig 6.7: Secret Message Extraction

Fig 6.7 shows the original message after extraction.

6.8 COMPARISION



original image:-
size of original image:0.075626mb
width: 800
height: 533

decoded image:-
size of decoded image: 0.0mb
width: 800
height: 533

Fig 6.8: Comparison

Fig 6.8 shows the comparison between the original and the decoded image.

6.9 AUDIO STEGANOGRAPHY

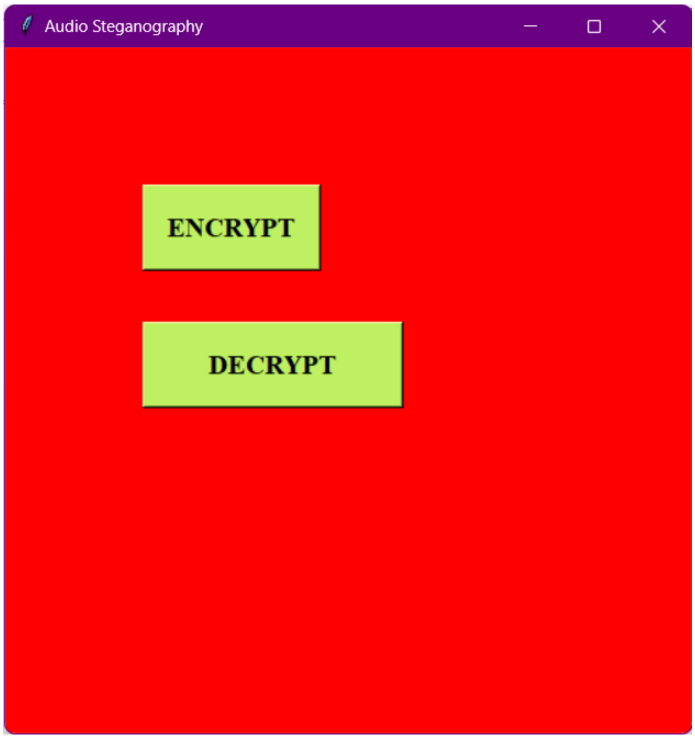
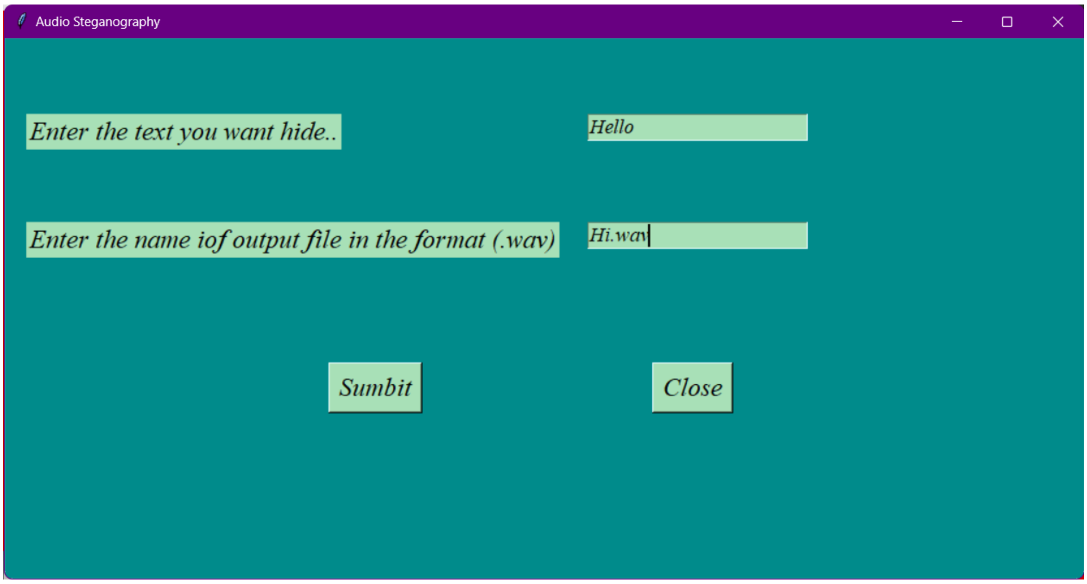


Fig 6.9: Audio Steganography

Fig 6.9 shows the starting window for Audio Steganography.

6.10 AUDIO FILE



Enter the text you want hide.. *Hello*

Enter the name iof output file in the format (.wav) *Hi.wav*

Fig 6.10: Audio File

Fig 6.10 shows the window to embed the secret message and to name the file being embedded.

6.11 EMBEDDED AUDIO FILE

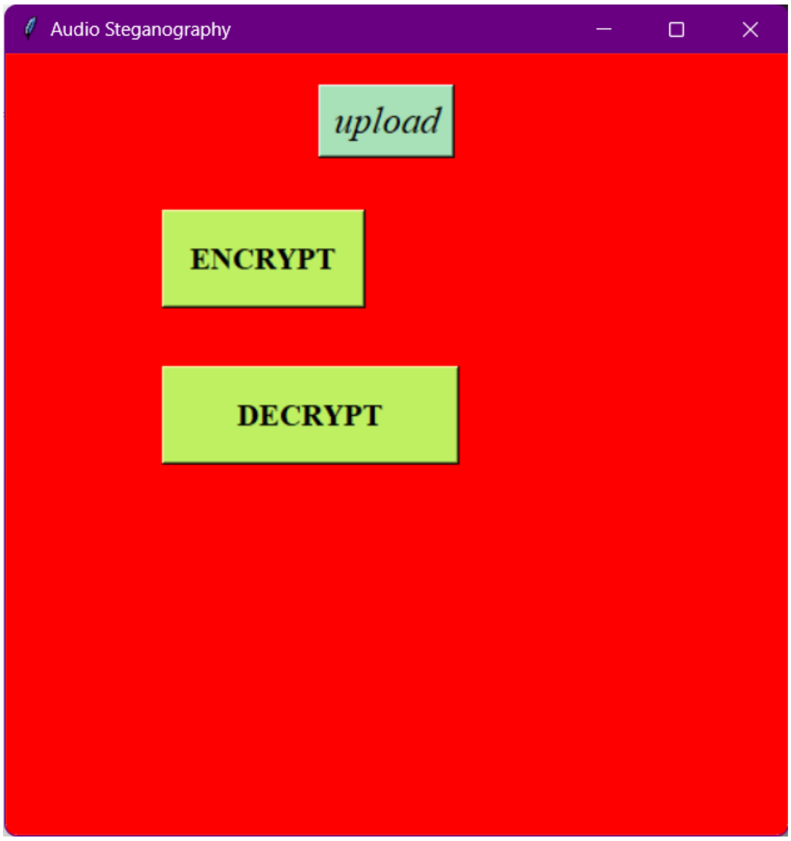
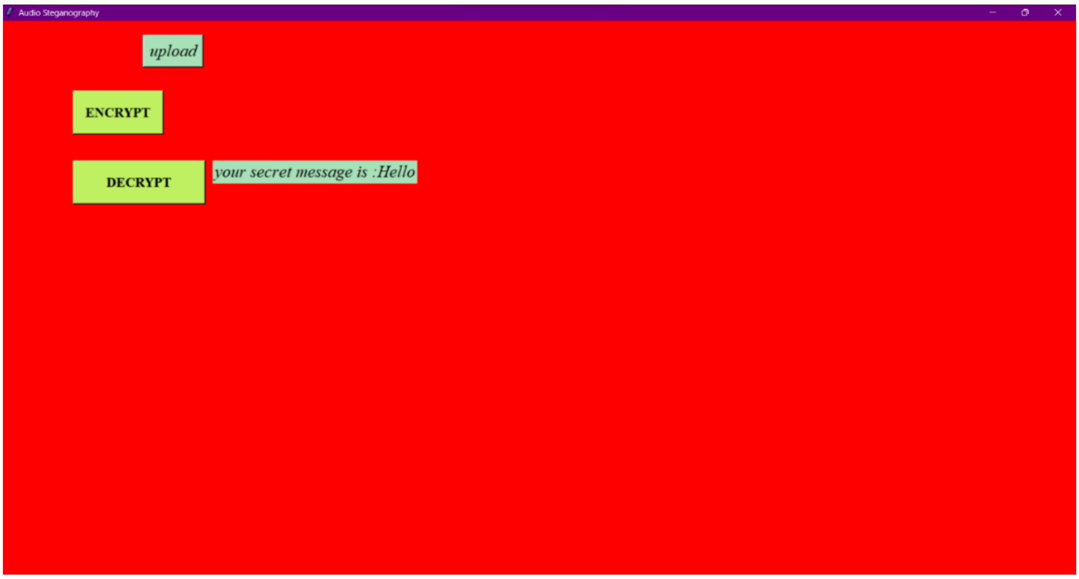


Fig 6.11: Embedded Audio File

Fig 6.11 shows the window to upload the embedded image.

6.12 SECRET MESSAGE IN AUDIO FILE



your secret message is :Hello

Fig 6.12: Secret Message in Audio File

Fig 6.12 shows the extracted message from the embedded image.

6.13 VIDEO STEGANOGRAPHY

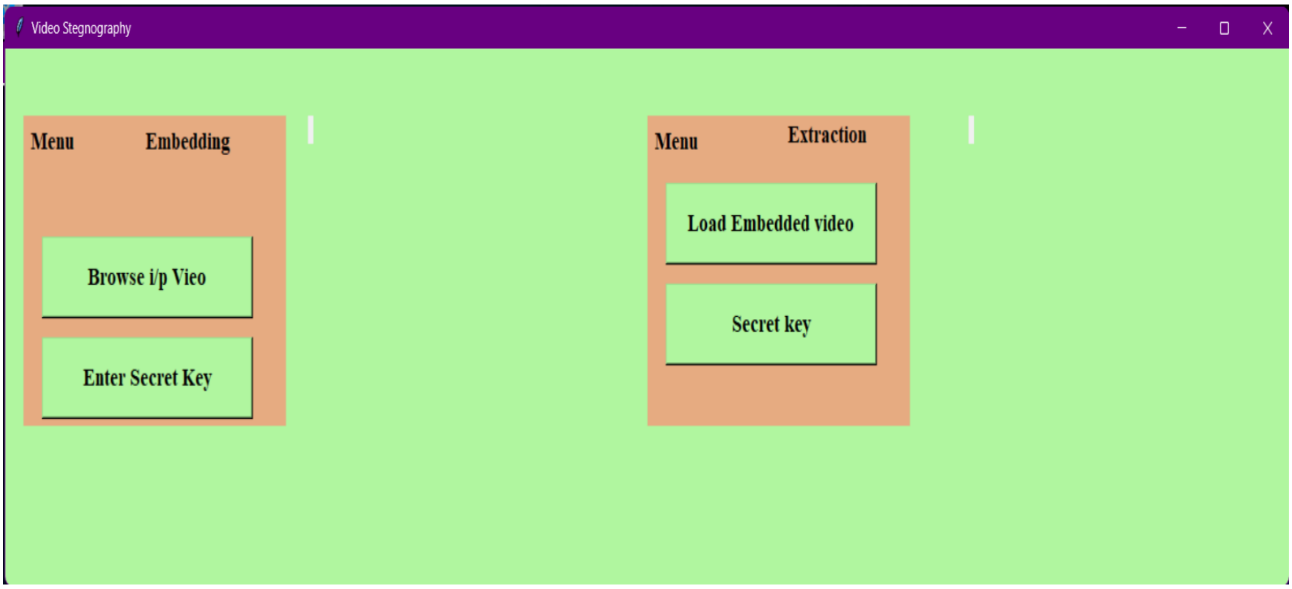


Fig 6.13: Video Steganography

Fig 6.13 shows the start window of Video Steganography.

6.14 VIDEO SELECTION

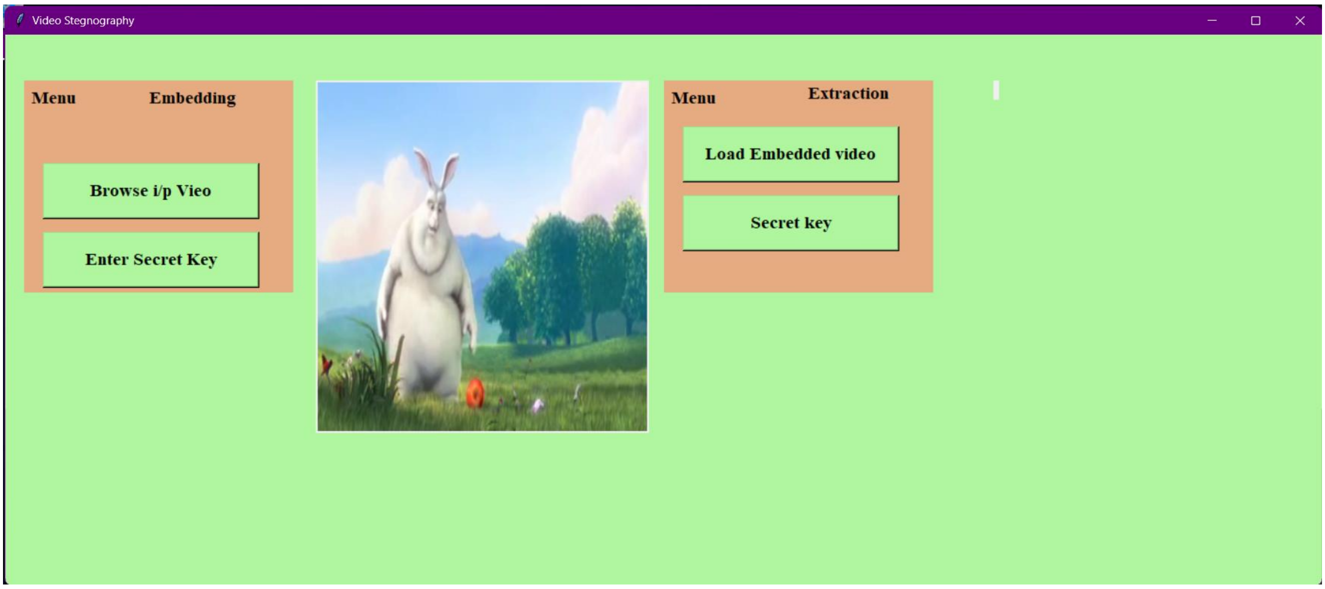
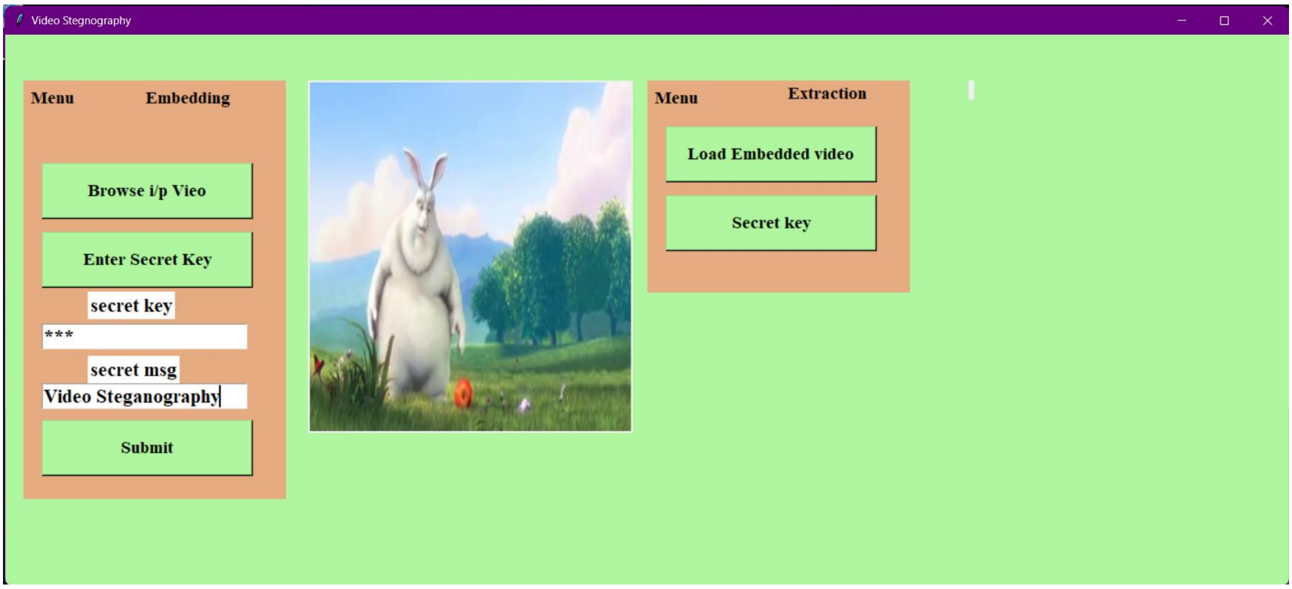


Fig 6.14: Video Selection

Fig 6.14 shows selected video being played.

6.15 SECRET KEY AND MESSAGE



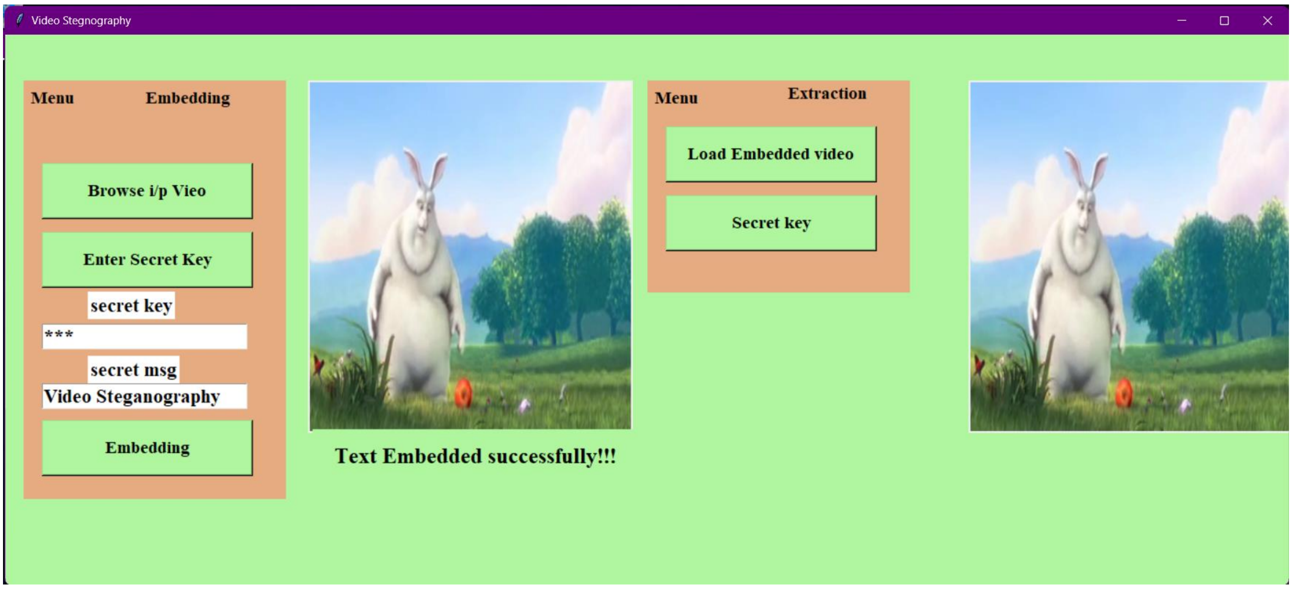
secret key
***

secret msg
Video Steganography

Fig 6.15: Secret key and message

Fig 6.15 shows the secret message and secret key being entered to be embedded into the video.

6.16 EMBEDDING SUCCESSFUL



secret key
***

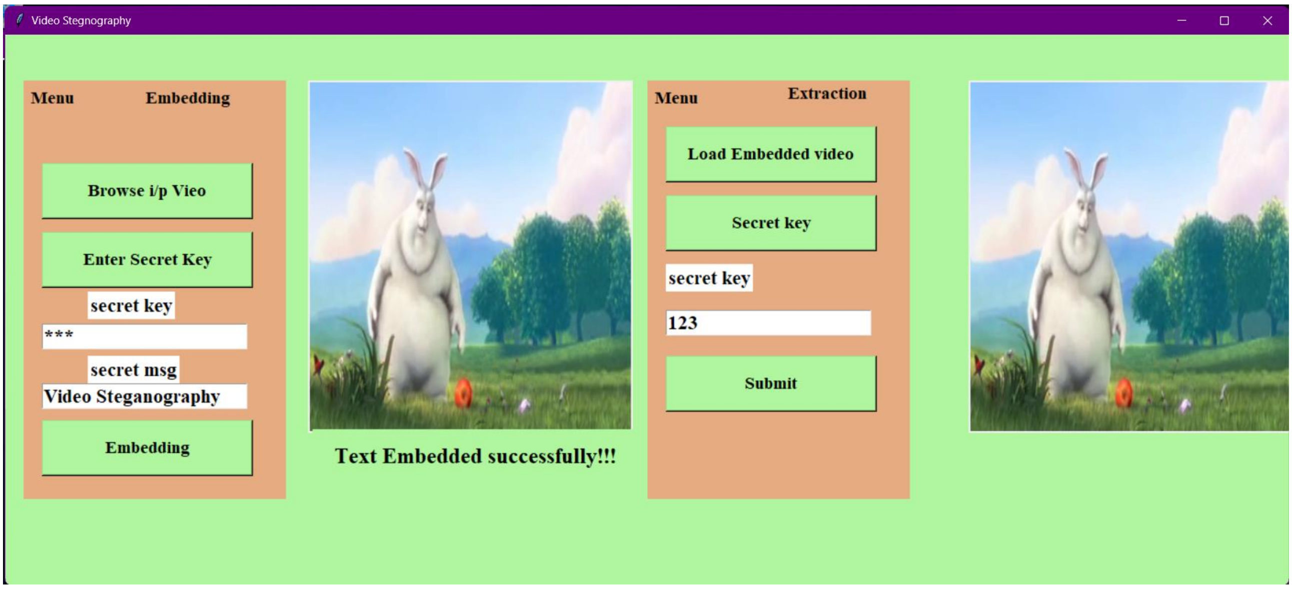
secret msg
Video Steganography

Text Embedded successfully!!!

Fig 6.16: Embedding successful

Fig 6.16 shows the message embedding successful after the message has been embedded into the image.

6.17 EMBEDDED MESSAGE UPLOAD



secret key: ***

secret msg: Video Steganography

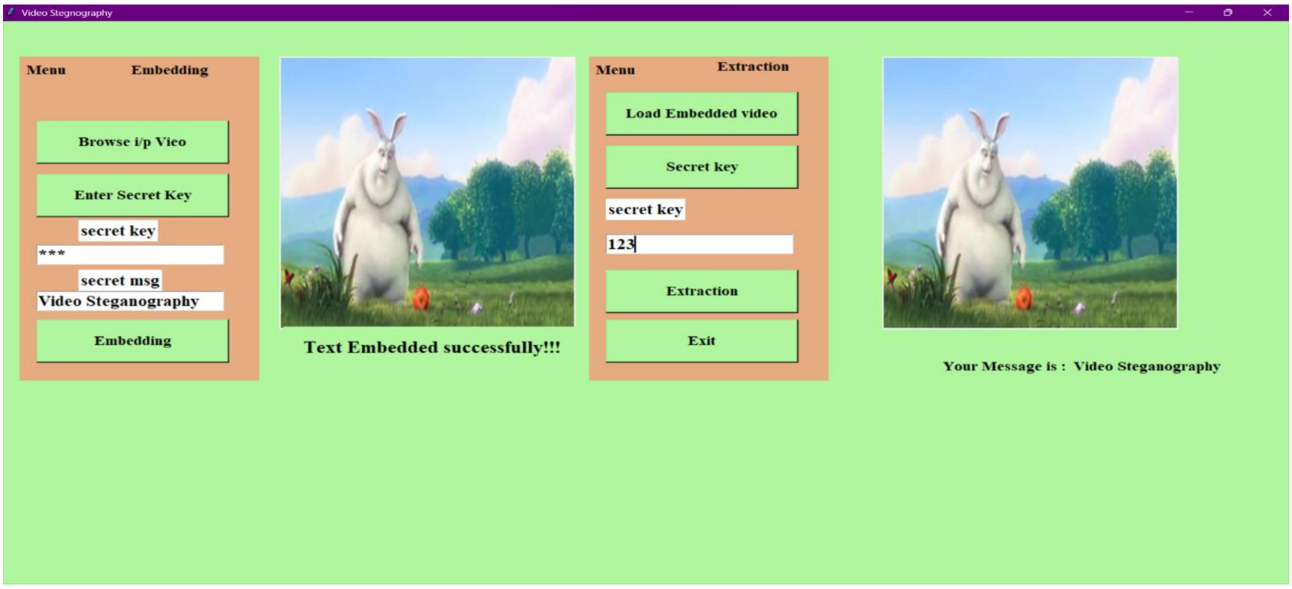
Text Embedded successfully!!!

secret key: 123

Fig 6.17: Embedded message upload

Fig 6.17 shows the embedded video being played in the window.

6.18 SECRET MESSAGE DECODED



secret key: ***

secret msg: Video Steganography

Text Embedded successfully!!!

secret key: 123

Your Message is : Video Steganography

Fig 6.18: Secret message decoded

Fig 6.18 shows the secret message after it has been decoded which matched the original message.

CONCLUSIONS

In conclusion, steganography using the Least Significant Bit (LSB) method has been used for hiding the secret messages within digital media files such as images, audio and video files. The LSB method involved replacing the least significant bit of each pixel or sample value in the cover file with a bit of the secret message. It was ascertained by us that, one of the benefits of using the LSB method for steganography is relatively easy to implement, and it allows for a lar amount of data to be hidden within a cover file without significantly altering its quality. Additionally, it was found that, the LSB method can be used successfully for different types of digital media files including: images, audio and video files. The present project work has been published in the form of an article [11].

REFERENCES

[1]. K. Rajalakshmi.; K. Mahesh, "Video steganography based on embedding the video using PCF technique," 2017 International Conference on Information Communication and Embedded Systems (ICICES), **2017**, pp. 1–4. Doi: 10.1109/ICICES.2017.8070726.

[2]. Pande Gede Pradnya Jaya, S.T.; B. Hidayat.; F. Y. Suratman, "Enhanced LSB Steganography with people detection as stego key generator," 2017 International Conference on Signals and Systems (ICSigSys), **2017**, pp. 99–104. Doi: 10.1109/ICSIGSYS.2017.7967078.

[3]. I. Bajaj.; R. K. Aggarwal, "RSA Secured Web Based Steganography Employing HTML Space Codes and Compression Technique," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), **2019**, pp. 865–868. Doi: 10.1109/ICCS45141.2019.9065640.

[4]. N. Adam.; M. Mashaly.; W. Alexan, "A 3DES Double-Layer Based Message Security Scheme," 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), **2019**, pp. 1–5. Doi: 10.1109/CAIS.2019.8769457.

[5]. S. F. Hussein.; A. H. Radie, "Reversible Data Hiding Based Bit-Plan Permutation and Absolute Moment Block Truncation Coding (AMBTC)," 2020 3rd International Conference on Engineering Technology and its Applications (IICETA), **2020**, pp. 24–28. Doi: 10.1109/IICETA50496.2020.9318861.

[6]. K. Tiwari.; S. J. Gangurde, "LSB Steganography Using Pixel Locator Sequence with AES," 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC), **2021**, pp. 302–307. Doi: 10.1109/ICSCCC51823.2021.9478162.

[7]. G. Suchi.; R. Manoj.; D. Deepika, (**2015**). Improved Detection of 1-2-4 LSB Steganography and RSA Cryptography in Color and Grayscale Images. 1120–1124. 10.1109/CICN.2015.220.

[8]. M. Prasad.; K. L. Sudha, (**2011**). Chaos Image Encryption using Pixel Shuffling. Computer Science & Information Technology. Doi: 10.5121/csit.2011.1217.

[9]. M. Ramalingam, (**2011**). Stego machine-video steganography using modified LSB algorithm. Doi: 10.5281/zenodo.1070342

[10]. R. K. Basak.; K. Dasgupta.; P. Dutta, "Steganography in Grey Scale Animated GIF using Hash-based Pixel Value Differencing," 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), **2018**, pp. 248–252. Doi: 10.1109/ICRCICN.2018.8718708.

[11]. C. Rohini.; A. Srikanth.; R. Prajwal Praneeth Kumar.; R. H. Basavaraj.; U. Vinay, “Advanced Data Security Using Modulo Operator And LSB Method”, J. Schol. Engg. Sci. & Manag., **2023**, 2 (5), pp. 26–37. Doi:10.5281/zenodo.7890771.