

SECURED PALETTE

Safeguarding Secrets with LSB Steganography and Visual Cryptography

Enrollment numbers	21103042	21103150	21103160
Names of students	Astha Raghuwanshi	Jahanvi Gupta	Ragini Mittal
Name of supervisor	Ms. Mradula Sharma		



December - 2023

Submitted in the partial fulfillment of the Degree of

Bachelor of Technology

in

Computer Science Engineering

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION
TECHNOLOGY**

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY , NOIDA

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Date			
Place	Jaypee Institute of Information Technology, Sector 62, Noida		
Name of supervisor	Ms. Mradula Sharma		
Signature of supervisor			
Enrollment numbers	21103042	21103150	21103160
Names of students	Astha Raghuwanshi	Jahanvi Gupta	Ragini Mittal
Signature of Students			

CERTIFICATE

This is to certify that the work titled **Secured Palette** submitted by our group in partial fulfillment for the **B.Tech** Computer Science of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of supervisor	
Name of supervisor	Ms. Mradula Sharma
Designation	
Date	

ACKNOWLEDGEMENT

Our team members would sincerely like to thank our college Jaypee Institute Of Information Technology, Sector 62 Noida that gave us the opportunity to work upon a project under the guidance of **Dr. Ankit Vidyarthi** sir.

We would like to express our sincere thanks and gratitude to our mentor **Dr. Ankit Vidyarthi** for letting us work on this project. We are grateful to him for his support and guidance in completing this project. We are very grateful to our supervisor **Ms. Mradula Sharma** for her willingness to give us valuable advice and direction whenever we approached her with a problem. We are also grateful to our mentor **Cyrus Thapa** for his constant mentorship and support throughout the course of the project.

Your valuable guidance and support helped us in various phases of completion of this project. We will always be thankful to you in this regard. We hope we will achieve more in our future endeavors.

Enrollment numbers	21103042	21103150	21103160
Names of students	Astha Raghuwanshi	Jahanvi Gupta	Ragini Mittal
Signature of Students			
Date			

SUMMARY

Secured Palette is a web application developed using Streamlit in Python, aiming to enhance information security through a combination of LSB (Least Significant Bit) steganography and visual cryptography techniques. In the encoding phase, the application takes an input image and text to be concealed. LSB steganography involves subtly adjusting the least significant bits of colored pixels in the image, embedding the text message within the image without perceptible alterations.

Following the steganographic process, visual cryptography is employed. The encoded image is divided into two shares using visual cryptography. Each share independently reveals no information about the original image or message, but when superimposed, the visual cryptography mechanism reconstructs the original image, making it visually perceptible. This process ensures that neither share alone discloses any sensitive information.

For decoding, the web application prompts the user to upload the shares of the visual cryptography scheme. Upon overlap, the shares reconstruct the hidden image. Internally, LSB steganography is applied again to extract the concealed text message from the reconstructed image. This dual-layered security approach combines the obscurity of LSB steganography with the cryptographic resilience of visual cryptography, providing a robust means of safeguarding secrets. The innovative fusion of these techniques fortifies data protection by making it challenging for unauthorized entities to discern both the existence of concealed information and its content.

Name of supervisor	Ms. Mradula Sharma		
Signature of supervisor			
Enrollment numbers	21103042	21103150	21103160
Names of students	Astha Raghuwanshi	Jahanvi Gupta	Ragini Mittal
Signature of Students			
Date			

TABLE OF CONTENTS

<u>Chapter No.</u>	<u>Topics</u>	<u>Page No.</u>
Chapter-1	Introduction 1.1 General Introduction 1.2 Problem Statement 1.3 Significance of the problem 1.4 Brief description of solution approach 1.5 Comparison of existing approaches to the problem framed	8 - 19
Chapter-2	Literature Survey 2.1 Literature review 2.2 Integrated summary of the literature studied	20- 22
Chapter-3	Requirement Analysis and Solution Approach 3.1 Overall description of the project 3.2 Requirement Analysis 3.3 Solution Approach	23 - 27
Chapter-4	Modeling and Implementation details 4.1 Design diagrams 4.1.1 Use Case diagram 4.1.2 Control Flow Diagrams 4.1.3 Class Diagram 4.1.4 Activity Diagram 4.2 Implementation details and issues 4.3 Risk Analysis and Mitigation	28 - 38
Chapter-5	Testing 5.1 Testing Plan 5.2 Component decomposition and type of testing required 5.3 List of test cases	39 - 46

5.4 Error and Exception Handling

5.5 Limitations of the solution

Chapter- 6

Findings, Conclusion, and Future Work

47 - 49

6.1 Findings

6.2 Conclusion

6.3 Future Work

References

50

CHAPTER-1

INTRODUCTION

1.1 GENERAL INTRODUCTION

In our increasingly interconnected digital landscape, the need for cryptographic techniques has become paramount. Cryptography serves as the safeguard for sensitive information in the face of evolving cybersecurity threats. As data transmission and storage have become integral aspects of modern communication, the vulnerability of information to unauthorized access and manipulation has grown exponentially. Cryptographic techniques provide a robust defense mechanism by transforming data into an unintelligible format, ensuring that even if intercepted, the content remains confidential. This heightened security extends beyond simple confidentiality; cryptography also verifies the integrity of data, confirming that it has not been altered during transmission. Moreover, cryptographic protocols facilitate secure authentication, allowing parties to establish trust and verify identities in digital interactions. Whether applied to secure financial transactions, protect personal information, or fortify communication channels, cryptographic techniques form an essential toolkit for upholding privacy, integrity, and authenticity in the dynamic and interconnected digital landscape.

What is Visual Cryptography?

Visual cryptography is a cryptographic technique which allows visual information (pictures, text, etc.) to be encrypted in such a way that the decrypted information appears as a visual image. One of the best-known techniques has been credited to Moni Naor and Adi Shamir, who developed it in 1994[7]. They demonstrated a visual secret sharing scheme, where an image was broken up into n shares so that only someone with all n shares could decrypt the image, while any $n - 1$ shares revealed no information about the original image. Each share was printed on a separate transparency, and decryption was performed by overlaying the shares. When all n shares were overlaid, the original image would appear.

In this example, the image has been split into two component images. Each component image has a *pair* of pixels for every pixel in the original image. These pixel pairs are shaded black or white according to the following rule: if the original image pixel was black, the pixel pairs in the component images must be complementary; randomly shade one ■□, and the other □■. When these complementary pairs are overlapped, they will appear dark gray. On the other hand, if the original image pixel was white, the pixel pairs in the component images must match: both ■□ or both □■. When these matching pairs are overlapped, they will appear light gray.

So, when the two component images are superimposed, the original image appears. However, without the other component, a component image reveals no information about the original image; it is indistinguishable from a random pattern of ■□ / □■ pairs. Moreover, if you have one component image, you can use the shading rules above to produce a *counterfeit* component image that combines with it to produce any image at all.

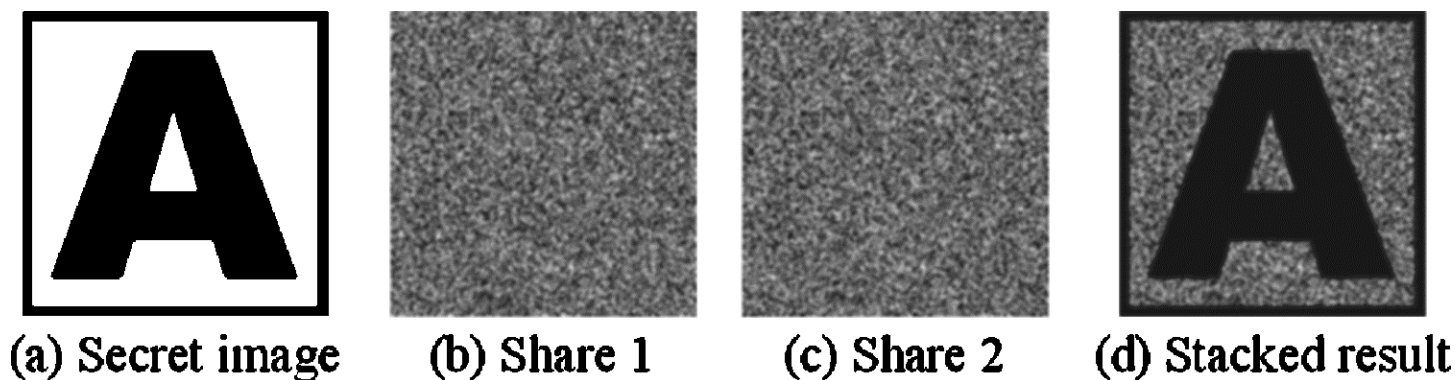


figure 1.1: Visual Cryptography example

Types of Visual Cryptography:

Threshold Visual Cryptography:

Definition: Threshold Visual Cryptography involves dividing the original image into shares, and a specific number of these shares (the threshold) are required to unveil the concealed information.

Key Feature: It enforces a minimum requirement for the number of shares needed to reveal the hidden content. If the threshold is not met, the secret remains secure.

Application: Often employed in scenarios where a predetermined level of authorization is necessary for accessing sensitive visual information.

General Visual Cryptography:

Definition: General Visual Cryptography is more flexible, allowing the revelation of the hidden content by stacking any combination of shares, without adhering to a specific threshold.

Key Feature: Offers versatility in the combination of shares required for decryption, providing freedom in the arrangement of shares during the decoding process.

Application: Suitable for situations demanding adaptability in the utilization of shares, offering a more dynamic approach to visual cryptography.

Understanding (k, n) Visual Cryptography:

Definition:

In (k, n) Visual Cryptography, the original image undergoes a division into 'n' shares, and the uniqueness lies in the fact that any 'k' out of these 'n' shares are essential for reconstructing the original secret image. Each share on its own provides no information about the hidden content; it is only when a sufficient number of shares are combined that the visual information becomes perceptible.

Key Features:

Scalable Security: The (k, n) approach allows for scalability in security. By adjusting the values of 'k' and 'n,' the level of security can be tailored to meet specific requirements. For instance, higher values of 'k' can enhance security, making more shares necessary for reconstruction.

Applications:

Customizable Security: The (k, n) model is versatile and can be customized for a broad spectrum of scenarios, catering to diverse security needs. For example, in situations demanding higher security, a larger value of 'k' can be chosen to mandate collaboration for decryption.

Applications of Visual Cryptography:

Secure Image Sharing:

Description: Visual Cryptography ensures secure sharing of images among multiple parties.

Use Case: Ideal for confidential image transmission in scenarios such as secure messaging, medical image sharing, and private collaboration [8].

Visual Authentication:

Description: Visual Cryptography is employed for verifying the authenticity of visual content.

Use Case: Used in watermarking, digital signatures, and authentication processes to confirm the integrity and origin of images [8].

Privacy-Preserving Communication:

Description: Visual Cryptography protects the privacy of visual information during communication.

Use Case: Valuable in scenarios where privacy is paramount, such as transmitting sensitive documents or personal images securely [8].

Secure Multimedia Storage:

Description: Visual Cryptography enhances the security of multimedia storage.

Use Case: Useful for safeguarding images stored in databases, ensuring confidentiality and protection against unauthorized access [8].

Biometric Security:

Description: Visual Cryptography contributes to secure biometric data transmission.

Use Case: Applied in scenarios where biometric images, such as fingerprints or facial features, need secure sharing and storage [8].

Document Authentication:

Description: Visual Cryptography verifies the authenticity of printed or digital documents.

Use Case: Employed in secure document transmission, authentication stamps, and protection against document forgery [8].

Cryptography Education:

Description: Visual Cryptography serves as a teaching tool for understanding cryptographic concepts.

Use Case: Used in educational settings to illustrate cryptographic principles in a visually intuitive manner [8].

Dynamic Visual Cryptography:

Description: Visual Cryptography adapts to dynamic scenarios.

Use Case: Useful in scenarios where the number of participants or access levels may change, providing flexibility in sharing visual secrets [8].

Advanced Topics and Research Directions(Visual Cryptography)

Color Visual Cryptography for Hiding Color Secret Images:

Research explores extending visual cryptography to handle color images, broadening its applicability beyond black and white.

Visual Cryptography with Additional Security Features (Encryption, Watermarking):

Efforts focus on integrating encryption for added protection and watermarking for authentication into visual cryptography, enhancing its security capabilities.

Integration of Visual Cryptography with Other Security Protocols:

Researchers are working on seamlessly combining visual cryptography with existing security protocols, creating a comprehensive framework for secure image sharing and authentication.

What is Steganography?

Steganography is the practice of representing information within another message or physical object, in such a manner that the presence of the information is not evident to human inspection. In computing/electronic contexts, a computer file, message, image, or video is concealed within another file, message, image, or video. Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program, or protocol. Media files are ideal for steganographic transmission because of their

large size. For example, a sender might start with an innocuous image file and adjust the color of every hundredth pixel to correspond to a letter in the alphabet. The change is so subtle that someone who is not specifically looking for it is unlikely to notice the change [10].

Steganography vs. Cryptography:

Steganography:

Objective: Conceals the existence of a message within a carrier medium.

Method: Embeds information subtly to prevent detection.

Focus: Secrecy through obscurity.

Cryptography:

Objective: Secures the content of a message using mathematical algorithms.

Method: Transforms data using keys for confidentiality.

Focus: Ensures privacy and integrity of the information.

Comparison:

Steganography hides the existence, while cryptography secures the content.

Steganography relies on secrecy, and cryptography on complexity.

Both can be combined for a comprehensive approach to information security.

in this also specify each of their advantages and disadvantages

Types of Steganography Techniques:

LSB Steganography:

Method: Embeds information in the least significant bits of digital data (e.g., image pixels).

Advantage: Simple and widely applicable, especially in image steganography.

Disadvantage: Prone to detection with advanced analysis tools.

Frequency Domain Steganography:

Method: Alters the frequency domain representation of data, hiding information in spectral components.

Advantage: Offers robustness against certain types of analysis.

Disadvantage: Complexity increases compared to LSB steganography.

Spread Spectrum Steganography:

Method: Distributes hidden data across the spectrum of the carrier signal.

Advantage: Resilient against interference and noise.

Disadvantage: Requires precise synchronization for successful extraction.

Transform Domain Steganography:

Method: Conceals information by applying mathematical transformations to the data.

Advantage: Provides security through transformations like Fourier or wavelet.

Disadvantage: Requires computational overhead during embedding and extraction.

Text Steganography:

Method: Hides information within text, using techniques like format-based or linguistic steganography.

Advantage: Conceals data in plain sight within textual content.

Disadvantage: Limited capacity for hiding large amounts of information.

Audio Steganography:

Method: Embeds data in audio files, exploiting imperceptible alterations.

Advantage: Allows for covert communication within audio content.

Disadvantage: Limited capacity and susceptibility to quality degradation.

Video Steganography:

Method: Conceals information within video streams, exploiting temporal and spatial redundancies.

Advantage: Offers higher data capacity compared to images or audio.

Disadvantage: Complexity increases with the multimedia nature of video.

Network Steganography:

Method: Hides data within network protocols and traffic.

Advantage: Useful for covert communication in networked environments.

Disadvantage: Requires careful consideration of network behavior to avoid detection.

LSB IMAGE STEGANOGRAPHY

LSB Steganography is an image steganography technique in which messages are hidden inside an image by replacing each pixel's least significant bit with the bits of the message to be hidden.

To understand better, let's consider a digital image to be a 2D array of pixels. Each pixel contains values depending on its type and depth. We will consider the most widely used modes — RGB(3x8-bit pixels, true-color) and RGBA(4x8-bit pixels, true-color with transparency mask). These values range from 0–255, (8-bit values).

We can convert the message into decimal values and then into binary, by using the ASCII Table. Then, we iterate over the pixel values one by one, after converting them to binary, we replace each least significant bit with that message bits in a sequence.

To decode an encoded image, we simply reverse the process. Collect and store the last bits of each pixel then split them into groups of 8 and convert it back to ASCII characters to get the hidden message [6].

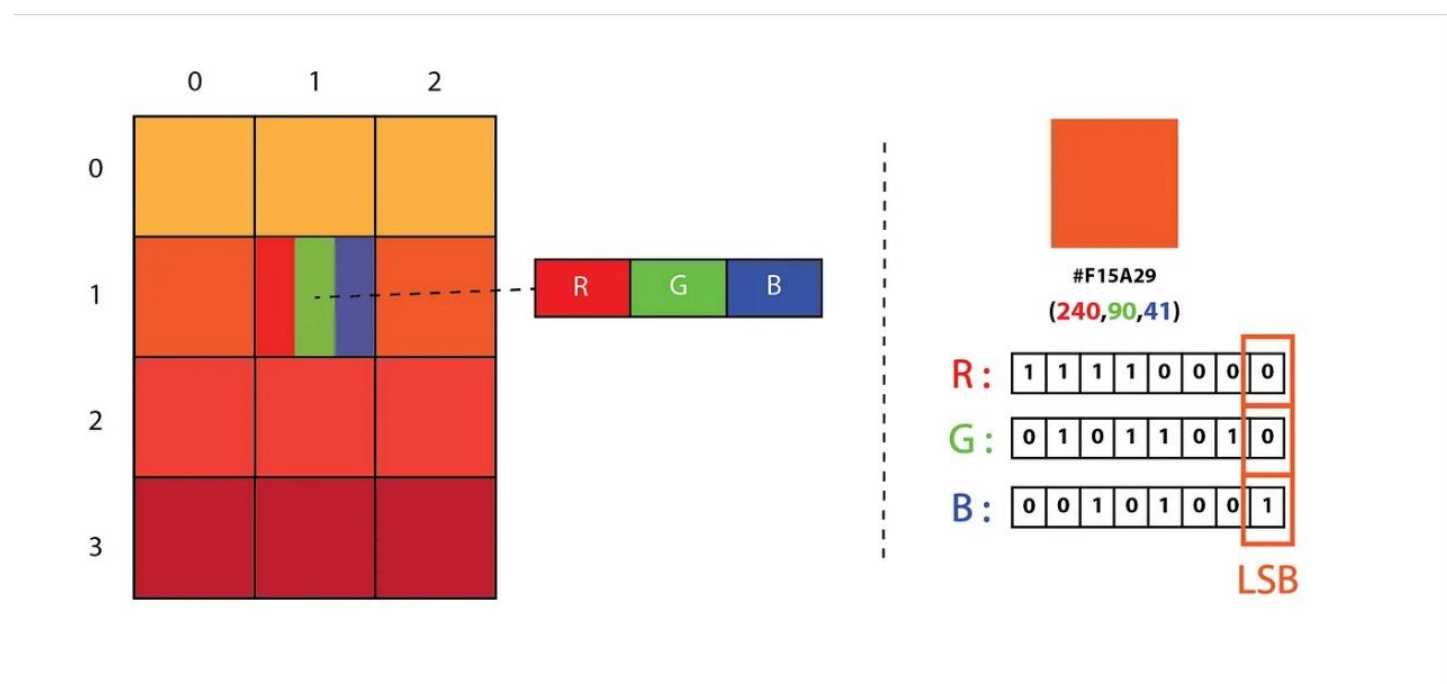


figure 1.2 LSB Steganography

Steganalysis and Detection Methods:

Steganalysis involves spotting hidden data in stego-images. Techniques include statistical and frequency analysis, and machine learning. Countermeasures involve algorithmic enhancements, encryption integration, and dynamic payload adjustment for added security.

Applications of LSB Steganography:

LSB Steganography finds use in covert communication, securing data transmission, digital watermarking for copyright protection, and forensic applications for identifying image tampering.

Advanced Topics and Research Directions:

Research focuses on robust steganography against attacks, extending steganography to audio and video formats, and integrating it with other security techniques like encryption for comprehensive data protection.

STREAMLIT

Streamlit is an open source app framework in Python language. It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc. With Streamlit, no callbacks are needed since widgets are treated as variables. Data caching simplifies and speeds up computation pipelines. Streamlit watches for changes on updates of the linked Git repository and the application will be deployed automatically in the shared link. It allows developers and data scientists to build interactive and customizable web interfaces using only Python code.

Simplified Web App Development:

Ease of Use: Streamlit is designed for simplicity. Developers can create web applications with just a few lines of Python code, making it accessible to both beginners and experienced programmers.

Rapid Prototyping: Streamlit enables quick prototyping of data-driven applications, reducing the time and effort needed for development.

Data Visualization:

Built-in Components: Streamlit provides a variety of built-in components for creating charts, plots, and interactive visualizations without the need for extensive coding.

Integration with Plotting Libraries: It seamlessly integrates with popular Python plotting libraries such as Matplotlib, Plotly, and Altair.

Interactive Widgets:

User Input Elements: Developers can easily add interactive widgets, such as sliders, buttons, and text inputs, allowing users to control and customize the application.

Real-Time Updates: Widgets can trigger real-time updates to the displayed content, enabling dynamic and interactive user experiences.

Deployment and Sharing:

Single-Command Deployment: Streamlit apps can be deployed with a single command, making it straightforward to share applications with others.

Compatibility: Streamlit apps can be deployed on various platforms, including cloud services, allowing for easy accessibility.

Customization and Theming:

Custom Components: Developers can create custom components for additional functionality tailored to their specific needs.

Theming: Streamlit supports theming options, allowing developers to customize the appearance of their applications.

Integration with Machine Learning:

Seamless Integration: Streamlit is often used in conjunction with popular machine learning libraries like TensorFlow and PyTorch to create interactive dashboards showcasing model outputs.

Open Source Community:

Active Community: Streamlit has a vibrant open-source community that contributes to its development and provides support through forums and documentation.

Supported File Types:

Image and Video Integration: Streamlit supports the integration of images and videos, enhancing the multimedia capabilities of web applications.

1.2 PROBLEM STATEMENT

The contemporary digital landscape faces a pressing challenge in ensuring secure communication and storage of sensitive information. Existing solutions often lack an optimal balance between concealment and efficient retrieval.. The primary challenge is to develop a method that securely transmits and stores confidential data while minimizing perceptual changes to the original image. This project responds to the need for a practical and efficient solution in the secure handling of sensitive information within various digital applications.

1.3 SIGNIFICANCE OF THE PROBLEM

The problem is significant as it directly addresses the challenge of balancing effective data concealment and efficient retrieval. In today's digital landscape, securing sensitive data is crucial, and current solutions often struggle to achieve this balance. Resolving this problem holds practical importance in enhancing data security practices, particularly in confidential communication and secure data storage. The successful resolution is expected to bring about meaningful improvements in digital security methodologies.

1.4 BRIEF DESCRIPTION OF SOLUTION APPROACH

The solution approach adopted in the "Secured Palette" project leverages a two-fold cryptographic strategy, combining LSB (Least Significant Bit) steganography with Visual Cryptography. LSB steganography is employed to subtly embed textual messages within the least significant bits of the image pixels, ensuring minimal perceptual changes to the cover image. This technique facilitates the covert transmission of information within the visual data. Simultaneously, Visual Cryptography is utilized to enhance the security of the concealed message. The original image is divided into two shares, generated in such a way that individual shares reveal no information about the hidden message. Only by combining both shares during the decoding process can the original image be reconstructed, adding an extra layer of security. The implementation is realized through a user-friendly web application developed using Streamlit in Python. This interface enables users to seamlessly input cover images and messages during encoding and upload image shares during decoding. The fusion of LSB steganography and Visual Cryptography in a user-centric web application presents a practical and innovative solution for secure data transmission and storage, with applications ranging from confidential communication to protected data archives

1.5 COMPARISON OF EXISTING APPROACH TO PROBLEM FRAMED

Existing Approaches	Technique	Weakness	How Secured Palette Overcomes Weakness
Digital noise	Adding noise to a digital signal	Can degrade the quality of the cover media	Secured Palette maintains image quality by using LSB steganography, which embeds data in the least significant bits of each pixel, minimizing visible distortions.
Redundancy in data	Utilizing the extra bits in error-correcting codes for embedding additional information	Can be limited by the amount of redundancy in the data	Secured Palette utilizes the entire capacity of the image for data embedding, maximizing the amount of information that can be hidden.

Physical modifications	Encoding data in physical modifications to a carrier medium	Can be difficult or expensive to implement	Secured Palette does not require any physical modifications to the carrier medium, making it more practical and cost-effective.
Secret sharing schemes	Dividing data into multiple shares for sharing and reconstruction	Can be complex to manage and requires multiple shares	Secured Palette employs visual cryptography, eliminating the need to manage multiple shares and enabling data reconstruction with just two shares.
Error-correcting codes	Embedding data in the redundancy added to the code for error protection	Can be complex to implement	Secured Palette's LSB steganography method is inherently self-correcting in case of minor errors, reducing the complexity and improving robustness.
Information dispersal algorithms	Dividing data into multiple shares for resilient data hiding	Can be complex to implement	Secured Palette does not require the implementation of complex algorithms, making it more accessible and user-friendly.
Spread spectrum modulation	Embedding data in the signal's frequency spectrum	Can be complex to implement	Secured Palette's LSB steganography method is simpler to implement compared to spread spectrum modulation, offering a balance between security and practicality.

Covert channels	Embedding data in the timing or structure of communication channels	Can be unreliable and susceptible to detection	Secured Palette utilizes robust steganography and visual cryptography techniques, making it less reliant on unreliable covert channels.
Zero-steganography	Exploiting limitations of data hiding algorithms for secure data embedding	Can be difficult to implement and requires careful design	Secured Palette's combination of LSB steganography and visual cryptography provides a more straightforward and practical approach to secure data embedding.

Table 1.1: comparison of existing approach to problem framed

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

VISUAL CRYPTOGRAPHY

- 1) Naor, Moni, and Adi Shamir. "Visual cryptography." In *Advances in Cryptology—EUROCRYPT'94: Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, May 9–12, 1994 Proceedings* 13, pp. 1-12. Springer Berlin Heidelberg, 1995.

The paper introduces a groundbreaking cryptographic scheme designed for the easy decryption of concealed images without the need for complex cryptographic computations. This visual cryptography approach combines a printed page of ciphertext with a printed transparency, serving as a secret key, allowing the original content to become visible when the two are stacked together, even though both initially appear as random noise

- 2) Pawar, Shital B., and N. M. Shahane. "Visual Secret Sharing Using Cryptography." *International Journal of Engineering Research* 3, no. 1 (2014): 31-33.

The paper discusses the Embedded Extended Visual Cryptography Scheme (EEVCS), which is a variation of the Visual Cryptography Scheme (VCS) used for sharing secret images. In VCS, a secret image is split into random shares, and the secret can be reconstructed by stacking these shares using logical OR operations. EEVCS enhances VCS by using meaningful covering shares, achieved by embedding random shares of the secret image into cover images.

- 3) Pandey, Anjney, and Subhranil Som. "Applications and usage of visual cryptography: A review." In *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pp. 375-381. IEEE, 2016.

The paper discusses the importance of data security and encryption in today's computer generation, particularly in light of the rise in cybercrime. Visual Cryptography is highlighted as a significant part of cryptography, with various application areas, including biometric security, watermarking, remote electronic voting, and bank customer identification.

- 4) Karolin, M., and Dr T. Meyyapan. "RGB based secret sharing scheme in color visual cryptography." *International Journal of Advanced Research in Computer and Communication Engineering* 4, no. 7 (2015).

The research paper introduces an RGB-based secret sharing scheme in the domain of Color Visual Cryptography, a cryptographic technique that divides images into transparent shares. Traditional Visual Cryptography has primarily focused on monochromatic images, and this study extends the methodology to the RGB color model, incorporating Red, Green, and Blue channels. The proposed scheme aims to secure information transmission by exploring the intricacies of color images, likely involving methods to split them

into transparent shares. By leveraging the RGB color model, the research enhances the applicability of Visual Cryptography to handle the complexities of color data, contributing to the broader field of secure image sharing and transmission.

IMAGE STEGANOGRAPHY

- 5) Subramanian, Nandhini, Omar Elharrouss, Somaya Al-Maadeed, and Ahmed Bouridane. "Image steganography: A review of the recent advances." IEEE access 9 (2021): 23409-23423.

The research paper, "Image Steganography: A Review of the Recent Advances," delves into the realm of multimedia data security with a particular emphasis on steganography and cryptography in the context of the Internet of Things (IoT). Focusing on conventional encryption for data security, the paper also introduces steganography as a method to clandestinely embed information within multimedia content. The integration of steganography and cryptography within the IoT framework is explored, showcasing the paper's comprehensive review of recent advances in image steganography.

- 6) Mstafa, Ramadhan J. "Information hiding in images using steganography techniques." ASEE, 2013.

The research paper, "Information Hiding in Images Using Steganography Techniques," addresses privacy concerns arising from the widespread distribution of information facilitated by technological advancements. The paper specifically concentrates on the role of steganography in mitigating these concerns by safeguarding sensitive data. Steganography is explored as a technique to conceal information within images, offering a method to protect data by embedding it within seemingly innocuous visual content. The focus on information hiding within images signifies the paper's contribution to enhancing privacy and security measures in the context of global information dissemination.

- 7) Somwanshi, D. R., and V. T. Humbe. "Half-Tone Visual Cryptography Scheme For RGB Color Images." Indian Journal of Science and Technology 16, no. 5 (2023): 357-366.

The linked paper titled "Half-Tone Visual Cryptography Scheme for RGB Color Images" explores a novel approach to Visual Cryptography tailored for RGB color images. Visual Cryptography involves dividing images into shares, and this paper focuses on adapting the technique to handle the complexities of color images represented in the RGB model. The term "Half-Tone" suggests a method involving the use of halftone patterns, commonly used in printing, to enhance the security of the visual cryptography scheme. By employing this approach, the research likely aims to provide a secure and efficient means of sharing and reconstructing RGB color images through visual cryptography.

- 8) Cheddad, Abbas, Joan Condell, Kevin Curran, and Paul Mc Kevitt. "Digital image steganography: Survey and analysis of current methods." Signal processing 90, no. 3 (2010): 727-752.

The linked paper, "Digital Image Steganography," likely delves into the realm of concealing information within digital images using steganographic techniques. While a detailed summary would require access to the full

paper, the title suggests a focus on digital image steganography. In general, digital image steganography involves hiding data within the pixels of an image without perceptibly altering its visual appearance. The paper may explore various steganographic methods, their applications, and potential implications for information security.

- 9) Gupta, Ravindra, Akanksha Jain, and Gajendra Singh. "Combine use of steganography and visual cryptography for secured data hiding in computer forensics." *International Journal of Computer Science and Information Technologies* 3, no. 3 (2012): 4366-4370.

The proposed system highlights a novel approach for creating a secure steganographic method and visual cryptography for data hiding in computer forensics. Although there has been extensive research work in the past, but majority of the research work has no much optimal consideration for robust security towards the encrypted image. The proposed method encodes the secret message in at least significant bits of the original image, where the pixels values of the encrypted image are modified by the genetic algorithm to retain their statistic characters, thereby making the detection of secret of message difficult. Use of Genetic algorithm has compelled the system for enhancing the security. The proposed system hides data in a real image and achieve its detection after under went to visual cryptography. The main aim of the proposed model is to design a feasible RS- resistance secure algorithm which combines the use of both steganography and visual cryptography for improving security, reliability and efficiency for secret message.

2.2 INTEGRATED SUMMARY OF LITERATURE STUDIED

The collective insights from these research papers align with the core principles of our project, "Secured Palette," which utilizes a combination of LSB Steganography and Visual Cryptography. The RGB-Based Secret Sharing Scheme in Color Visual Cryptography resonates with our project's emphasis on securing information transmission, especially in the context of color images. The exploration of Image Steganography's recent advances within the IoT context aligns with our project's focus on modern data security challenges.

Additionally, the investigation into privacy concerns and the role of Steganography in safeguarding sensitive data aligns with our project's overarching goal of concealing information within images for enhanced privacy. The proposal of a Half-Tone Visual Cryptography Scheme for RGB color images further contributes to our project's exploration of innovative approaches to secure Visual Cryptography, especially in the context of color images.

Furthermore, the collaborative research on combining Steganography and Visual Cryptography for secured data hiding in computer forensics echoes the practical integration aspect of our project. It emphasizes the real-world applicability of these techniques, aligning with our project's objective of providing a robust solution for secure information sharing and protection. In summary, the insights from these research papers complement and validate the methodologies employed in our project, reinforcing the relevance and effectiveness of combining LSB Steganography and Visual Cryptography for enhanced data security and confidentiality.

CHAPTER-3

REQUIREMENT ANALYSIS AND SOLUTION APPROACH

3.1 OVERALL DESCRIPTION OF THE PROJECT

Secured Palette is a web application that utilizes a combination of Least Significant Bit (LSB) steganography and Visual Cryptography (VC) to securely embed and retrieve sensitive information within digital images. The application seamlessly integrates both techniques, providing a robust and user-friendly platform for data protection.

ENCODING PROCESS:

Information Embedding (LSB Steganography): The user uploads an image and enters the confidential text message to be hidden. The application applies LSB steganography, modifying the least significant bits of the image's color pixels to embed the secret message. The altered image, now carrying the concealed message, is generated.

IMAGE SHARING (VISUAL CRYPTOGRAPHY):

The application splits the stego-image (the image carrying the hidden message) into two shares using VC. Each share contains a portion of the visual information required to reconstruct the original image. The user can download or save the individual shares for secure transmission or storage.

DECODING PROCESS:

Image Reconstruction (Visual Cryptography): The user uploads the two shares generated during encoding. The application aligns and superimposes the shares, employing VC to reveal the original stego-image.

Message Retrieval (LSB Steganography): The application extracts the hidden message from the reconstructed stego-image by extracting the modified least significant bits of the color pixels. The retrieved secret message is displayed to the user.

KEY FEATURES:

- **SECURE DATA HIDING:** Utilizes LSB steganography to embed confidential information within images
- **VISUAL CRYPTOGRAPHY IMPLEMENTATION:** Employs VC to split the stego-image into two shares for secure transmission and reconstruction
- **USER-FRIENDLY WEB INTERFACE:** Provides a simple and intuitive web interface for encoding and decoding messages

3.2 REQUIREMENT ANALYSIS

REQUIREMENT	DESCRIPTION
MATLAB to Python Conversion	The original steganography code was initially developed in MATLAB for concept analysis. Python was chosen as the primary programming language for its versatility, extensive libraries, and widespread community support. The MATLAB code was converted to Python to ensure compatibility with the Streamlit web application framework.
Streamlit for Web Development	Streamlit, a Python library designed for creating interactive and data-driven web applications, was chosen for its simplicity and ease of use. The Streamlit framework enabled the development of a user-friendly web interface for encoding and decoding messages. Streamlit's built-in functionalities for file uploading, image processing, and data visualization facilitated the implementation of the steganography and visual cryptography algorithms.
Python as the Primary Programming Language	Python was selected as the primary programming language due to its:
Versatility	Python's general-purpose nature made it suitable for various tasks, including image processing, data manipulation, and web development.
Extensive Libraries	Python's rich ecosystem of libraries provided ready-made tools for image processing, cryptography, and web development.
Widespread Community Support	Python's large and active community ensured easy access to support and resources.
Git for Version Control	Git, a distributed version control system, was employed to:
Track changes in the codebase	Git maintained a history of changes, allowing developers to revert to previous versions if necessary.

Manage collaborative development	Git facilitated collaboration among developers by enabling them to merge their changes and work on different parts of the project simultaneously.
Ensure project stability	Git's branching and merging capabilities helped maintain project stability by isolating changes and preventing conflicts.

Table 3.1 Requirement Analysis

3.3 SOLUTION APPROACH

1. INFORMATION EMBEDDING USING LSB STEGANOGRAPHY:

Utilize LSB steganography to embed confidential data within the least significant bits of the image's color pixels. Employ a robust embedding algorithm that ensures minimal distortion to the original image, preserving its visual quality.

2. IMAGE SHARING USING VISUAL CRYPTOGRAPHY:

Employ Visual Cryptography (VC) to split the stego-image (the image carrying the hidden message) into two shares. Each share contains a portion of the visual information required to reconstruct the original image. Distribute the shares securely, ensuring that only authorized parties have access to both shares. Implement the algorithm using Python and the Pillow library. Save the shares as PNG images.

Technical Details of Visual Cryptography for Colored Pixels:

Color-Based Visual Cryptography with Halftone Masks. Visual cryptography, a cryptographic technique, allows for secure image sharing by dividing an original image into two or more shares. Reconstructing the original image requires superimposing these shares. Color-based visual cryptography extends this technique to colored images by employing different halftone masks for each of the CMYK channels (cyan, magenta, yellow, and black).

Halftone Masks

Halftone masks are binary patterns that determine how the pixels of an original image are distributed among the shares. In color-based visual cryptography, these masks are specifically designed for each color channel. When the shares are superimposed, the halftone masks interact, allowing the human visual system to perceive the average of the pixels, effectively reconstructing the original image's colors.

Algorithm Implementation

Implementing color-based visual cryptography typically involves the following steps:

Image Preprocessing: Convert the original RGB image to a CMYK image to facilitate the halftone masking process.

Halftone Mask Generation: Generate two distinct halftone masks, one for each share, for each CMYK channel. The masks should be designed to ensure the reconstruction of the original image upon superposition.

Image Sharing: Divide the CMYK image into four sub-images, one for each channel. Apply the corresponding halftone mask to each sub-image, generating the two shares.

Image Reconstruction: Superimpose the corresponding shares from each channel to reconstruct the original CMYK image.

Color Restoration: Convert the reconstructed CMYK image back to RGB to obtain the final reconstructed colored image.

Advantages of Color-Based Visual Cryptography

Secure Image Sharing: Color-based visual cryptography effectively protects sensitive information embedded within images by requiring both shares for reconstruction.

Color Preservation: The technique preserves the original image's colors, enabling the sharing of color images without compromising visual fidelity.

Versatility: It can be applied to images of any size and resolution.

Disadvantages of Color-Based Visual Cryptography

Halftone Mask Size: The halftone masks can be large, increasing storage and transmission requirements.

Reconstructed Image Quality: The reconstructed image may exhibit some noise or artifacts depending on the halftone mask design.

3. MESSAGE RETRIEVAL USING LSB STEGANOGRAPHY:

Superimpose the two shares using VC to reconstruct the original stego-image.

Extract the hidden message from the reconstructed stego-image by extracting the modified least significant bits of the color pixels.

Least Significant Bit (LSB) Steganography

LSB steganography is a widely used technique for embedding confidential data within digital images. It exploits the human visual system's limited ability to perceive subtle changes in the least significant bits (LSBs) of an image's pixel values. By modifying these LSBs, secret data can be embedded without significantly altering the image's appearance.

Embedding Process

In LSB steganography, the embedding process involves modifying the LSBs of the image's color pixels to encode the secret data. This process typically involves the following steps:

Data Conversion: Convert the secret message into a binary format, represented as a sequence of 0s and 1s.

Bit Embedding: Embed the binary data into the image's LSBs. Replace the LSBs of selected pixels with the

corresponding bits of the secret message.

Data Hiding: Save the modified image, which now carries the hidden message, without any noticeable visual alterations.

Embedding Strategies

There are various strategies for embedding data using LSB steganography:

Random LSB Replacement: Randomly select LSBs to embed the data, spreading the modifications across the image.

Sequential LSB Embedding: Embed the data sequentially, replacing the LSBs of pixels in a row-wise or column-wise manner.

Adaptive LSB Replacement: Modify the LSBs based on the local pixel intensity, attempting to minimize any visible changes.

Data Extraction

The process of extracting the embedded data from the stego-image (the image carrying the hidden message) involves reversing the embedding process. This typically involves the following steps:

Bit Extraction: Extract the modified LSBs from the corresponding image pixels.

Data Reconfiguration: Reconstruct the binary sequence from the extracted LSBs.

Data Decoding: Convert the binary sequence back into the original secret message.

Embedding Capacity

The embedding capacity of LSB steganography depends on the image's size and color depth. Generally, higher-resolution and deeper color images can accommodate more embedded data without compromising visual quality.

Security Considerations

While LSB steganography provides a degree of concealment, it is not entirely foolproof. Sophisticated steganalysis techniques can detect the presence of hidden data, especially when embedded at high rates. To enhance security, additional layers of encryption or data obfuscation can be employed alongside LSB steganography.

CHAPTER-4

MODELING AND IMPLEMENTATION DETAILS

4.1 DESIGN DIAGRAMS

4.1.1 USE CASE DIAGRAM

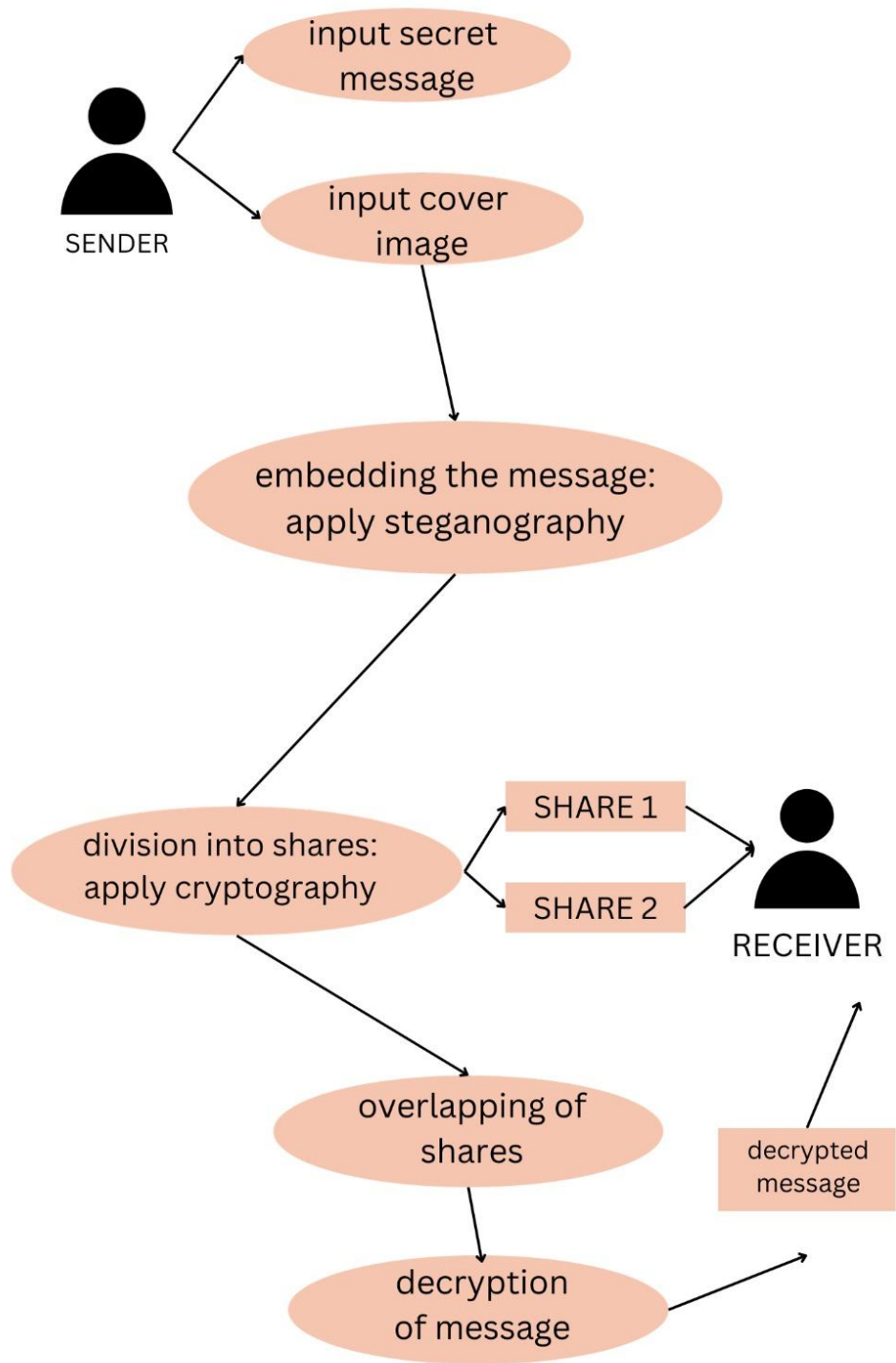


figure 4.1: use case diagram

4.1.2 CONTROL FLOW DIAGRAMS

ENCODING FLOW

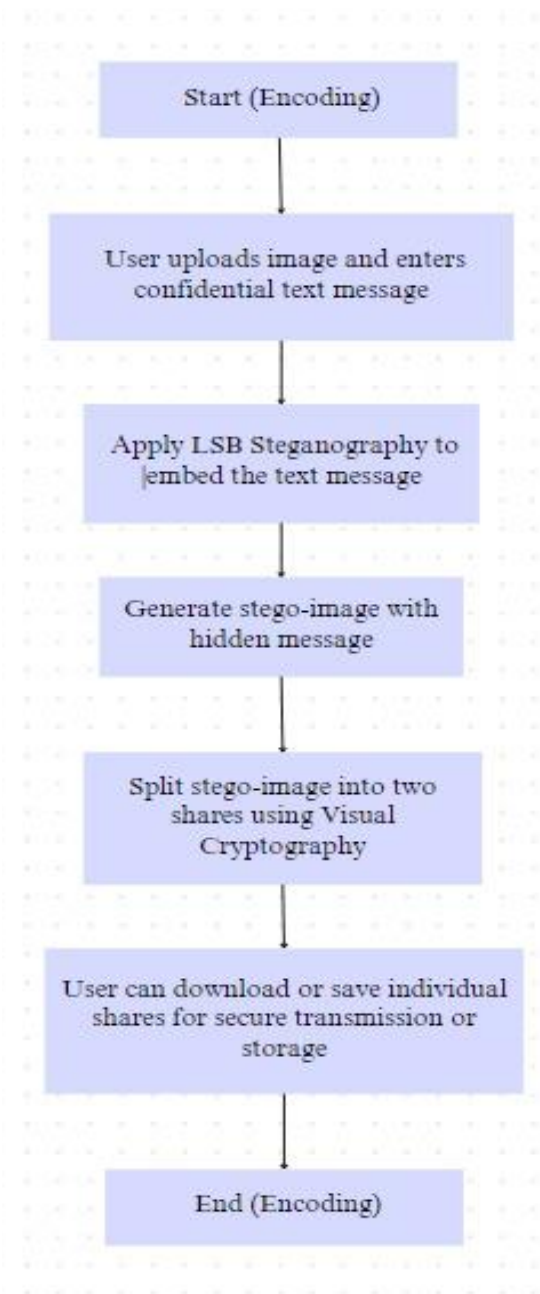


figure 4.2: use case diagram - encoding

DECODING FLOW

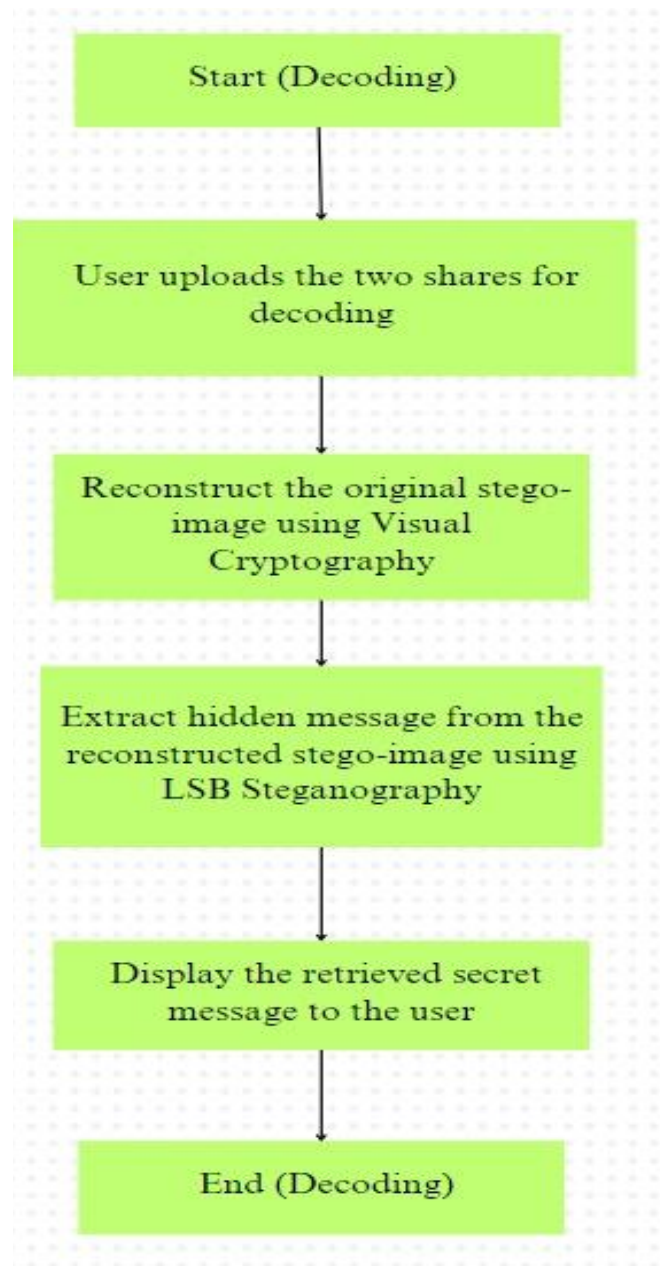


figure 4.3: use case diagram - decoding

4.1.3 CLASS DIAGRAM

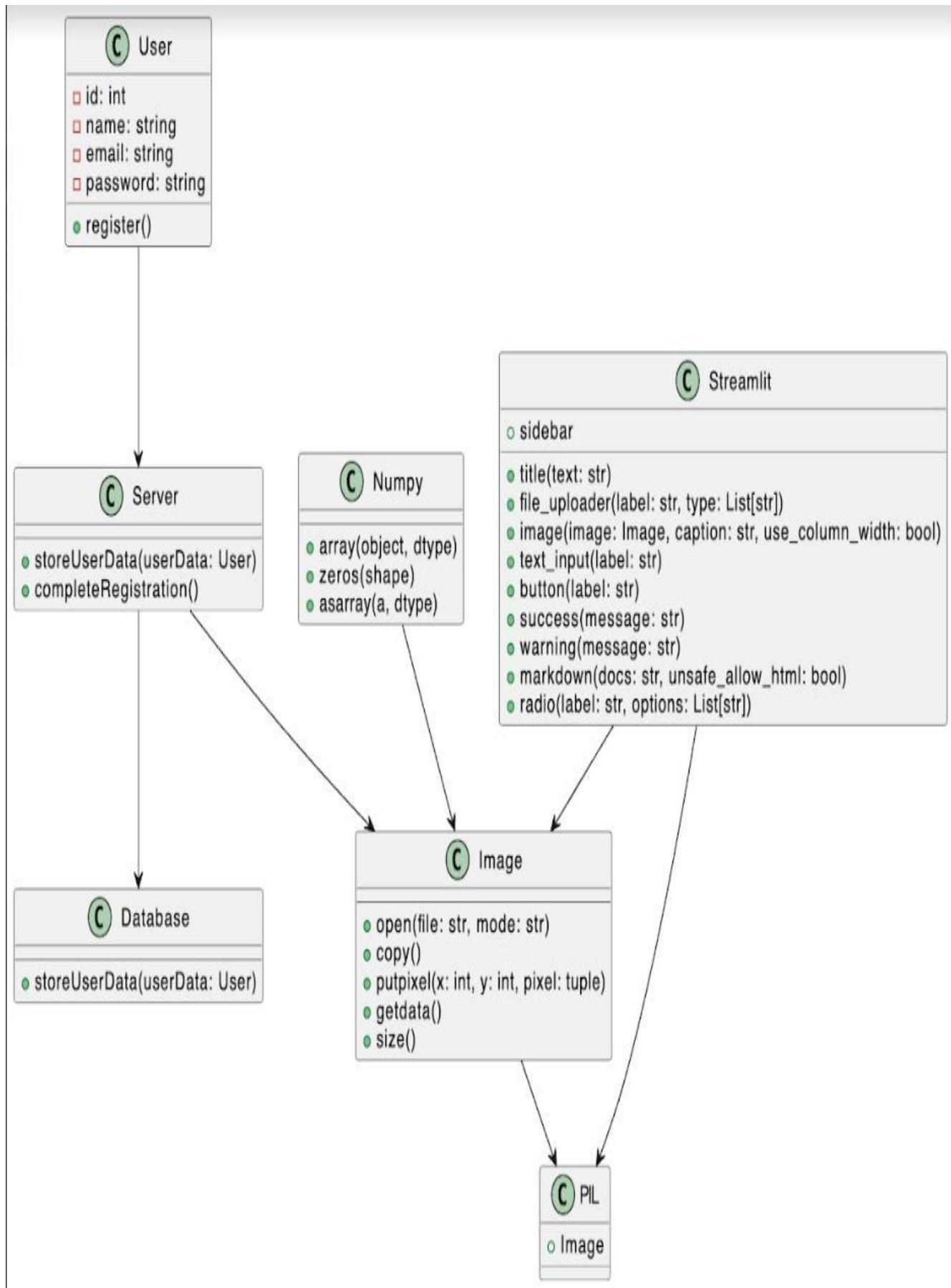


figure 4.4: class diagram

4.1.4 ACTIVITY DIAGRAM

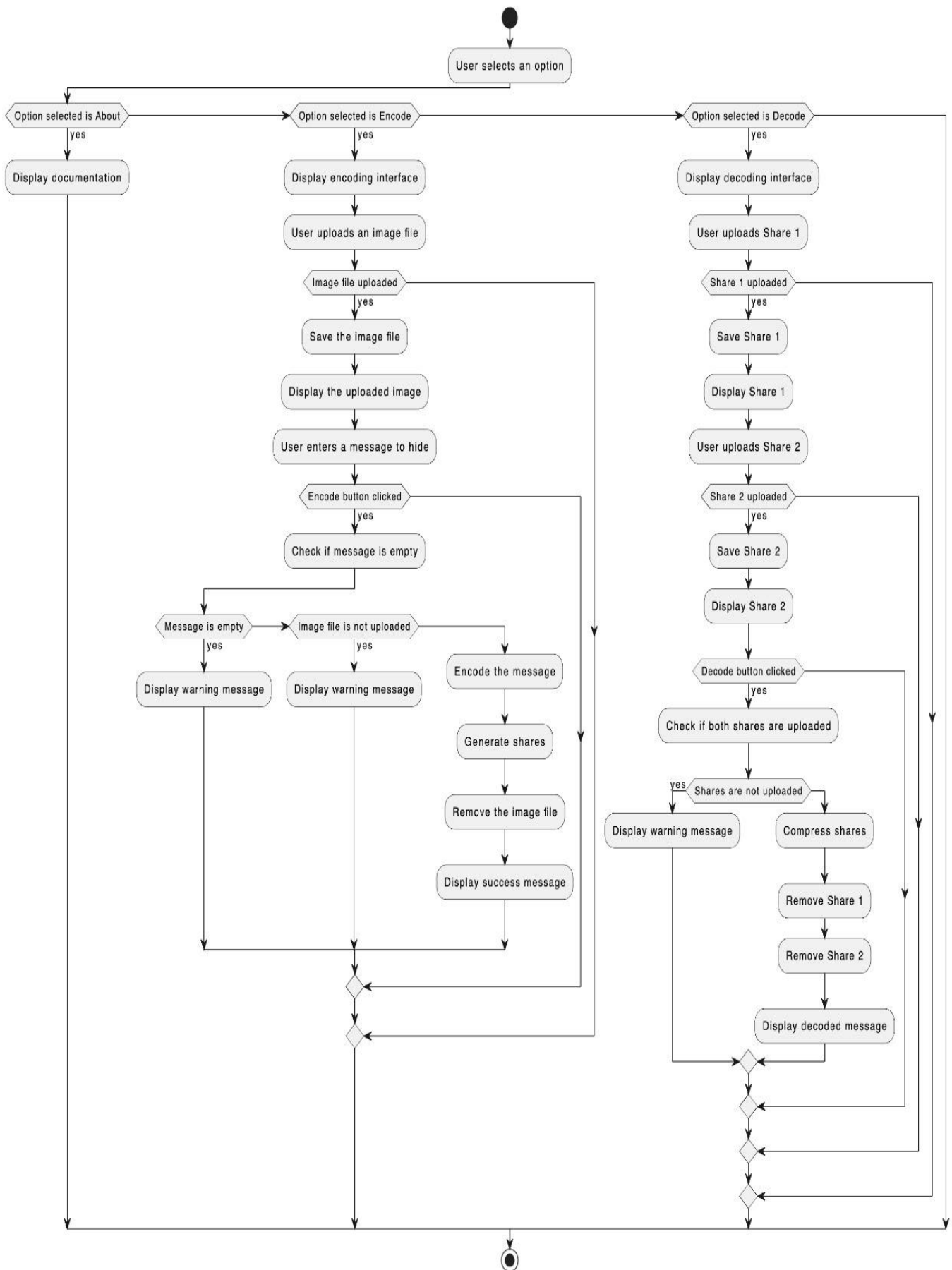


figure 4.5: activity diagram

4.2 IMPLEMENTATION DETAILS AND ISSUES

IMPLEMENTATION DETAILS

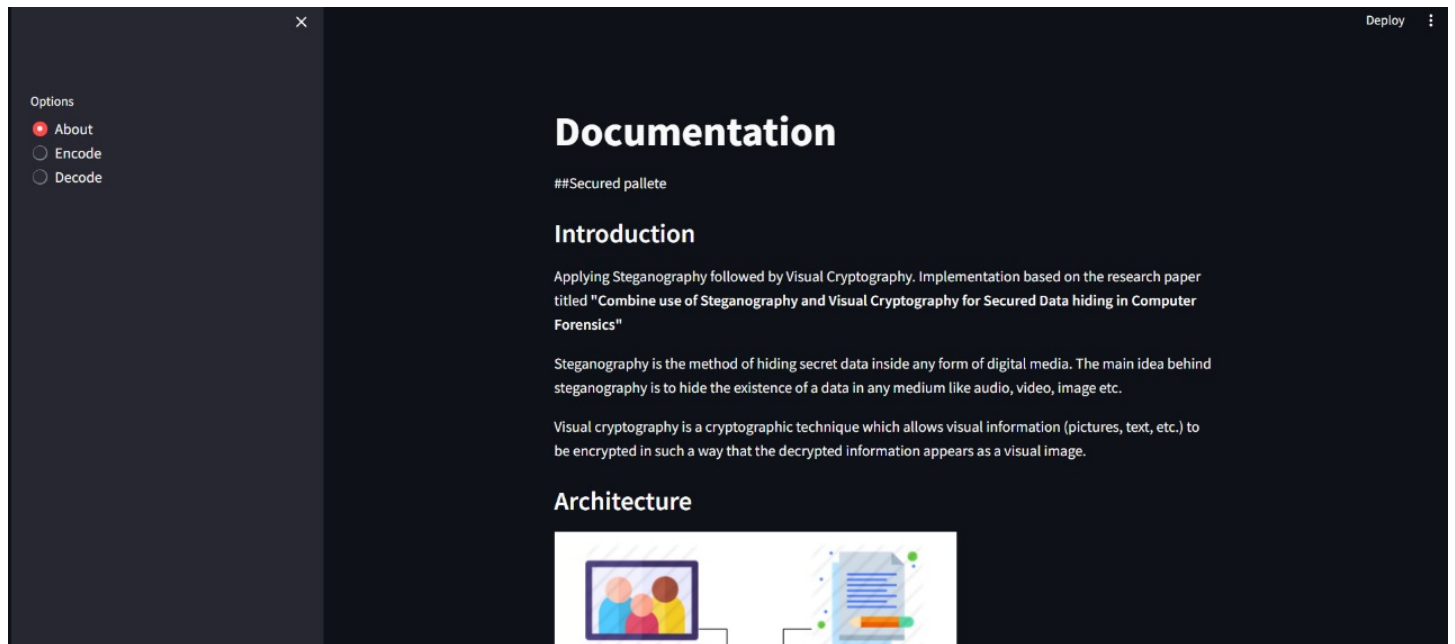


Figure 4.6: Leveraging the principles of steganography and visual cryptography the project facilitates secure collaboration and information sharing.

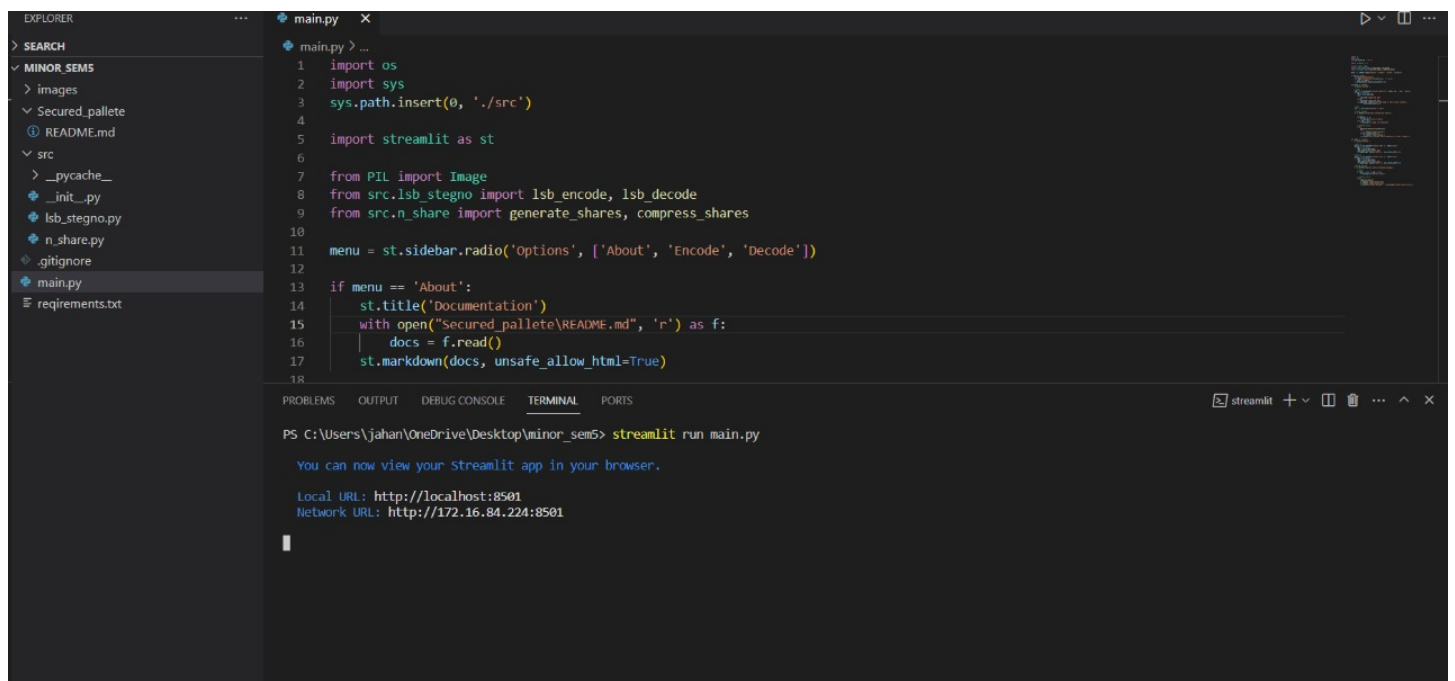


Figure 4.7: Streamlit Integration: The project leverages the Streamlit framework to create a dynamic and interactive web interface.

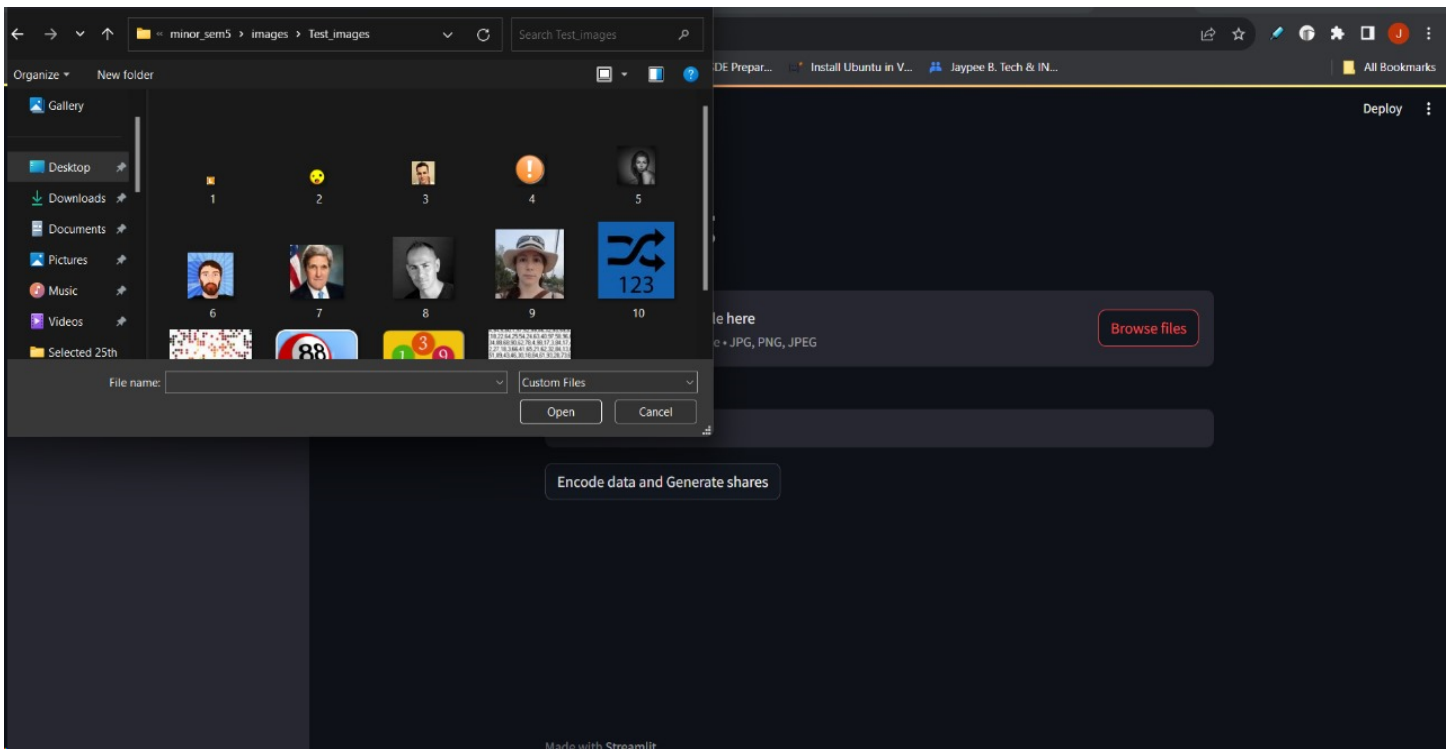


Figure 4.8: Image Selection: Users can choose any image from their local device to serve as the cover for hiding the secret message. A user-friendly interface allows for easy navigation and selection of images.

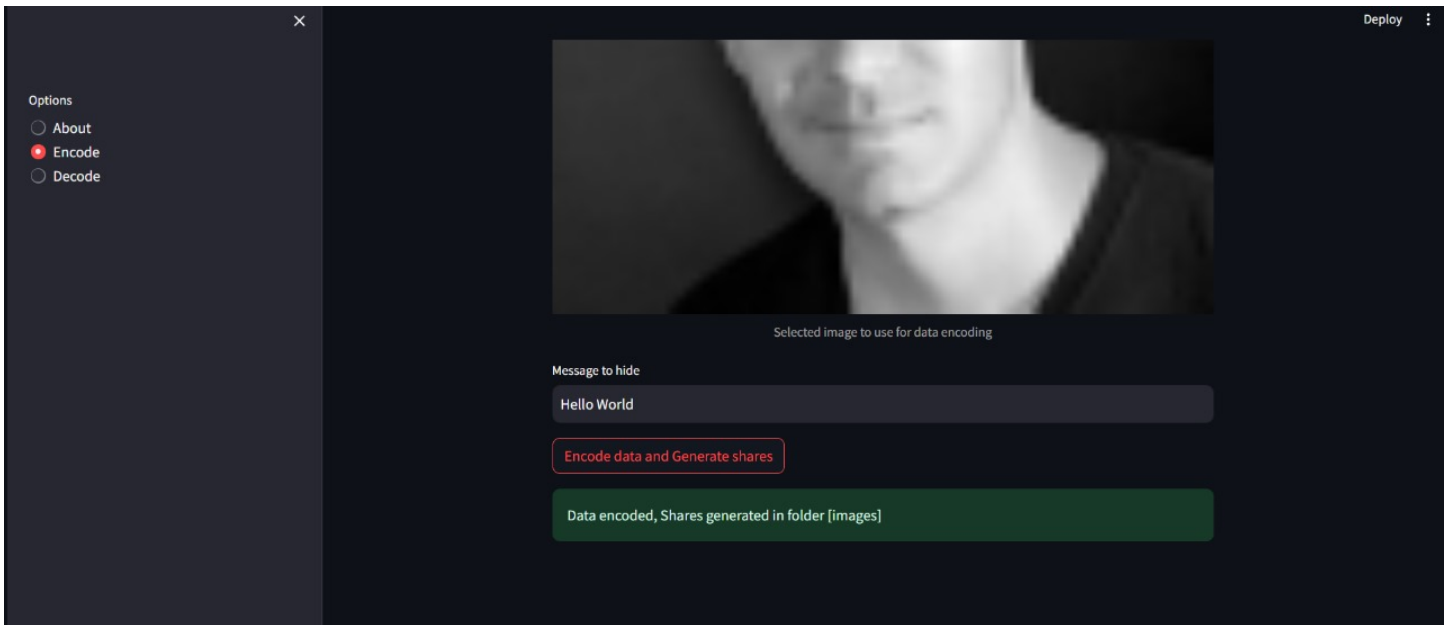


Figure 4.9: Message Input: The web interface includes a text input field where users can type the secret message they want to embed in the selected image.

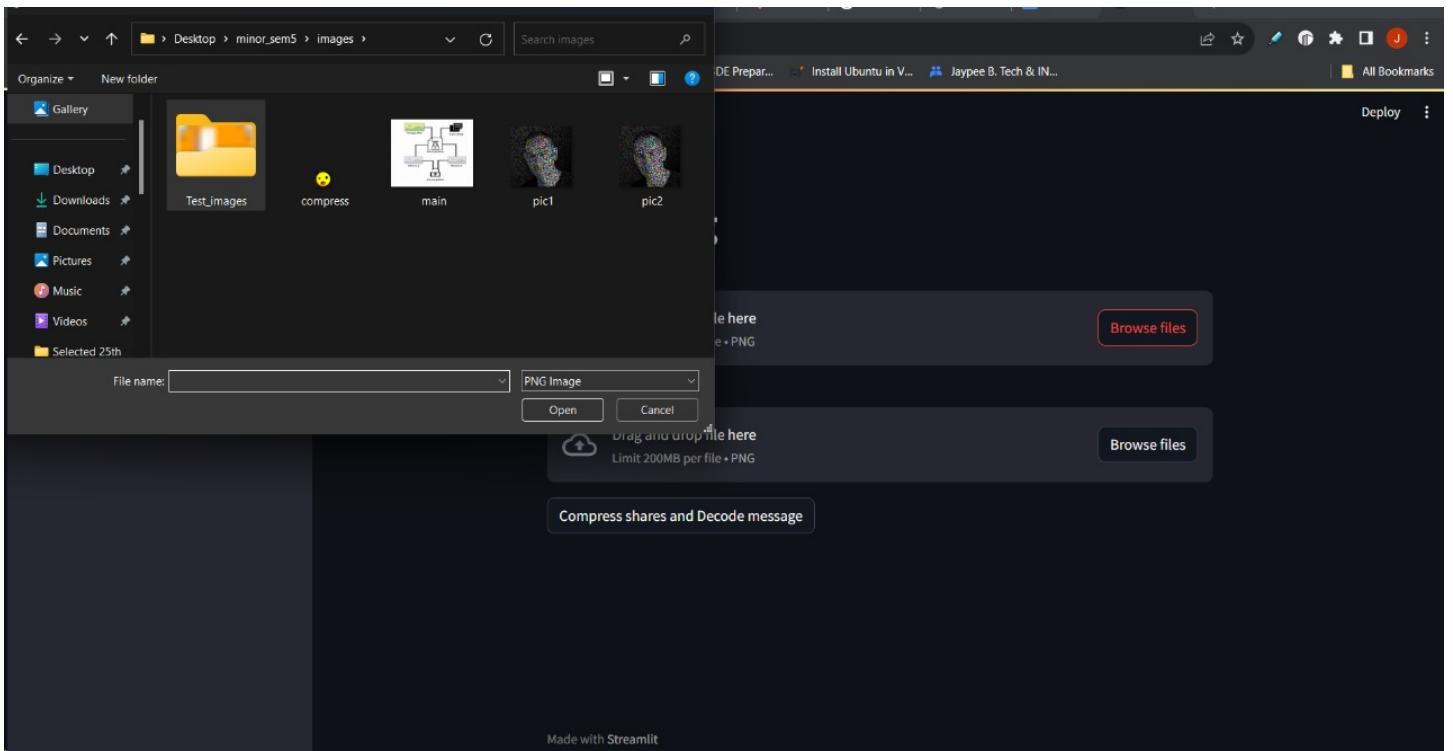


Figure 4.10: Share Upload: Users can upload the generated shares of the encoded image through the web interface.

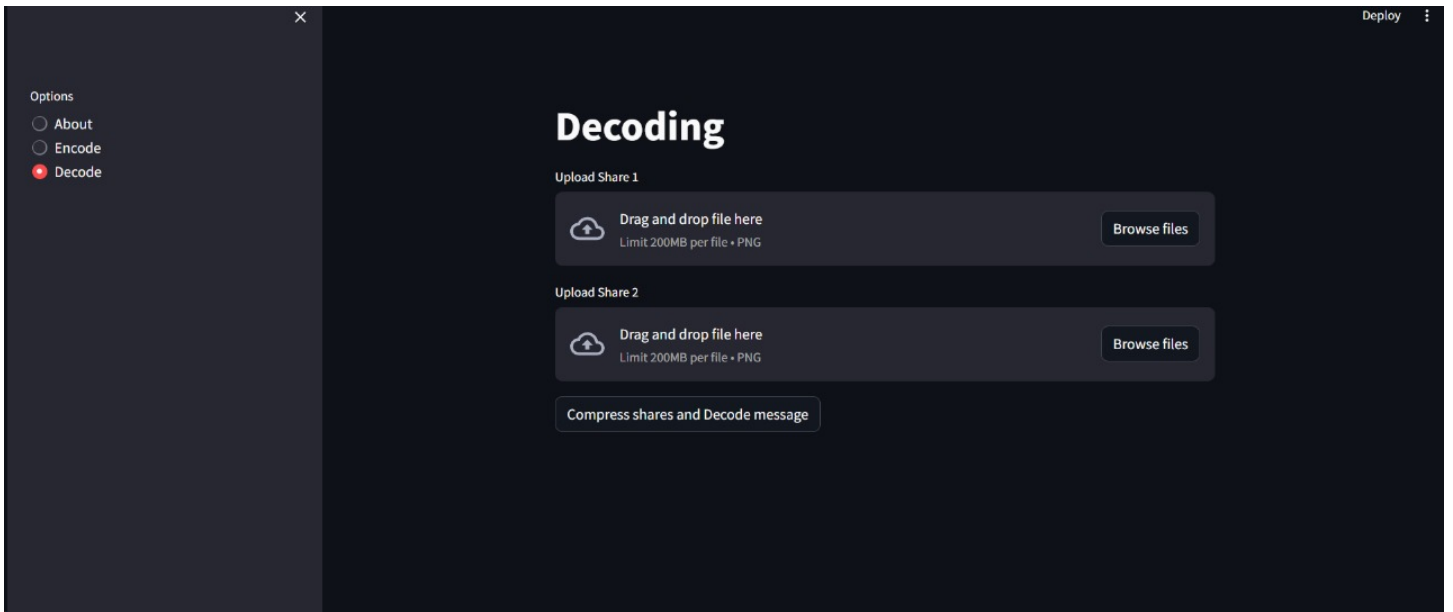


Figure 4.11: Decoding Process: The web interface includes interactive controls to initiate the decoding process. Visual feedback informs users about the progress and success of the decoding operation.

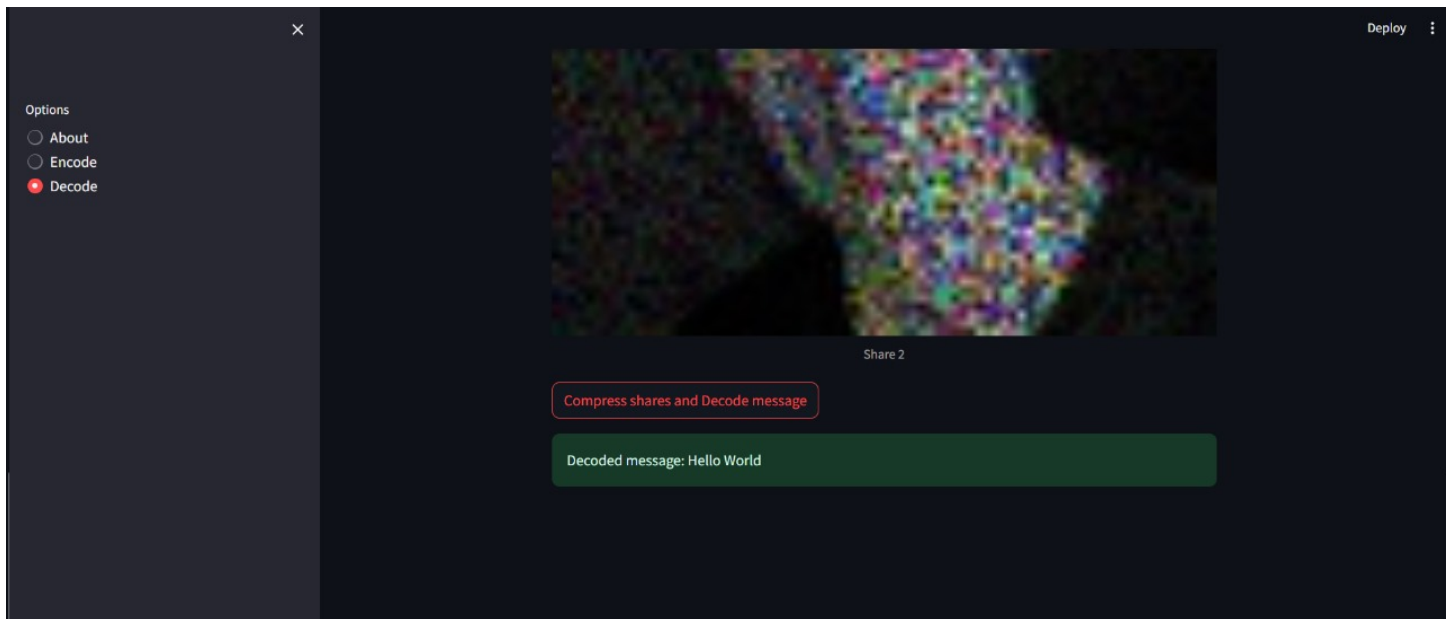


Figure 4.12: Message Display: Once the decoding is successful, the secret message is displayed on the interface for the user to read. Clear and concise messaging ensures a straightforward user experience.

ALGORITHMS

Algorithm: Decode Binary Data to String

Input: pixels - a list of pixel values (integers)

Output: data - a decoded string

Procedure:

1. Start
2. Initialize an empty string variable binstr
3. Loop through the first 8 pixels in the list (pixels[:8])
 - 3.1 If the current pixel value (i) is even, append '0' to binstr
 - 3.2 Else, append '1' to binstr
4. Convert binstr to an integer using base 2 and then to a character, and append it to the data string
5. Check if the last pixel value in the list (pixels[-1]) is odd
 - 5.1 If true, return the final decoded data string

6. Repeat the decoding process for the next set of 8 pixels

7. Stop

Algorithm: Split Image Based on Random Factor

Input:

- n: a random integer factor
- data: a 3D array representing image data

Output:

- img1: first split of the image
- img2: second split of the image

Procedure:

1. Start
2. Initialize variables i, j, k to iterate through the image data
3. Loop through each pixel in the image data:
 - 3.1 Generate a random integer n using `np.random.randint(data[i, j, k] + 1)`
 - 3.2 Assign `img1[i, j, k] = n`
 - 3.3 Assign `img2[i, j, k] = data[i, j, k] - n`
4. Continue looping until all pixels are processed
5. Output img1 and img2 as the two splits of the image based on the random factor
6. Stop

Algorithm: Compress Shares

Input:

- img1: first set of image shares
- img2: second set of image shares

Output:

- img: compressed image data

Procedure:

1. Start
2. Initialize variables i, j, k to iterate through the image data
3. Loop through each pixel in the image data:
 - 3.1 Set $\text{img}[i, j, k] = \text{img1}[i, j, k] + \text{img2}[i, j, k]$
4. Continue looping until all pixels are processed
5. Output img as the compressed image data
6. Stop

4.3 RISK ANALYSIS AND MITIGATION

Risk	Description	Likelihood	Impact	Mitigation Strategy
Data loss	The image and/or text message could be lost during the encoding or decoding process.	Medium	High	Implement data redundancy and error-correction techniques to ensure data integrity.
Unauthorized access	The image and/or text message could be accessed by unauthorized individuals.	Low	Medium	Implement strong encryption and access control mechanisms to protect sensitive data.
Steganalysis	The presence of the hidden message could be detected using steganalysis techniques.	Low	Medium	Employ more sophisticated steganography algorithms that are more resistant to detection.
Visual cryptography imperfections	The decoded image may have imperfections due to the visual cryptography process.	Medium	Low	Use high-quality image processing techniques to minimize imperfections.

Table 4.1 Risk Analysis and Mitigation

CHAPTER-5

TESTING

5.1 TESTING PLAN

This testing plan outlines the various testing methodologies, test cases, and error handling strategies employed to validate the application's functionality and resilience.

Testing Methodologies:

Unit Testing: Individual modules and functions are tested in isolation to ensure their correct operation and adherence to specifications.

Integration Testing: Different modules are integrated and tested as a cohesive unit to verify their interactions and data flow.

System Testing: The entire web application is tested as a complete system to validate its functionality, performance, and security against real-world scenarios.

Robustness Testing: The application is subjected to unexpected inputs, stress conditions, and edge cases to assess its resilience and ability to handle unexpected situations.

5.2 COMPONENT DECOMPOSITION AND TYPE OF TESTING REQUIRED

Component	Type of Testing
Image Upload	Functional, Input Validation, Boundary Value Analysis
Image Preprocessing	Functional, Performance, Error Handling
LSB Steganography Encoding	Functional, Security, Steganalysis Resistance
Visual Cryptography Encoding	Functional, Data Integrity, Error Diffusion
Image Overlapping and Decoding	Functional, Visual Quality, Error Correction
LSB Steganography Decoding	Functional, Security, Data Recovery
User Interface	Usability, Accessibility, Responsiveness

Table: 5.1

Category 1: Test Cases for Different Image Formats

Objective: Verify that the application can successfully encode and decode messages in images of different formats, including PNG, JPG, and JPEG.

Test Cases:

PNG: Encode a message in a PNG image and decode it successfully.




IMAGE	DIMENSIONS
	30 x 30
	50 x 50
	130 x 130

Table: 5.2

JPG: Encode a message in a JPG image and decode it successfully.




IMAGE	DIMENSIONS
	40 x 40
	60 x 60
	90 x 90

Table: 5.3

JPEG: Encode a message in a JPEG image and decode it successfully.





IMAGE	DIMENSIONS
	20 x 20
	70 x 70
	90 x 90
	100 x 100

Table: 5.4

Category 2: Test Cases for Different Image Sizes

Objective: Verify that the application can handle images of varying sizes, ensuring that the encoding and decoding processes remain functional regardless of image dimensions.

Test Cases:

Encode a message in a small image, medium-sized image and large image and decode it successfully.


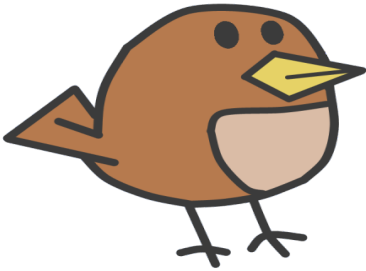

IMAGE	DIMENSIONS
	20 x 20
	512 x 512
	1024 x 1024

Table: 5.5

Category 3: Test Cases for Different Image Types (Colored and Black and White)

Objective: Verify that the application can successfully encode and decode messages in both colored and black and white images, ensuring that the color information does not interfere with the steganographic process.

Test Cases:

Encode a message in a colored image and black and white and decode it successfully.


IMAGE	PIXEL DETAILS
	Coloured pixels
<pre>14,34,2,60,1,37,34,33,06,34,33,33,0, 18,22,64,25,54,24,63,40,97,58,96, 34,88,68,90,62,78,4,98,17,3,84,17, 2,27,18,3,66,41,65,21,62,32,84,13, 51,89,43,46,30,18,84,61,93,28,73,6 7,73,93,23,0,38,15,37,54,22,52,77, 35,64,71,87,73,78,65,19,61,90,29, 98,29,69,47,37,39,85,81,11,14,90, 11,75,64,21,69,82,21,39,71,37,93,3 7,83,46,88,15,45,17,32,38,33,3,47, 25,68,20,10,23,78,2,53,50,54,27,50 61,63,9,86,100,40,20,44,73,100,27 36,97,13,86,43,14,68,28,27,62,55, 1,22,60,60,22,40,65,24,100,77,61,61</pre>	Grayscale pixels

Table: 5.6

Additional Testing Considerations:

Message Length: Test the application's ability to handle messages of varying lengths, ensuring that the encoding and decoding processes remain efficient and accurate regardless of message size.

Image Quality: Evaluate the impact of image quality on the encoding and decoding processes, ensuring that message integrity is maintained even with compressed or low-quality images.

Error Handling: Test the application's error handling mechanisms, ensuring that it can gracefully handle unexpected scenarios and provide appropriate feedback to the user.

5.3 ERROR AND EXCEPTION HANDLING

Error Handling in main.py

The try-except block in main.py is used to handle the potential error of not being able to open the image file selected by the user. The user is prompted to upload an image file, and if the file is not uploaded or is not of a valid format, an error message is displayed. This error handling prevents the application from crashing if the user provides an invalid image file.

Python

```
img = st.file_uploader('Upload image file', type=['jpg', 'png', 'jpeg'])
```

```
if img is
```

```
not
```

```
None:
```

```
    img = Image.open(img)
```

```
    try:
```

```
        img.save('images/img.jpg')
```

```
    except:
```

```
        img.save('images/img.png')
```

Error Handling in lsb_stego.py

The try-except block in lsb_stego.py is used to handle the potential error of not being able to open the image file that contains the encoded data. The lsb_decode function attempts to open the image file, and if the file is not found or is not of a valid format, an error message is returned. This error handling prevents the application from crashing if the user provides an invalid image file.

Python

```
try:
```

```
    image = Image.open(file_name, 'r')
```

```
except:
```

```
    print('Error opening image file')
```

```
    return None
```

These try-except blocks are essential for ensuring that the application can handle unexpected errors gracefully and provide appropriate feedback to the user. Without these error handling mechanisms, the application could crash or behave unpredictably, leading to a poor user experience.

5.4 LIMITATIONS OF THE SOLUTION

Image Size Limitations:

The Secured Palette web application is currently limited to handling images of a specific size range. This limitation arises from the computational complexity of both steganography and visual cryptography algorithms. As the size of an image increases, the computational cost of these algorithms grows exponentially, making it impractical to process large images in real time.

To address this limitation, future work could explore more efficient steganography and visual cryptography algorithms that can handle larger images without compromising performance or security. Additionally, implementing parallel processing techniques could help distribute the computational load and improve scalability.

Image Quality Degradation:

The steganography process involved in hiding secret messages within images may introduce some visual artifacts into the encoded image. These artifacts are caused by the subtle modifications made to the pixel values of the image to embed the message. While these artifacts may not be noticeable to the naked eye, they can become more apparent when the image is zoomed in or subjected to image processing techniques.

To minimize image quality degradation, future work could investigate alternative steganography methods that are less prone to introducing visual artifacts. Additionally, implementing adaptive steganography techniques that adjust the embedding strength based on the local image characteristics could help preserve image quality while maintaining message integrity.

Visual Cryptography Imperfections:

The decoded image obtained from visual cryptography may exhibit minor imperfections due to the inherent nature of the technique. Visual cryptography relies on the overlap of two or more image shares to reveal the hidden message. However, due to the discrete nature of pixel values, the edges of the decoded image may appear slightly jagged or distorted.

To address these imperfections, future work could explore alternative visual cryptography techniques that utilize more sophisticated encoding and decoding algorithms. Additionally, implementing image processing techniques, such as edge smoothing or image fusion, could help enhance the quality of the decoded image.

Steganalysis Vulnerabilities:

The Secured Palette web application may be susceptible to advanced steganalysis techniques. Steganalysis aims to detect the presence of hidden messages within digital media, such as images. While the specific steganography algorithm used in the application is designed to be resistant to common steganalysis methods, more sophisticated techniques may be able to detect the presence of hidden messages.

To enhance the security of the application, future work could investigate more robust steganography algorithms that are less susceptible to steganalysis. Additionally, implementing techniques such as stegano-noise and cover image selection could further improve the resistance against steganalysis attacks.

CHAPTER-6

FINDINGS, CONCLUSION AND FUTURE WORK

6.1 FINDINGS

The Secured Palette web application successfully demonstrates the integration of LSB steganography and visual cryptography techniques for secure data hiding and transmission. The encoding process effectively embeds the text message within the input image using LSB steganography, imperceptibly altering the image without compromising its visual quality. The visual cryptography scheme reliably divides the encoded image into two shares, ensuring that neither share alone reveals any sensitive information.

The decoding process accurately reconstructs the original image and extracts the hidden text message using a combination of visual cryptography and LSB steganography. Overlapping the shares reconstructs the image, and applying LSB steganography decoding reveals the embedded text message.

The Secured Palette web application provides a user-friendly interface that facilitates both encoding and decoding processes. Users can easily input the image, enter the text message to be hidden, and browse the shares for decoding. The application seamlessly handles the encryption and decryption of sensitive information, ensuring secure data transmission.

The Secured Palette web application effectively embeds and extracts text messages using a combination of LSB steganography and visual cryptography.

The encoding process maintains the visual quality of the image while concealing the text message.

The visual cryptography scheme ensures that neither share alone reveals any sensitive information.

The decoding process accurately reconstructs the original image and extracts the hidden text message.

The Secured Palette web application provides a user-friendly interface for both encoding and decoding.

Overall, the Secured Palette web application demonstrates a robust approach to secure data hiding and transmission, offering a practical solution for safeguarding sensitive information.

6.2 CONCLUSION

The Secured Palette web application effectively combines LSB steganography and visual cryptography (VC) techniques to achieve secure data hiding and transmission. This approach offers a robust solution for safeguarding confidential information by embedding it within images while maintaining image quality and ensuring data security. LSB steganography adeptly conceals the confidential message within the least significant bits (LSBs) of the image's pixels.

This imperceptible alteration preserves the image's visual quality, making it challenging for unauthorized individuals to detect the hidden message. The embedding process leverages the inherent redundancy in digital images to seamlessly conceal the data without raising suspicion. VC provides an additional layer of security by splitting the stego-image, the image containing the embedded message, into two shares. When viewed individually, each share appears as random noise, rendering the hidden message indecipherable. Only by overlapping the two shares does the original image and the embedded message become visible.

This VC mechanism ensures that even if one share is compromised, the hidden message remains protected. The proposed approach prioritizes preserving the original image's visual quality. By carefully selecting the LSBs for data embedding, the algorithm minimizes perceptual changes to the image, ensuring that the hidden message is embedded without compromising the image's authenticity or aesthetic appeal. This preservation of image quality is crucial for maintaining the plausibility of the encoded image and preventing suspicion about the presence of hidden data.

6.3 FUTURE WORK

The Secured Palette web application has demonstrated the effectiveness of combining LSB steganography and visual cryptography for secure data hiding and transmission. To further enhance its capabilities and explore its potential, we plan to work upon the following domains in the future:

1. Integration with Encryption Techniques

Investigate the integration of encryption techniques, such as AES or RSA, with the proposed approach to provide an additional layer of security for the hidden message. This would ensure that even if the visual cryptography shares are compromised, the embedded message would remain protected by encryption.

2. Application to Video Data

Explore the applicability of the proposed approach to secure data hiding in videos, considering the challenges and opportunities unique to video steganography. Videos present larger data volumes and temporal complexities, requiring specialized techniques for embedding and extracting data while maintaining video quality and frame synchronization.

3. Performance Optimization

Investigate optimization techniques to improve the computational efficiency of the encoding and decoding processes while maintaining data security and image quality. This could involve optimizing the LSB embedding process, utilizing parallel processing techniques, or developing lightweight visual cryptography algorithms.

4. Human Imperceptibility Analysis

Conduct extensive human perceptual studies to quantitatively assess the imperceptibility of the hidden message in embedded images. This would involve evaluating the project under various lighting conditions and image resolutions, ensuring that the hidden message remains undetectable to the human eye.

5. Multi-Device Collaboration

Incorporate multi-device collaboration into the application, enabling users to upload the two visual cryptographic shares from different devices. This would cater to scenarios where individuals are geographically dispersed or using different devices to access the application.

6. Cloud Integration

Integrate the application with cloud storage platforms for secure data storage and retrieval. This would allow users to store and access their encoded images and visual cryptography shares in a secure and centralized location.

7. Steganography Algorithm Diversification

Explore the integration of alternative steganography algorithms, such as F5 or HUGO, with the proposed approach to evaluate their performance and robustness compared to LSB steganography.

8. Visual Cryptography Scheme Enhancement

Investigate advanced visual cryptography schemes that offer higher levels of security and improved image quality. This could involve exploring techniques such as multiple-secret sharing or error correction coding.

9. Application to Real-Time Data Streaming

Explore the applicability of the proposed approach to secure data hiding in real-time data streams, such as video conferencing or live surveillance feeds. This would require optimizing the encoding and decoding processes to handle real-time data transmission while maintaining security and imperceptibility.

10. Integration with Existing Applications

Develop integrations with existing applications, such as email clients or document management systems, to seamlessly incorporate secure data hiding capabilities into everyday workflows. This would enhance the adoption and usability of the proposed approach. Addressing these areas for future work, we plan to further enhance the capabilities of the Secured Palette web application, expand its applicability to a wider range of secure data hiding and transmission scenarios, and make it a valuable tool for secure communication and data protection in various domains.

REFERENCES

- [1] Cheddad, Abbas, Joan Condell, Kevin Curran, and Paul Mc Kevitt. "Digital image steganography: Survey and analysis of current methods." *Signal processing* 90, no. 3 (2010): 727-752.
- [2] Gupta, Ravindra, Akanksha Jain, and Gajendra Singh. "Combine use of steganography and visual cryptography for secured data hiding in computer forensics." *International Journal of Computer Science and Information Technologies* 3, no. 3 (2012): 4366-4370.
- [3] Information on steganography available at <https://en.wikipedia.org/wiki/Steganography>
- [4] Information on visual cryptography available at https://en.wikipedia.org/wiki/Visual_cryptography#:~:text=Visual%20cryptography%20is%20a%20cryptographic,who%20developed%20it%20in%201994
- [5] Karolin, M., and Dr T. Meyyapan. "RGB based secret sharing scheme in color visual cryptography." *International Journal of Advanced Research in Computer and Communication Engineering* 4, no. 7 (2015).
- [6] Mstafa, Ramadhan J. "Information hiding in images using steganography techniques." *ASEE*, 2013.
- [7] Naor, Moni, and Adi Shamir. "Visual cryptography." In *Advances in Cryptology—EUROCRYPT'94: Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, May 9–12, 1994 Proceedings* 13, pp. 1-12. Springer Berlin Heidelberg, 1995.
- [8] Pandey, Anjney, and Subhranil Som. "Applications and usage of visual cryptography: A review." In *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pp. 375-381. IEEE, 2016.
- [9] Pawar, Shital B., and N. M. Shahane. "Visual Secret Sharing Using Cryptography." *International Journal of Engineering Research* 3, no. 1 (2014): 31-33.
- [10] Subramanian, Nandhini, Omar Elharrouss, Somaya Al-Maadeed, and Ahmed Bouridane. "Image steganography: A review of the recent advances." *IEEE access* 9 (2021): 23409-23423.
- [11] Somwanshi, D. R., and V. T. Humbe. "Half-Tone Visual Cryptography Scheme For RGB Color Images." *Indian Journal of Science and Technology* 16, no. 5 (2023): 357-366.