IMAGE STEGANOGRAPHY USING AES ALGORITHM

MINI PROJECT REPORT

Submitted by

FARHAN CHOLAKKAL (RCE20CS016)

MOHAMMED BILAL BASHEER (RCE20CS031)

FARSEEN N V (RCE20CS017)

LAZIM (RCE20CS026)

HISHAM C (RCE20CS022)

In partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

ROYAL COLLEGE OF ENGINEERING & TECHNOLOGY
AKKIKAVU

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

JUNE 2023

IMAGE STEGANOGRAPHY USING AES ALGORITHM

MINI PROJECT REPORT

Submitted by

FARHAN CHOLAKKAL (RCE20CS016)

MOHAMMED BILAL BASHEER (RCE20CS031)

FARSEEN N V (RCE20CS017)

LAZIM (RCE20CS026)

HISHAM C (RCE20CS022)

In partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

ROYAL COLLEGE OF ENGINEERING & TECHNOLOGY
AKKIKAVU

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

JUNE 2023

ROYAL COLLEGE OF ENGINEERING & TECHNOLOGY AKKIKAVU
(NAAC ACCREDITED INSTITUTION)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that this seminar report **"IMAGE STEGANOGRAPHY USING AES ALGORITHM"** is the Bonafide work of **"FARHAN CHOLAKKAL (RCE20CS016), FARSEEN N V(RCE20CS017), LAZIM (RCE20CS026), HISHAM C (RCE20CS022), MOHAMMED BILAL BASHEER (RCE20CS031)"** of Department of Computer Science and Engineering in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering under the APJ Abdul Kalam Technological University – KTU during the year 2022-2023.

Ms. Anu Appukuttan	**Ms. Ihsana Muhammed**	**Ms. Niya E C**
Faculty Supervisor	HOD In-charge	Project Coordinator

Place: Akkikavu

Date:

INSTITUTE VISION AND MISSION

Vision

“To continuously grow as a

Resourceful,

Outstanding,

Youthful,

Adaptive Institution in the field of Engineering and Technology habituating

Lifelong learning”.

Mission

“To groom the youth into eminent technocrats with lifelong learning skills to meet future requirements, deep sense of social responsibility, strong ethical values and a global outlook, to face the challenges of the changing world”.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Vision

"To grow as a premier department of Computer Science and Engineering capable of facing challenges of the modern Computing industry for the betterment of Society through most appropriate and Ethical practices."

Mission

- To impart high quality education to students with strong foundation of Computer Science and Engineering through Outcome Based Education (OBE).
- To empower the students with the required skills to solve the complex technological problems of modem society and to conduct multidisciplinary research for developing innovative solutions.
- To provide a learning ambience to enhance problem solving skills, leadership qualities, team spirits and ethical responsibilities with a commitment to lifelong learning.
- To establish industry institute interaction activities to enhance the entrepreneurship skills.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Programme specific outcomes (PSOs)

- Ability to analyze, design and implement ethical sustainable solutions in the field of computer science.
- Ability to use problem solving skills in the broad area of programming concepts and manage different projects in interdisciplinary field.
- Ability to understand the evolutionary changes in computing and creating an innovative career path to be an entrepreneur and lifelong learner with moral values and ethics.

Programme Educational Objectives (PEOs)

- To enable the graduates as globally competent engineering specialists, to solve engineering problems in the field of Computer Science based on industry and social requirements.
- To impart knowledge and expertise in undergoing socially innovative projects with ethical practices which enable the students to become leaders, entrepreneurs and social reformers.
- To encourage higher studies and research, opening wider opportunities to students in teaching, innovation and product development.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Programme Outcomes (POs)

PO1.Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3.Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4.Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6. The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8> Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ACKNOWLEDGEMENT

Every success stands as a testimony not only to the hardship but also to hearts behind it. Likewise, the present project work has been undertaken and completed with direct and indirect help from many people and I would like to acknowledge the same.

First and foremost, I take immense pleasure in thanking the **Management** and respected Principal, **Dr. Devi V.**, for providing me with the wider facilities.

I express my sincere thanks to **Ms. Ihasana Muhammed**, HoD in charge of Computer Science and engineering for giving me opportunity to present this project and for timely suggestions.

I wish to express my deep sense of gratitude to the project coordinator **Ms. Niya E C.** Assistant Professor, Department of Computer Science and Engineering, who coordinated in right path.

Words are inadequate in offering my thanks to Guide **Ms. Anu Appukuttan,** Assistant Professor, Department of Computer Science and Engineering, for her encouragement and guidance in carrying out the project.

Needless to mention that the **teaching and non - teaching faculty members** had been the source of inspiration and timely support in the conduct of my seminar. I would also like to express my heartfelt thanks to my beloved **parents** for their blessings, my **classmates** for their help and wishes for the successful completion of this project.

Above all I would like to thank the **Almighty** for the blessings that helped me to complete this venture smoothly.

ABSTRACT

Image steganography, a method of concealing secret data within digital images, is a critical aspect of information security. This abstract presents a novel approach to image steganography using AES (Advanced Encryption Standard) encryption, which enhances the confidentiality and integrity of the hidden information. The proposed system utilizes the AES algorithm to encrypt the secret message, providing robust protection. Subsequently, the encrypted message is embedded within the cover image using a steganographic technique that alters the least significant bits (LSBs) of the image's color channels. This process ensures imperceptibility, preserving the visual quality of the image while effectively hiding the secret data. To retrieve the hidden message, the stego-image is processed, and the embedded data is extracted. The extracted message is then decrypted using the appropriate AES decryption algorithm and the secret key. This approach combines the strengths of AES encryption and steganography, offering a secure and efficient solution for concealing sensitive information within images. The integration of AES encryption ensures that only authorized recipients, possessing the correct key, can access and decipher the original secret message, providing an additional layer of protection against unauthorized access and maintaining the confidentiality of the hidden data.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Steganography, derived from the Greek words "steganos" (meaning covered) and "graphein" (meaning writing), is the art and science of hiding secret information within seemingly innocuous carriers to ensure covert communication. Unlike encryption, which focuses on scrambling the contents of a message to prevent unauthorized access, steganography aims to conceal the very existence of the communication itself. By embedding secret data within digital media, such as images, audio files, or videos, steganography provides a clandestine channel for transmitting sensitive information. The concept of steganography dates back centuries, with historical examples including invisible ink, microdots, and hidden messages within paintings. In the digital age, steganography has gained significant prominence due to the widespread use of digital media and the ease of sharing and transmitting such content. The primary objective of steganography is to ensure that the embedded information remains undetectable to unintended recipients. The carrier medium, known as the cover or stego medium, undergoes subtle modifications that can be imperceptible to human senses. These modifications exploit the characteristics of the carrier medium, such as the least significant bits of image pixels or the inaudible frequency ranges in audio signals, to hide the secret data.

1. BACKGROUND

In recent times, Internet has essentially become the most effective and fast media for digital communication. Large amount of data such as image, text, audio, video etc., are transmitted over internet. Providing security, confidentiality and authenticity of data being transmitted is an important issue. Steganography, a technique that predates modern cryptography, has a rich history dating back to ancient times. The practice of hiding messages can be traced to ancient Greece, where secret communications were concealed on wax tablets covered with a layer of wax for unsuspecting messengers. In medieval times, invisible inks were used to hide written messages, and during World War II, microdots were employed to transmit secret information. In the digital era, steganography has evolved to exploit the characteristics of digital media. Digital steganography emerged in the 1980s, leveraging the discrete nature of digital data to embed information within files such as images, audio, and video. The proliferation of digital media and the ease of sharing such files have made steganography increasingly relevant.

Steganography operates on the principle of imperceptibility, aiming to make the hidden information undetectable to unauthorized observers. Various techniques have been developed, including the manipulation of least significant bits (LSBs) in images, the modification of audio frequency ranges, and the manipulation of file structures.

1.1 MOTIVATION

The motivation behind image steganography lies in the need for secure and covert communication of sensitive information in the digital age. There are several key reasons that drive the use and development of image steganography:

- **Covert Communication:** Image steganography enables individuals or organizations to communicate secretly without drawing attention to the existence of the communication itself. By concealing messages within innocent-looking images, steganography provides a clandestine channel for transmitting sensitive data, reducing the risk of interception.
- **Enhanced Security:** Traditional encryption methods attract attention to encrypted data, potentially inviting decryption attempts by adversaries. In contrast, steganography offers an additional layer of security by making the very existence of the secret message less conspicuous, as the carrier image appears unaltered to casual observers.
- **Inconspicuous Data Exchange:** In scenarios where, direct encryption might raise suspicion or be impractical, such as in restrictive environments or covert operations, steganography proves invaluable. It allows parties to exchange information without drawing attention to their activities.
- **Digital Media Proliferation:** With the exponential growth of digital media sharing on the internet, images have become a ubiquitous means of communication. This widespread use of images makes them a natural choice for hiding secret information.
- **Watermarking and Copyright Protection:** Steganography is utilized in digital watermarking to embed copyright information within images and videos, enabling content creators to claim ownership and protect their intellectual property.
- **Data Hiding in Steganalysis:** On the flip side, steganalysis, the detection of hidden information, motivates researchers to develop better steganography techniques to create more robust and undetectable hiding schemes.

1.2 OBJECTIVES

Objectives for Image Steganography Using AES Encryption:

- **Enhanced Security:** The primary objective is to enhance the security of hidden information within digital images by integrating AES encryption. AES is a widely recognized and robust encryption algorithm, known for its resistance against cryptographic attacks. By leveraging AES encryption, the system aims to provide a higher level of confidentiality and integrity for the hidden data.
- **Imperceptibility:** Another objective is to ensure that the modifications made to the cover image during the embedding process are imperceptible to the human eye. By carefully manipulating the least significant bits (LSBs) of the image pixels, the system aims to maintain the visual quality and integrity of the cover image while effectively hiding the secret message.
- **Authorized Access Only:** The system seeks to ensure that only authorized recipients possessing the correct decryption key can access and decipher the original secret message. By employing AES encryption, the system strengthens the access control mechanism, protecting the hidden data from unauthorized access or decryption attempts.
- **Practical Applicability:** An objective is to develop a practical and efficient system for image steganography using AES encryption. The system should be capable of handling various image formats, accommodating different message sizes, and performing the encryption and embedding processes in a timely manner.

REPORT ORGANIZATION

Chapter 1 describes the introduction of the project; Chapter 2 includes the literature survey. Chapter 3 discuss about various problems in a current technology. Chapter 4 discusses about the proposed method. Chapter 5 gives the overview of the implementation of the proposed system. Chapter 6 describes the conclusion and recommendation of the paper.

CHAPTER 2

LITERATURE SURVEY

Steganography is the study of invisible communication that usually deals with the ways of hiding the existence of the communicated message.

Noor Alhuda F. Abbas[1], Nida Abdulredha[2], Raed Khalid Ibrahim[3][1], proposed the paper "Security and imperceptibility improving of image steganography using pixel allocation and random function techniques". Information security is one of the main aspects of processes and methodologies in the technical age of information and communication. The security of information should be a key priority in the secret exchange of information between two parties. In order to ensure the security of information, there are some strategies that are used, and they include steganography and cryptography. An effective digital image-steganographic method based on odd/even pixel allocation and random function to increase the security and imperceptibility has been improved. This lately developed outline has been verified for increasing the security and imperceptibility to determine the existent problems. Huffman coding has been used to modify secret data prior embedding stage; this modified equivalent secret data that prevent the secret data from attackers to increase the secret data capacities. The main objective of our scheme is to boost the peak-signal-to-noise-ratio (PSNR) of the stego cover and stop against any attack. The size of the secret data also increases. The results confirm good PSNR values in addition of these findings confirmed the proposed method eligibility.

M. Mary Shanthi Rani , K.Rosemary Euphrasia [2], proposed the paper "Data security through QR code encryption and steganography". A The art of information hiding has become an important issue in the recent years as security of information has become a big concern in this internet era. Cryptography and Steganography play major role for secured data transfer. Steganography stands for concealed writing; it hides the message inside a cover medium. Cryptography conceals the content of a message by encryption. QR (Quick Response) Codes are 2-dimensional bar codes that encode text strings. They are able to encode information in both vertical and horizontal direction, thus able to encode more information.

In this paper a novel approach is proposed for secret communication by combining the concepts of Steganography and QR codes. The proposed system makes use of the advantages of QR codes and Steganography to enhance data security. In this algorithm the secret message is encoded into QR codes using QR code generator.

Moni Naor and Adi Shamir [3], proposed the paper "Visual Cryptography". In this paper we consider the problem of encrypting written material (printed text, handwritten notes, pictures, etc.) in a perfectly secure way which can be decoded directly by the human visual system. The basic model consists of a printed page of ciphertext .This basic model can be extended into a visual variant of the k out of n secret sharing problem: Given a written message, we would like to generate n transparencies so that the original message is visible if any k of them are stacked together, but totally invisible if fewer than k transparencies are stacked together (or analyzed by any other method). The original encryption problem can be considered as a 2 out of 2 secret sharing problem. The main results of this work include practical implementations of a k out of n visual secret sharing scheme for small values of k and n, as well as efficient asymptotic constructions which can be proven optimal within certain classes of schemes.

S. M. Masud Karim, Md. Saifur Rahman, Md. Ismail Hossain [4], proposed the paper "A New Approach for LSB Based Image Steganography using Secret Key'' This paper introduces a best approach for Least Significant Bit (LSB) based on image steganography that enhances the existing LSB substitution techniques to improve the security level of hidden information. It is a new approach to substitute LSB of RGB true color image. The new security conception hides secret information within the LSB of image where a secret key encrypts the hidden information to protect it from unauthorized users. anyone can extract the hidden information. In our paper, hidden information is stored into different position of LSB of image depending on the secret key.

The experimental results show that the proposed method is an effective way to integrate hidden information reporting without significant distortion. And it is very difficult for the unauthorized users to identify the changes in stego image. The use of the secret key gives a way to secure the information from illegal user.

Shahrin bin Sahib Alaa [5], proposed the paper "An Introduction to Image Steganography Techniques". In this paper Art of data hiding in digital media, steganography and watermarking, aims to embed secret data into cover with purpose of identification, copyright protection, and annotation. The main constraint factors of this process are message data quantity, necessity of invariability of embedded data under distortions like lossy compression, third party removal, or modification. Data hiding techniques fall into three categories of cryptography, steganography, and watermarking. Watermarking and particularly steganography tends to conceal presence of hidden data while cryptography makes data gibberish.

Data hiding has formidable technical challenges. Perceptual or statistical holes to be filled with data in host signals are likely to be removed by means of lossy signal compression. Important factor to achieve successful data hiding technique is to find holes which are not convenient to be exploited by compression algorithms.

CHAPTER 3

PROBLEMS OF CURRENT TECHNOLOGIES

Current image steganography technologies face several challenges and limitations thatt impact their effectiveness and security. Here are some of the key problems associated with existing approaches:

- **Capacity and Payload Size:** One of the major challenges is the limited capacity for hiding data within images. Most steganographic techniques can only embed a relatively small amount of data, which may be insufficient for certain applications. Increasing the payload size often leads to a higher likelihood of detection, compromising the security and effectiveness of the technique.
- **Robustness and Security:** Many existing steganography methods are susceptible to attacks and detection. Advanced steganalysis techniques, such as statistical analysis, machine learning algorithms, or visual inspection, can potentially uncover the presence of hidden information. The challenge lies in developing robust hiding techniques that can withstand these detection methods, ensuring the security and integrity of the hidden data.
- **Visual Quality and Distortion:** Embedding data within an image can introduce visual artifacts or distortions that are detectable to the human eye. Maintaining the visual quality of the cover image while effectively hiding the secret data is a significant challenge. Striking a balance between imperceptibility and the amount of data that can be concealed within an image remains a constant challenge.
- **Resistance against Image Processing Operations:** Many steganographic techniques are vulnerable to common image processing operations, such as compression, resizing, or format conversion. These operations can alter the embedded data or cause its loss, rendering the hidden information unrecoverable. Developing steganographic methods that are resilient to such operations is crucial for maintaining the integrity of the hidden data.
- **Key Management and Authentication:** Proper key management and authentication are essential for secure steganographic communication.

Ensuring that only authorized parties have access to the encryption and decryption keys is crucial for maintaining the confidentiality of the hidden data. Establishing secure key exchange protocols and robust authentication mechanisms is a challenge that needs to be addressed.

- **Application and Platform Dependency:** Some steganographic techniques may be specific to certain platforms, software, or file formats, limiting their applicability and interoperability across different systems. Developing platform-independent and widely compatible steganography methods would enhance their usability and practicality.

CHAPTER 4

PROPOSED METHOD

The proposed system of image steganography using AES encryption aims to address the limitations and challenges of current approaches. It combines the robust AES encryption algorithm with imperceptible data hiding techniques. The system encrypts the secret message using AES and then embeds it within the cover image using steganographic methods that modify the least significant bits of the image pixels. The embedded message is seamlessly incorporated, maintaining visual quality. To extract the hidden message, the stego-image is processed, and the data is retrieved. By integrating AES encryption with steganography, the system provides enhanced security, confidentiality, and authorized access to the hidden information within digital images. The seamless integration of steganography techniques maintains the visual quality of the cover image, making it difficult for unauthorized individuals to detect the presence of the hidden message. Furthermore, the system allows for efficient extraction and decryption of the hidden message by authorized recipients possessing the correct key. This combined approach provides a secure and practical solution for concealing sensitive information within digital images, catering to the need for confidential communication and data protection.

Cover image + secret key + hidden information = Stego-image

Advantages

Image steganography using AES encryption offers several significant advantages:

- **Enhanced Security:** AES encryption, known for its robustness and widespread adoption, provides a high level of security for the hidden data within digital images. The encryption process ensures that the secret message remains confidential and protected from unauthorized access or decryption attempts.
- **Confidentiality:** By integrating AES encryption, the system ensures that only authorized recipients possessing the correct decryption key can access and decipher the original secret message. This helps maintain the confidentiality of the hidden information and prevents unauthorized disclosure.

- **Imperceptibility:** The system employs steganography techniques to seamlessly embed the encrypted message within the cover image. These techniques carefully manipulate the least significant bits (LSBs) of the image pixels, ensuring that the modifications are imperceptible to the human eye. The visual quality and integrity of the cover image are maintained, preventing suspicion or detection.
- **Resistance to Attacks:** AES encryption provides robust protection against cryptographic attacks, making it highly resistant to brute-force and other common attack methods. The integration of AES encryption in image steganography enhances the overall security of the system, safeguarding the hidden data from various attacks and ensuring its integrity.
- **Versatility and Applicability:** The combination of image steganography and AES encryption can be applied to various image formats and sizes, making it a versatile solution for concealing sensitive information. The system can be integrated into existing applications or communication channels, offering practical applicability for secure data transmission.
- **Compliance and Standardization:** AES encryption is widely recognized and standardized, ensuring compatibility and interoperability across different systems and platforms. This promotes the adoption of the proposed image steganography system and allows for seamless integration with existing security frameworks.

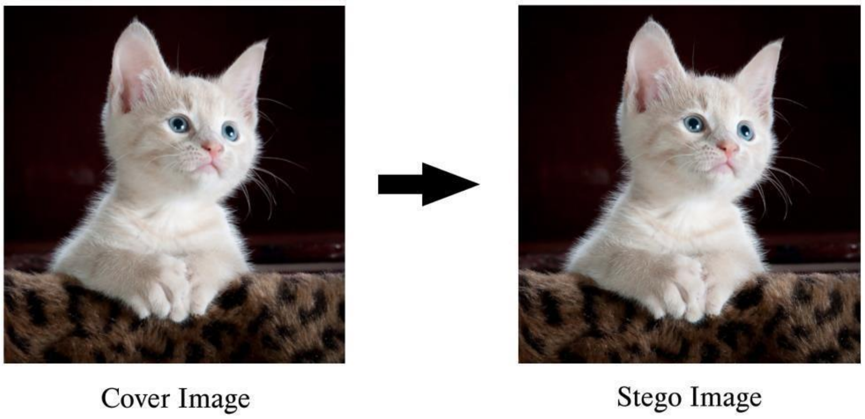


Fig 4.1 Image to Stego-Image

CHAPTER 5

WORKING OF PROPOSED METHOD

5.1 ENCRYPTION OF SECRET MESSAGE USING AES

The encryption module of image steganography using AES encryption is a crucial component of the system that ensures the confidentiality and integrity of the hidden information. This module performs the encryption of the secret message using the AES algorithm before embedding it into the cover image.

The AES encryption process involves the following steps:

- **Key Generation:** A secure key is generated using a random number generator or derived from a user-defined passphrase. The key size is selected based on the desired level of security, such as AES-128, AES-192, or AES-256.
- **Padding and Block Cipher Mode:** The secret message is divided into fixed-size blocks to align with the AES block size (128 bits). Padding techniques, such as PKCS7, may be applied to ensure that the last block is properly filled.
- **Encryption:** Each block of the secret message is encrypted using the AES algorithm and the generated encryption key. AES encryption involves multiple rounds of transformations, including substitution, permutation, and bitwise operations, to ensure robust encryption.
- **Cipher Block Chaining (CBC):** To enhance security, the AES encryption process may incorporate the Cipher Block Chaining mode. In CBC mode, each block is XORed with the previous ciphertext block before encryption, ensuring that even identical plaintext blocks result in different ciphertext blocks.
- **Output:** The encrypted blocks are concatenated to form the encrypted message, ready to be embedded into the cover image.

5.2 EMBEDING OF ENCRYPTED MESSAGE INTO COVER IMAGE

After the secret message is encrypted using AES, the next step in the image steganography process is to embed the encrypted message into the least significant bits (LSBs) of the cover image. This embedding process involves the following steps:

- **Pixel Selection:** The cover image is divided into individual pixels, each consisting of color components (e.g., RGB for color images). Pixels are selected sequentially or using a predetermined pattern to ensure even distribution of the embedded message throughout the image.
- **LSB Replacement:** For each selected pixel, the LSBs of the color components (or grayscale value) are modified to represent the bits of the encrypted message. The LSBs are replaced with the corresponding bits of the encrypted message, one bit at a time.
- **Message Length Indication:** To ensure proper extraction of the hidden message, a message length indication technique may be employed. This involves allocating a few LSBs of the cover image to indicate the length of the embedded message, allowing the extraction process to determine the boundaries of the hidden data.
- **Iterative Process:** The embedding process continues until all bits of the encrypted message, including any required padding, are embedded into the LSBs of the cover image. This iterative process ensures that the hidden message is evenly distributed throughout the image, minimizing visual distortions or artifacts.
- **Image Stego-Image Generation:** The resulting image, known as the stego-image, is the modified cover image with the encrypted message embedded within its LSBs. The stego-image appears visually similar to the original cover image, making it challenging for unauthorized observers to detect the presence of the hidden message.

5.3 EXTRACTION OF ENCRYPTED MESSAGE FROM THE STEGO-IMAGE

The extraction of the encrypted secret message from the stego image in image steganography using AES encryption involves the following steps:

- **Stego-Image Selection:** Choose the stego image from which you want to extract the hidden message.
- **Pixel Selection:** Similar to the embedding process, select pixels in a predetermined pattern or sequentially from the stego image.
- **LSB Extraction:** For each selected pixel, extract the LSBs of the color components (or grayscale value) to retrieve the bits of the encrypted message. Collect the LSBs from each pixel, one bit at a time, and concatenate them to reconstruct the encrypted message.

- **Message Length Extraction:** If a message length indication technique was employed during embedding, extract the LSBs allocated for indicating the length of the hidden message. Use this information to determine the boundaries of the encrypted message.
- **Decryption:** Apply the AES decryption algorithm using the appropriate decryption key to decrypt the extracted encrypted message. This process will restore the original secret message.
- **Message Reconstruction:** Once the encrypted message is decrypted, any padding that was added during encryption can be removed.

5.4 DECRYPTION OF ENCRYPTED MESSAGE USING AES

The decryption of an encrypted message using the AES algorithm involves the following steps:

- **Key Preparation:** Ensure that you have the correct decryption key that corresponds to the encryption key used to encrypt the message. The key should be securely stored and accessible for the decryption process.
- **Cipher Initialization:** Initialize the AES cipher in decryption mode using the decryption key. The key size should match the encryption key size used during encryption (e.g., 128 bits, 192 bits, or 256 bits).
- **Cipher Block Chaining (CBC):** If the encryption process used the CBC mode, ensure that the initialization vector (IV) used during encryption is available. The IV is required for decrypting the first ciphertext block.
- **Decryption:** Process the encrypted message in blocks of the appropriate size (e.g., 128 bits). Apply the AES decryption algorithm to each block using the decryption key and the previous ciphertext block (or IV in the case of the first block). This process involves multiple rounds of decryption, including substitution, permutation, and bitwise operations.
- **Padding Removal:** If padding was added during encryption, remove it from the decrypted message. Padding schemes like PKCS7 ensure that the original message length is preserved and can be easily identified for removal.

- **Message Reconstruction:** Concatenate the decrypted blocks, excluding any removed padding, to reconstruct the original plaintext message.

5.5 BLOCK DIAGRAM

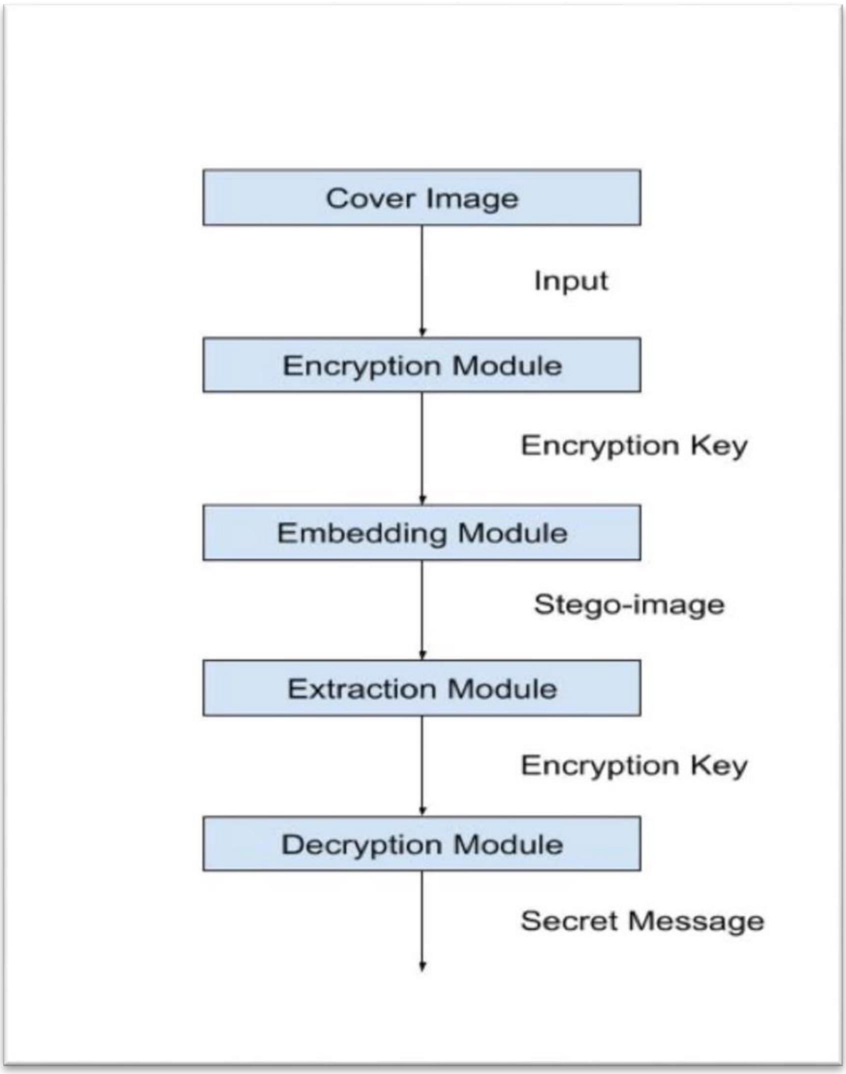
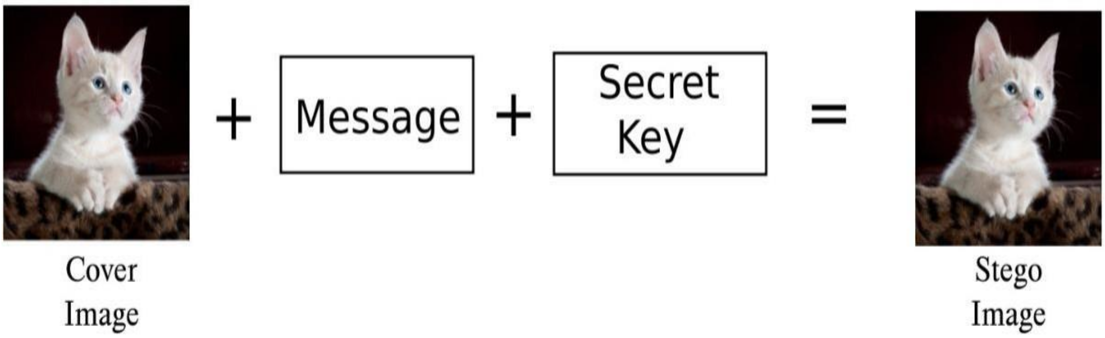


Fig 5.5 Block Diagram

5.6 PROPOSED SYSTEM DESIGN



5.6 Proposed System Design

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementing image steganography using the AES algorithm involves a combination of image processing techniques and AES encryption/decryption. Here is a high-level overview of the implementation steps:

- **Image Preparation:** Select a cover image and convert it to a suitable format, such as grayscale or RGB, depending on the desired application. Ensure that the cover image and the encrypted message are appropriately sized and compatible.
- **AES Encryption:** Encrypt the secret message using the AES algorithm with a secure encryption key. Divide the message into fixed-size blocks, and apply padding if necessary. Perform AES encryption on each block using the encryption key.
- **Stego-Image Generation:** Iterate through the cover image pixels, selecting them in a predetermined pattern. Replace the least significant bits (LSBs) of the cover image pixels with the encrypted message bits, ensuring imperceptibility. Incorporate techniques like message length indication to determine the boundaries of the hidden data.
- **Stego-Image Storage:** Save the modified cover image, now the stego-image, in a suitable format (e.g., JPEG, PNG) while preserving its quality and compatibility.
- **Stego-Image Extraction:** To extract the hidden message from the stego-image, follow the reverse process. Select pixels in the same pattern used during embedding. Extract the LSBs of the pixels and concatenate them to obtain the encrypted message. Use the correct decryption key and AES decryption algorithm to decrypt the message, removing any padding.
- **Message Reconstruction:** Combine the decrypted message blocks to reconstruct the original plaintext message.

Implementation is the stage of the project, where the theoretical design is turned into a working system. At this stage the main workload, the greatest upheaval and the major impact on existing practices shift to user department.

If the implementation stage is not carefully planned and controlled, it can cause chaos. Thus, it can be considered to be the more crucial stage in achieving a successful new stage and in giving the user confidence that the system will work and be effective

The implementation stage is a system project in its own right. It involves careful planning. investigation of the current system and it's constraints on implementation, design of methods.

The implementation plan consists of the following steps:

- Testing the developed system with sampled data.
- Detection and correction of errors.
- Making necessary changes in the system.
- Training and involvement of the user personnel.
- Installation of Hardware and Software utilities.

6.2 DEVELOPMENT TOOLS

- WINDOWS 10/11
- PyCharm or Vscode

CHAPTER 7

TESTING

Testing is mainly done for rectifying the error from the program that is design for particular problem.

- Testing is a process of executing a program with the intent of finding an error
- A good test case is one that has a high probability of finding an as-yet undiscovered error.
- A successful test is one that uncovers an as-yet undiscovered error. If a testing is conducted successfully (according to the objectives stated previously) it will uncover error in the system.

Testing Principle:

Testing principles are fundamental guidelines that help ensure the effectiveness and reliability of software testing. These principles guide testers in designing and executing tests to identify defects and assess the quality of the software being tested. Here are some key testing principles:

- **Testing Early:** Start testing activities as early as possible in the software development lifecycle. Early testing helps identify and address issues at an early stage, reducing the cost and effort required to fix them later.
- **Testing Exhaustively is Impossible:** Complete and exhaustive testing of a complex system is practically impossible. Testing efforts should be focused on critical functionalities, high-risk areas, and scenarios that are most likely to uncover defects.
- **Defect Clustering:** The Pareto principle applies to software defects, where a small number of modules or components tend to contain the majority of defects. Focus testing efforts on those high-risk areas to maximize defect detection.
- **Bugs Follow Patterns:** Software defects often exhibit patterns. Understanding common defect patterns, such as boundary value errors or input validation issues, can help testers design targeted tests to uncover such defects efficiently.
- **Test Automation:** Utilize test automation tools and frameworks to automate repetitive and labor-intensive testing tasks. Automation helps improve test coverage, efficiency, and consistency while reducing human error.

5.1 TESTING AND DEBUGGING

Testing is the process of systematically evaluating software or a system to identify defects, validate functionality, and assess its quality. It involves designing and executing tests, comparing actual results with expected results, and reporting any discrepancies. Testing aims to ensure that software meets the specified requirements and works as intended.

Debugging, on the other hand, is the process of identifying, isolating, and fixing defects or issues found during testing or in the operational phase. It involves analyzing the root cause of the problem, tracing its impact, and making necessary code modifications to resolve the issue and restore proper functionality. Debugging is an essential step in the software development lifecycle to enhance software reliability and performance.

5.2 FUNCTION TESTING

System design may have so many functions. Each program has been defined into number of functions. Each function has its own task. Each function to perform an accurate result We must debug each function. Function is a block of code that performs a particular task returns a particular value.

5.3 STRUCTURAL TESTING

Each program has a structure, and contains the function, variable, controls, statement decision-making loops. We can test program structure these are defined properly in our program. So, the programmer set the structure of the program

5.4 COMBINING STRUCTURAL AND FUNCTIONAL TESTING

After testing in our program function make the setup of the programs so that each function is run according to definition to the structure Program may have several structure and functions Programmer can arrange these method and structures. These properly perform Our task.

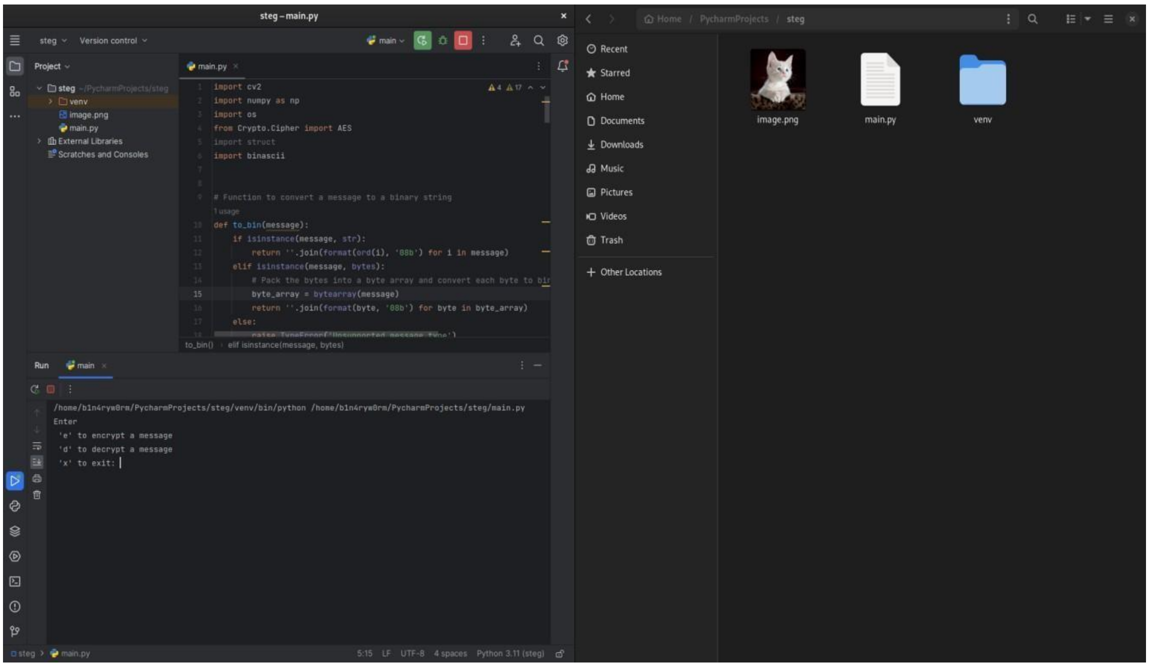
CHAPTER 8

EXPERIMENTATIONS AND RESULTS

Focused on enhancing the security of image steganography, a technique used to hide secret information within digital images. Steganography plays a crucial role in secure communication by providing a covert means of transmitting sensitive data. However, with the increasing sophistication of digital forensic tools, it has become essential to develop more robust and secure methods to protect the hidden information.

To achieve our goal, we proposed and implemented several enhancements to traditional image steganography algorithms. Firstly, we introduced a novel encryption scheme that encrypts the secret message before embedding it into the image. This encryption adds an extra layer of security by ensuring that even if the steganographic algorithm is compromised, the hidden message remains unreadable without the decryption key. We utilized a strong encryption algorithm, such as AES (Advanced Encryption Standard), to ensure the confidentiality of the secret message. Secondly, we incorporated a technique known as adaptive embedding, which dynamically adjusts the embedding capacity based on the characteristics of the host image.

8. SCREEN SHOTS

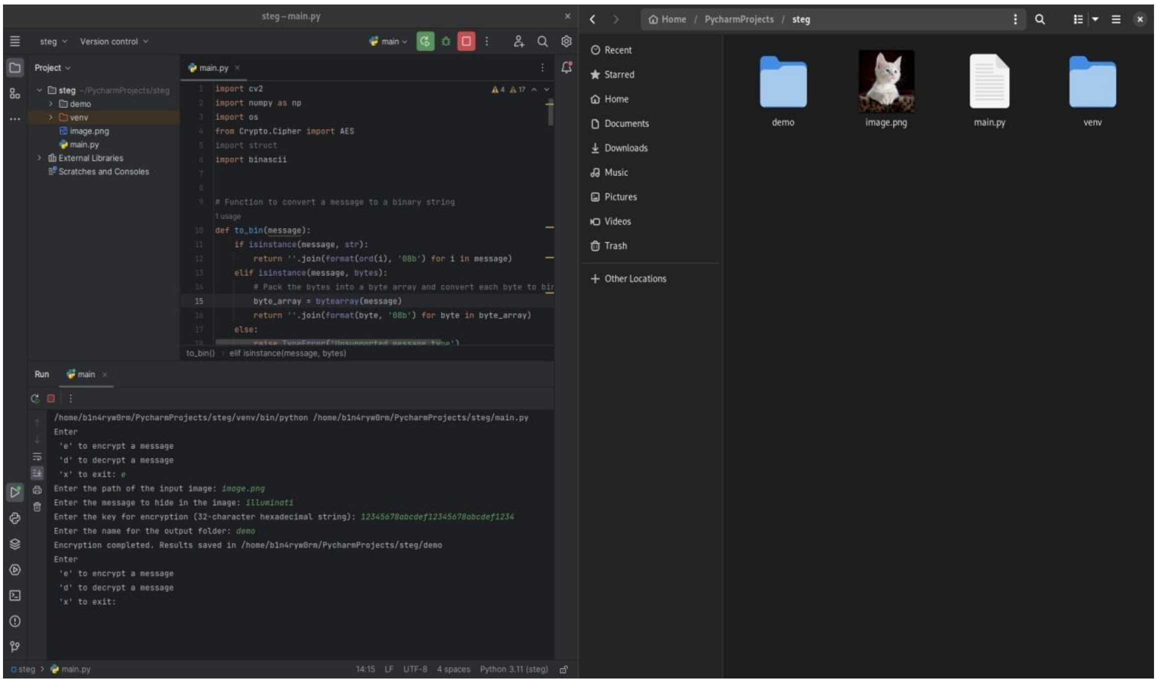


```python
import cv2
import numpy as np
import os
from Crypto.Cipher import AES
import struct
import binascii


# Function to convert a message to a binary string
def to_bin(message):
    if isinstance(message, str):
        return ''.join(format(ord(i), '08b') for i in message)
    elif isinstance(message, bytes):
        # Pack the bytes into a byte array and convert each byte to bi
        byte_array = bytearray(message)
        return ''.join(format(byte, '08b') for byte in byte_array)
    else:
        raise TypeError('Unsupported message type')
```

```
/home/b1n4ryw0rm/PycharmProjects/steg/venv/bin/python /home/b1n4ryw0rm/PycharmProjects/steg/main.py
Enter
 'e' to encrypt a message
 'd' to decrypt a message
 'x' to exit:
```

Fig 8.1 Before Encryption

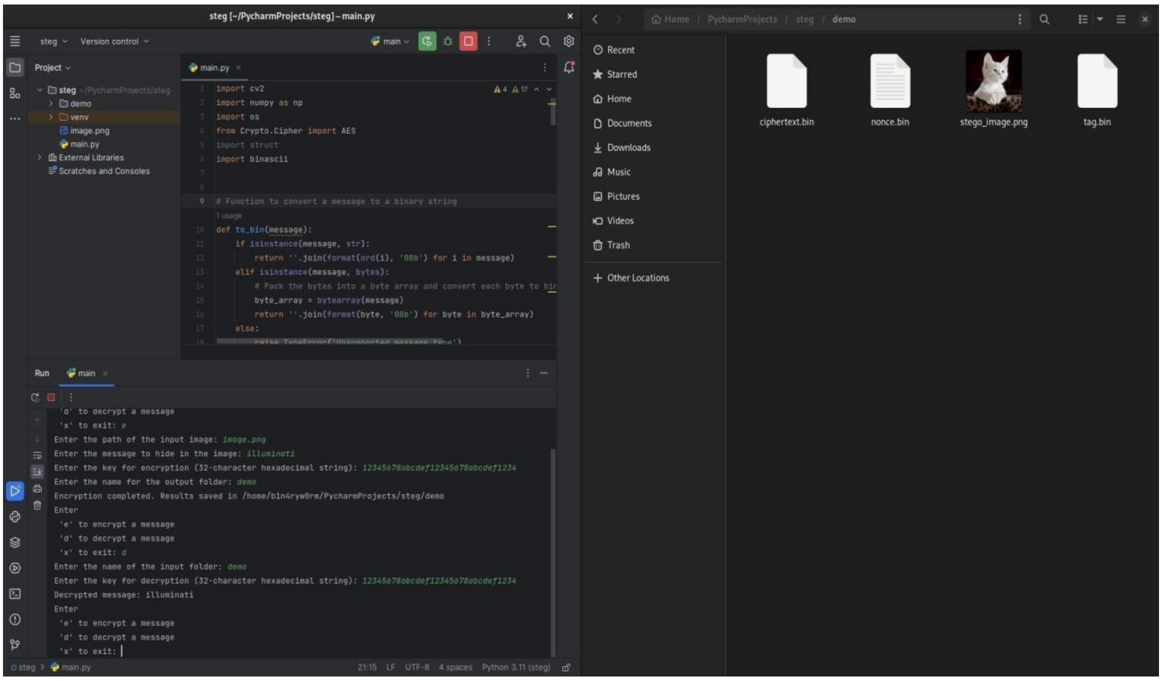


```
import cv2
import numpy as np
import os
from Crypto.Cipher import AES
import struct
import binascii


# Function to convert a message to a binary string
def to_bin(message):
    if isinstance(message, str):
        return ''.join(format(ord(i), '08b') for i in message)
    elif isinstance(message, bytes):
        # Pack the bytes into a byte array and convert each byte to bin
        byte_array = bytearray(message)
        return ''.join(format(byte, '08b') for byte in byte_array)
    else:
        raise TypeError('Unsupported message type')
```

```
/home/b1n4ryw0rm/PycharmProjects/steg/venv/bin/python /home/b1n4ryw0rm/PycharmProjects/steg/main.py
Enter
 'e' to encrypt a message
 'd' to decrypt a message
 'x' to exit: e
Enter the path of the input image: image.png
Enter the message to hide in the image: illuminati
Enter the key for encryption (32-character hexadecimal string): 12345678abcdef12345678abcdef1234
Enter the name for the output folder: demo
Encryption completed. Results saved in /home/b1n4ryw0rm/PycharmProjects/steg/demo
Enter
 'e' to encrypt a message
 'd' to decrypt a message
 'x' to exit:
```

Fig 8.2 After Encryption



```
import cv2
import numpy as np
import os
from Crypto.Cipher import AES
import struct
import binascii


# Function to convert a message to a binary string
def to_bin(message):
    if isinstance(message, str):
        return ''.join(format(ord(i), '08b') for i in message)
    elif isinstance(message, bytes):
        # Pack the bytes into a byte array and convert each byte to bin
        byte_array = bytearray(message)
        return ''.join(format(byte, '08b') for byte in byte_array)
    else:
        raise TypeError('Unsupported message type')
```

```
 'd' to decrypt a message
 'x' to exit: e
Enter the path of the input image: image.png
Enter the message to hide in the image: illuminati
Enter the key for encryption (32-character hexadecimal string): 12345678abcdef12345678abcdef1234
Enter the name for the output folder: demo
Encryption completed. Results saved in /home/b1n4ryw0rm/PycharmProjects/steg/demo
Enter
 'e' to encrypt a message
 'd' to decrypt a message
 'x' to exit: d
Enter the name of the input folder: demo
Enter the key for decryption (32-character hexadecimal string): 12345678abcdef12345678abcdef1234
Decrypted message: illuminati
Enter
 'e' to encrypt a message
 'd' to decrypt a message
 'x' to exit:
```

Fig 8.3 After Decryption

CHAPTER 9

CONCLUSION AND RECOMMENDATIONS

In conclusion, image steganography using AES provides a powerful and secure approach for hiding sensitive information within digital images. By combining the robust encryption capabilities of AES with the imperceptibility of steganography techniques, this method offers enhanced security and confidentiality for hidden messages. The integration of AES ensures that only authorized recipients with the correct decryption key can access and decipher the hidden information, while the imperceptible nature of the embedded message helps maintain the visual integrity of the cover image. The advantages of this approach include improved security, imperceptibility, resistance to attacks, versatility, and compliance with standardized encryption algorithms. Image steganography using AES encryption has practical applications in secure communication, data transmission, and digital watermarking etc.

REFERENCES

[1] S. M. Masud Karim, M. S. Rahman and M. I. Hossain, "A new approach for LSB based image steganography using secret key," 14th International Conference on Computer and Information Technology (ICCIT 2011), Dhaka, 2011, pp. 286-291.

[2] Osunade, O. and Isau Aremu Ganiyu. “Enhancing the Least Significant Bit (LSB) Algorithm for Steganography.” (2016).

[3] G. Krishnan S. and D. Loganathan, "Color image cryptography scheme based on visual cryptography," 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies, Thuckafay, 2011, pp. 404-407.

[4] Rani, M. M. S., & Rosemary Euphrasia, K. (2016). Data Security Through QR Code Encryption and Steganography. Advanced Computing: An International Journal (ACIJ), 7(1/2), 1-7.

[5] M. Naor and A. Shamir, "Visual cryptography," in Advances in Cryptology - EUROCRYPT'94, A. D. Santis., Ed., vol. 950. Springer Verlag, 1995, pp. 1-12.

APPENDIX

main.py

```
import cv2
import numpy as np
import os
from Crypto.Cipher import AES
import struct
import binascii

# Function to convert a message to a binary string
def to_bin(message):
if isinstance(message, str):
return ''.join(format(ord(i), '08b') for i in message)
elif isinstance(message, bytes):
# Pack the bytes into a byte array and convert each byte to binary
byte_array = bytearray(message)
return ''.join(format(byte, '08b') for byte in byte_array)
else:
raise TypeError('Unsupported message type')

# Function to convert a binary string to a message
def to_message(binary):
message = ''
for i in range(0, len(binary), 8):
byte = binary[i:i + 8]
message += chr(int(byte, 2))
return message

# Function to encrypt a message using AES in GCM mode
def encrypt(key, nonce, message):
cipher = AES.new(key, AES.MODE_GCM, nonce=nonce)
ciphertext, tag = cipher.encrypt_and_digest(message)
return ciphertext, tag, cipher.nonce

# Function to decrypt a ciphertext using AES in GCM mode
def decrypt(key, nonce, ciphertext, tag):
cipher = AES.new(key, AES.MODE_GCM, nonce=nonce)
message = cipher.decrypt_and_verify(ciphertext, tag)
return message

# Function to hide a message in the least significant bit of an image
def hide_message(image, message):
binary = to_bin(message)
binary += '0' * ((len(image.flatten()) * 3) - len(binary)) # Padding
binary = np.array(list(binary), dtype=int)
binary = binary.flatten()[:image.shape[0] * image.shape[1] * 3]
binary = binary.reshape(image.shape[0], image.shape[1], 3)
stego_image = image.copy()
for i in range(stego_image.shape[0]):
```

```
for j in range(stego_image.shape[1]):
for k in range(3):
if binary[i][j][k] == 0 and stego_image[i][j][k] % 2 == 1:
stego_image[i][j][k] -= 1
elif binary[i][j][k] == 1 and stego_image[i][j][k] % 2 == 0:
stego_image[i][j][k] += 1
return stego_image

# Function to extract a message hidden in the least significant bit of an image
def extract_message(stego_image):
binary = ''
for i in range(stego_image.shape[0]):
for j in range(stego_image.shape[1]):
for k in range(3):
binary += str(stego_image[i][j][k] % 2)
binary = binary.rstrip('0') # Remove padding
message = to_message(binary)
return message

# Get user input for encryption or decryption
while True:
mode = input("Enter\n 'e' to encrypt a message \n 'd' to decrypt a message \n 'x' to exit: ")
if mode == 'e':
# Load the input image
image_path = input('Enter the path of the input image: ')
image = cv2.imread(image_path)

# Get the message and encryption key from the user
message = input("Enter the message to hide in the image: ")
key_hex = input("Enter the key for encryption (32-character hexadecimal string): ")

# Convert the key from hexadecimal to bytes
key = binascii.unhexlify(key_hex)

nonce = os.urandom(16)
ciphertext, tag, nonce = encrypt(key, nonce, message.encode())
stego_image = hide_message(image, ciphertext)

# Get the folder name from the user
output_folder_name = input("Enter the name for the output folder: ")
output_folder_path = os.path.join(os.getcwd(), output_folder_name)
os.makedirs(output_folder_path, exist_ok=True)

# Save the modified image and the ciphertext, nonce, and tag in the output folder
stego_image_path = os.path.join(output_folder_path, 'stego_image.png')
cv2.imwrite(stego_image_path, stego_image)

ciphertext_path = os.path.join(output_folder_path, 'ciphertext.bin')
with open(ciphertext_path, 'wb') as f:
f.write(ciphertext)

nonce_path = os.path.join(output_folder_path, 'nonce.bin')
```

```
with open(nonce_path, 'wb') as f:
f.write(nonce)

tag_path = os.path.join(output_folder_path, 'tag.bin')
with open(tag_path, 'wb') as f:
f.write(tag)

print("Encryption completed. Results saved in", output_folder_path)

elif mode == 'd':
# Get the folder name from the user
input_folder_name = input("Enter the name of the input folder: ")
input_folder_path = os.path.join(os.getcwd(), input_folder_name)

# Load the stego image and the ciphertext, nonce, and tag from the input folder
stego_image_path = os.path.join(input_folder_path, 'stego_image.png')
stego_image = cv2.imread(stego_image_path)

ciphertext_path = os.path.join(input_folder_path, 'ciphertext.bin')
with open(ciphertext_path, 'rb') as f:
ciphertext = f.read()

nonce_path = os.path.join(input_folder_path, 'nonce.bin')
with open(nonce_path, 'rb') as f:
nonce = f.read()

tag_path = os.path.join(input_folder_path, 'tag.bin')
with open(tag_path, 'rb') as f:
tag = f.read()

# Get the decryption key from the user
key_hex = input("Enter the key for decryption (32-character hexadecimal string): ")

# Convert the key from hexadecimal to bytes
key = binascii.unhexlify(key_hex)

decrypted_message = decrypt(key, nonce, ciphertext, tag)

# Print the extracted and decrypted message
extracted_message = extract_message(stego_image)
print('Decrypted message:', decrypted_message.decode())

elif mode == 'x':
exit()
```