

A Mini Project with Seminar On
Image Steganography Using AES-256, RSA and Randomized LSB
Submitted in partial fulfillment of the requirements for the award of the
Bachelor of Technology

In
Department of Computer Science and Engineering
(Artificial Intelligence and Machine Learning)

By
A. Priya Krishna **21241A66D1**
K. Harshitha **21241A66F9**
M. Harini Reddy **21241A66G5**

Under the esteemed guidance of

Ms. V. Manasa
Assistant Professor



Department of Computer Science and Engineering
(Artificial Intelligence and Machine Learning)
GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND
TECHNOLOGY

(Approved by AICTE, Autonomous under JNTUH, Hyderabad)
Bachupally, Kukatpally, Hyderabad-500090



**GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND
TECHNOLOGY**

(Autonomous)

Hyderabad-500090

CERTIFICATE

This is to certify that the mini project entitled “**Image Steganography Using AES-256, RSA and Randomized LSB**” is submitted by **A. Priya Krishna(21241A66D1), K. Harshitha (21241A66F9), M. Harini Reddy (21241A66G5)** in partial fulfillment of the award of degree in BACHELOR OF TECHNOLOGY in Computer Science and Engineering (Artificial Intelligence and Machine Learning) during Academic year 2024- 2025.

Internal Guide

Ms. V. Manasa

Head of the Department

Dr. G. Karuna

External Examiner

ACKNOWLEDGEMENT

There are many people who helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all. First, we would like to express our deep gratitude towards our internal guide **Ms. V. Manasa**, Assistant Professor, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), for her support in the completion of our dissertation. We wish to express our sincere thanks to **Dr. G. Karuna**, Head of the Department, and to our Principal **Dr. J. Praveen**, for providing the facilities to complete the dissertation. We are thankful to mini project coordinator **Ms M. Shamila**, Assistant Professor, for her valuable suggestions and comments during this project period. We would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

A. Priya Krishna (21241A66D1)

K. Harshitha (21241A66F9)

M. Harini Reddy (21241A66G5)

DECLARATION

We hereby declare that the mini project titled “**Image Steganography using AES-256, RSA and Randomized LSB**” is the work done during the period from 6th Feb 2024 to 29th June 2024 and is submitted in the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence and Machine Learning) from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad). The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

A. Priya Krishna(21241A66D1)

K. Harshitha(21241A66F9)

M. Harini Reddy(21241A66G5)

ABSTRACT

The objective of this project is to securely hide confidential information within images by the integration of the AES-256, RSA, and Randomized LSB techniques to produce a resilient model. Confidential information gets dual security through the encryption process because of the AES and RSA. Confidential information gets dual security through the encryption process because of the AES and RSA. It provides dual-layer encryption of data. It increases the hiding capacity of the data. It adds randomness to the image the text can be distributed anywhere in the image. It also leverages the advantage of the random generator; the encrypted data gets distributed randomly inside the image making the image imperceptible to the human eyes and protecting the confidential information from unauthorized access. It helps protect the visual quality of the image throughout the process. The combination of these three layers adds dual-layer security to the image. It secures the image data by making it resistant to steganalysis attacks due to the addition of randomization to the process it is difficult to detect the patterns.

LIST OF FIGURES

Figure No.	Figure Name	Page No.
1	Asymmetric Encryption with RSA	4
2	Symmetric Encryption with AES	5
3	Cipher Block Chaining	6
4	Image Steganography	7
5	Data Compression	8
6	Block Diagram	9
7	Architecture Diagram	31
8	Modules-Connectivity Diagram	32
9	Encryption Module	36
10	Image Encoder Module	38
11	Image Decoder Module	40
12	Decryption Module	42
13	UML Class Diagram	44
14	UML Sequence Diagram for Encoding	45
15	UML Sequence Diagram for Extracting	46
16	UML Use Case Diagram	47
17	UML Activity Diagram	48
18	Sand image	52

LIST OF TABLES

Table No.	Table Name	Page No.
1	Summary of Existing Approaches	18
2	Result	53

LIST OF ACRONYMS

RSA	Rivest, Shamir, Adleman
AES	Advanced Encryption Standard
LSB	Least Significant Bit
PSNR	Peak Signal-to-Noise Ratio
CBC	Cipher Block Chaining

TABLE OF CONTENTS

Chapter No.	Chapter Name	Page No.
	Certificate	ii
	Acknowledgement	iii
	Declaration	iv
	Abstract	v
	List of Figures	vi
	List of Tables	vii
	List of Acronyms	viii
1	Introduction	
	1.1 Introduction to Project Work	1
	1.2 Objective of the Project	3
	1.3 Methodology Adopted	3
	1.4 Block Diagram	9
	1.5 Organization of the Report	11
2	Literature Survey	
	2.1 Summary of Existing Approaches	12
	2.2 Summary: Drawbacks of Existing Approaches	26
3	Proposed Method	
	3.1 Problem Statement and Objective of the project	28
	3.2 Architecture Diagram	31
	3.3 Module Connectivity Diagram	32
	3.4 Software and Hardware Requirements	34
	3.5 Modules and their Description	35
	3.6 Requirements Engineering	42
	3.7 Analysis and Design through UML	43
	3.8 Testing	

4	Results and Discussions	
	4.1 Detailed Explanation of the Experimentation Results	49
	4.2 Significance of the proposed method with its advantages	52
5	Conclusion and Future Enhancements	55
6	Appendices	57
	References	61

CHAPTER 1

INTRODUCTION

The following section describes the significance of image steganography which holds immense importance in covert data transmission within images while maintaining visual integrity.

1.1. Introduction to project work

Previously, in the olden days for the secure transfer of secret information techniques like microdots, invisible ink, and other cryptographic techniques were used. Invisible ink helps to protect secret information by writing the information using certain things that allow us to view the information under some conditions, microdots use condensed tiny drops that contain secret information. Encryption techniques like classical cipher and other cryptographic techniques were used to encode the secret information before sending it to the receiver. These techniques transmitted the secret information in the physical form but in the modern steganographic techniques data is sent through images.

In recent times, Steganographic processes use Images as a cover medium to transfer secret information, because of their large capacity and security. Initially, the secret information is embedded in the LSB bits of pixels of the cover image. These primary techniques were only used for experimental purposes after these techniques were enhanced further and new techniques were introduced. Image Steganographic methods were further improved using advanced algorithms and clever data-hiding strategies. In the present day, steganographic techniques take the help of the additional information present in the image, which makes the embedding and retrieving process easy.

Techniques like embedding data in different image spaces (like where the pixels are) or frequency-based embedding (using methods such as Discrete Cosine Transform – DCT and Least Significant Bits - LSB) have significantly improved how much data can be hidden and the security of hiding it. Moreover, Steganographic techniques after being combined with encryption methods made the transmission process robust in modern communication. The hybrid model offers an efficient transfer of information, increases the security of the data, and is robust against security attacks making the model more versatile.

AES-256 is a symmetric encryption, means for both encryption and decryption only a single key is required. It is renowned for being strong and resistant to direct assaults. It operates on data blocks with a size of 128 bits and has a 256-bit key. This encryption techniques add dual layer of security to the data even if the data in the image gets detected, the data can only be accessed through decryption key. In the encryption process AES key transforms plain text into cipher text. The encryption process repeats for 14 rounds. AES can be used in various modes of operation i.e, EBC, CBC and CFB. AES's Security, Speed and performance makes as the strongest encryption technique many security systems leverages it's capabilities to strengthen the security system.

RSA (Rivest-Shamir-Adleman) is a symmetric encryption technique where two keys are used in the process of encryption and decryption of the text. Initially, two keys are generated one is a public key and other is a private key. The public key is used to encrypt the AES key generated from AES encryption technique. After the key is encrypted then the generated private key is used for decryption of the AES key which is encrypted by the corresponding public key. RSA Algorithm ensures secure transmission of AES between Sender and Receiver. RSA provides strength to the encryption system. It is a robust technique. It is very important in the modern data communication. It uses various mathematical for the generation of these keys. It is used for digital certifications.

To address all key considerations, the proposed system focuses on hiding secret messages in images using a mix of hiding patterns and strong encryption. First, we lock up the message using complex codes (RSA and AES-256). Then carefully this message is locked into specific parts of the image's hidden details. It helps protect the secret message and keeps it hidden inside the host image. The encryption process involves the encryption of the secret message using the AES algorithm and then spreading the secret message into the image using a random pattern making it imperceptible to the human eyes. This similar random pattern is used in the case of decryption to decrypt the hidden message which is encrypted using keys. This combined information-hiding process includes the safe retrieval of information and the security of messages.

1.2. Objective of the Project

- To integrate AES-256 encryption for heightened security.
- To ensure imperceptible changes to the host image post-data embedding.
- To implement seeded LSB embedding with a novel pattern for enhanced data hiding in images.
- To encryption of session key for enhanced security.
- To incorporate customization.
- To Optimize Performance.

1.3. Methodology Adopted

In this section the following information is presented:

- Asymmetric Encryption with RSA.
- Symmetric Encryption with AES
- Cipher block chaining.
- LSB Steganography with randomization.
- Data Compression.

1.3.1 Asymmetric Encryption with RSA:

Below Figure-1 represents asymmetric encryption. Asymmetric encryption is a public key cryptography that consists of two keys for the encryption and decryption process. One of them is public which is used in the encryption process and the other is a private key which is used in the decryption process. public can be shared with anyone but only the owner contains the private key. Anyone who contains the public key can encrypt the data but only the owner can decrypt the text. private key is used to decrypt the data that is encrypted by the corresponding public key.

RSA is a type of public key cryptography where it contains key pair consisting of public and private keys. RSA key generation depends on factoring the two large prime numbers. RSA private key is used to encrypt the normal plain text to the cipher text in the encryption process. These are generated in such a way that only the private key is used to decrypt data encrypted by the corresponding public key. It is used to generate this pair of cryptographic keys, to ensure secured encryption and decryption of the session key and they are serialized (storable or transmittable format) to make sure that they are securely saved and

loaded for encryption and decryption operations. These keys are very important in the process of encrypting the session and symmetric keys. It provides a double-layer security to the encryption process even if the keys are compromised the access of the keys is challenging for the intruders without knowing the passphrase.

Steps involved:

- Choose 2 different large random prime numbers P and Q.

- Calculate $n = p * q$.

- Calculate $\phi(n) = (p - 1) * (q - 1)$

- Choose 'e' such that $1 < e < \phi(n)$

'e' is co-prime to $\phi(n)$

$\text{Gcd}(e, \phi(n)) = 1$

E -> public key

- Calculate d, such that $de = 1 \text{ mod } \phi(n)$

i.e, $d * e \text{ mod } \phi(n) = 1$

d -> private key

$de = 1 + k \phi(n), k = 0, 1, 2$

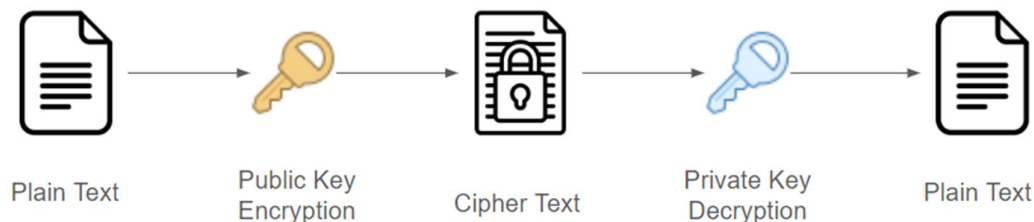


Figure-1 Asymmetric Encryption with RSA

1.3.2 Symmetric Encryption with AES:

Below Figure-2 represents symmetric encryption. The Advanced Encryption Standard (AES) plays a key role in contemporary cryptography due to its performance and robustness. AES works on a set block size of 128 bits. It divides the text into equal block sizes of 128 bits. It uses key sizes between 128-256 bits. AES uses the symmetric encryption method, which permits the sender and receiver to possess the same mystery key for encryption and decryption procedures, making sure of a secure and safe communicate.

Once the blocks are encrypted, they may be joined collectively to form the cipher block. A substitution-permutation community is hired, which includes a series of interlinked operations in which precise outputs update inputs, and bits are shuffled. Various complex changes are concerned within the encryption method, which includes sub-bytes, shift rows, combination columns, and upload-round keys, which might be numerous based on the critical element length, therefore increasing the safety of the encrypted facts. AES is well-known for its superior protection, robustness in competition to cryptanalysis, and complicated design and key space.

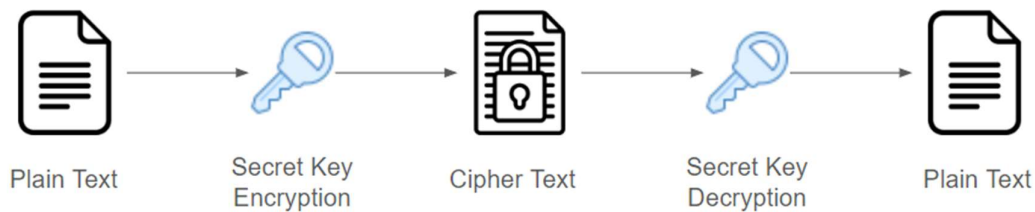


Figure-2 Symmetric Encryption with AES

AES encryption is a Symmetric encryption where a single is used for the encryption and decryption processes. First, a session key is generated then a symmetric key is derived from it. The session key is encrypted using RSA keys. The same session key is used in the decryption process. The decryption process is the reverse of the encryption process. AES provides a dual-layer security for the secret text. AES is performed before embedding the secret text into the carrier medium in steganography.

1.3.3 Cipher Block Chaining:

Below Figure-3 represents Cipher Block Chaining. In Cipher block Chaining previous cipher text block is XOR-ed with the next plain text block an IV is used to start the process and chaining aids the initialization vector. The weight of the particular block depends on its previous blocks. Any cryptography error in one of the cipher text's blocks affects all of the blocks for decryption. The change in the order of the ciphertext disturbs the decryption process. The input of the encryption algorithm is XOR of the current plain text block and the preceding ciphertext block. So, the repeating patterns are not exposed. The same key is used for encryption and decryption. The initialization vector containing a data block of the same size is used in 1st round of encryption and 1st round of decryption. To get the identical ciphertext

block for the same plaintext block, you must use the equal encryption key and IV, and the ciphertext ought to stay within the same order.

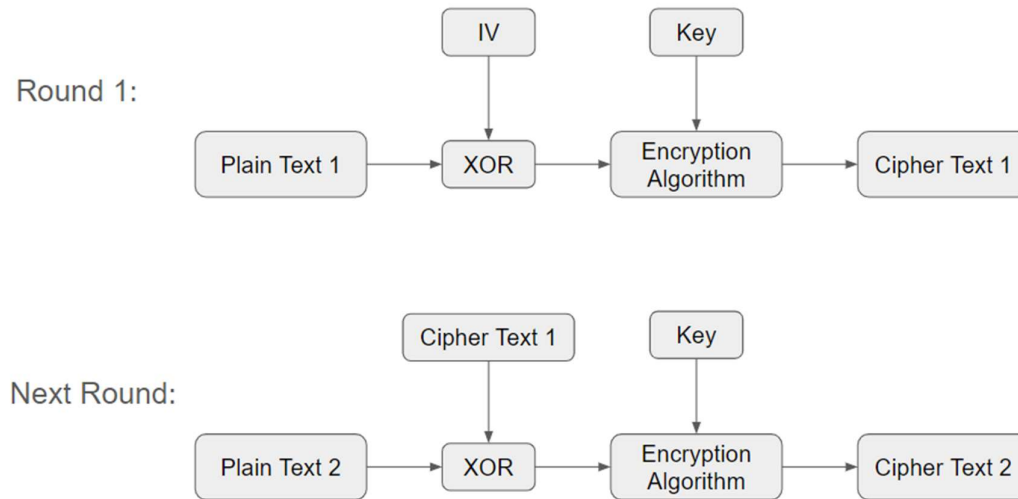


Figure-3 Cipher Block Chaining

If a single encrypted key is used for encrypting two data transmissions, then the initialization vector used for both transmissions should be different. Initialization vectors used for communications should be unique and there is no need to hide these initialization vectors. Cipher block used in this process is generated by performing the XOR of the previous ciphertext and current plain text.

1.3.4 LSB Steganography with Randomization:

Below Figure-4 represents LSB Steganography. Randomized LSB is a method used in Steganography and Cryptography for hiding data inside the LSB bits of the image pixels in random order to improve security and privacy. It helps in changing the LSB bits in the unpredictable and pseudo-random order, this method is often used in digital media, i.e. images, videos, and text. This technique is vulnerable to steganalysis attacks. Steganography using Randomized LSB makes it a powerful tool, it helps in altering the LSB bits of the pixel image in pseudo-random order making the detection of the information difficult through statistical analysis.

In the Steganographic approach, the pixel values in the cover images spread over the entire image randomly to prevent the chance of detection by an attacker. Secret Message bits

are embedded into the host image at distinct pixel locations, instead of placing them sequentially. After the secret message is embedded into the host image, the host image becomes a Stego image. It looks similar to the original image without significant distortion of the visual quality and is highly imperceptible. The method exhibits minimal susceptibility to statistical attacks like RS Analysis and Sample Pair Analysis.

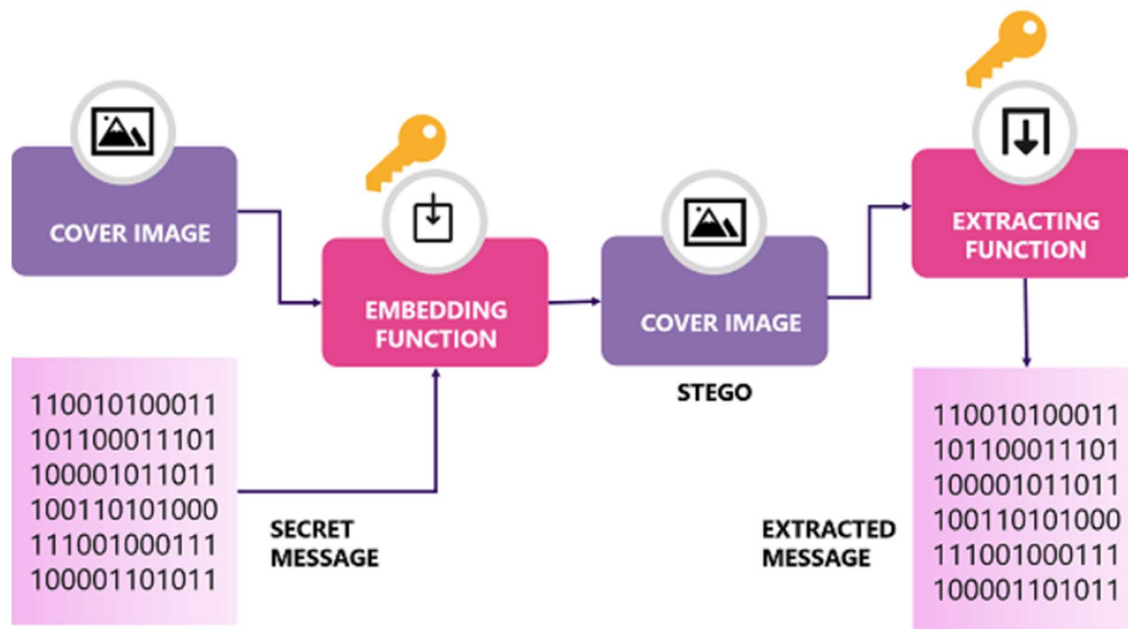


Figure-4 Image Steganography (Courtesy: Source [19])

Cryptographic algorithms get dual-layer security after the addition of the Randomized LSB. It is challenging for adversaries to decipher the hidden information as the data inside the carrier medium gets randomly distributed. It also helps maintain data integrity in cryptographic algorithms by making randomization changes. It can be used as error detection for the slightest changes done by an unauthorized person and can be identified by the consistency of the least significant bit (LSB) pattern. It also assists in privacy protection by safeguarding sensitive information in digital communication. It aids in concealing the identity of the intended user by incorporating the information in a random order in an untraceable manner.

Modifying the LSB bits in random order may lead to a reduction in the quality of the original information embedded in the image. If the randomization pattern gets comprised the security of the technique gets weakened. Randomization is an exceptional technique that finds its application in cryptography, watermarking, and steganography by improving security and robustness.

1.3.5 Data Compression:

Figure-5 represents how data compression works. Data Compression is a helpful tool that aids in reducing the size of the text that has to be transmitted. Before encryption, the data is compressed using Zlib to reduce its size. Zlib is a software tool, used for compression and decompression process. Zlib comes with a .zlib extension. The compression range is between 2:1 to 5:1. This not only makes the process more efficient but also helps in minimizing patterns that could be detected by steganalysis tools. Secret text that has to be embedded into the cover medium is compressed first and after it is embedded into the cover medium. Steganography uses a lossless compression technique to promote data integrity while restoring the data in the original file.

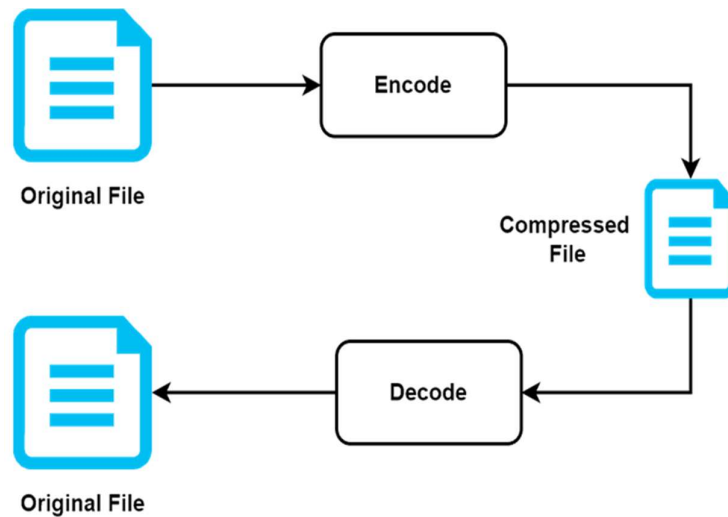


Figure-5 Data Compression(Courtesy: Source [20])

1.4 Block Diagram

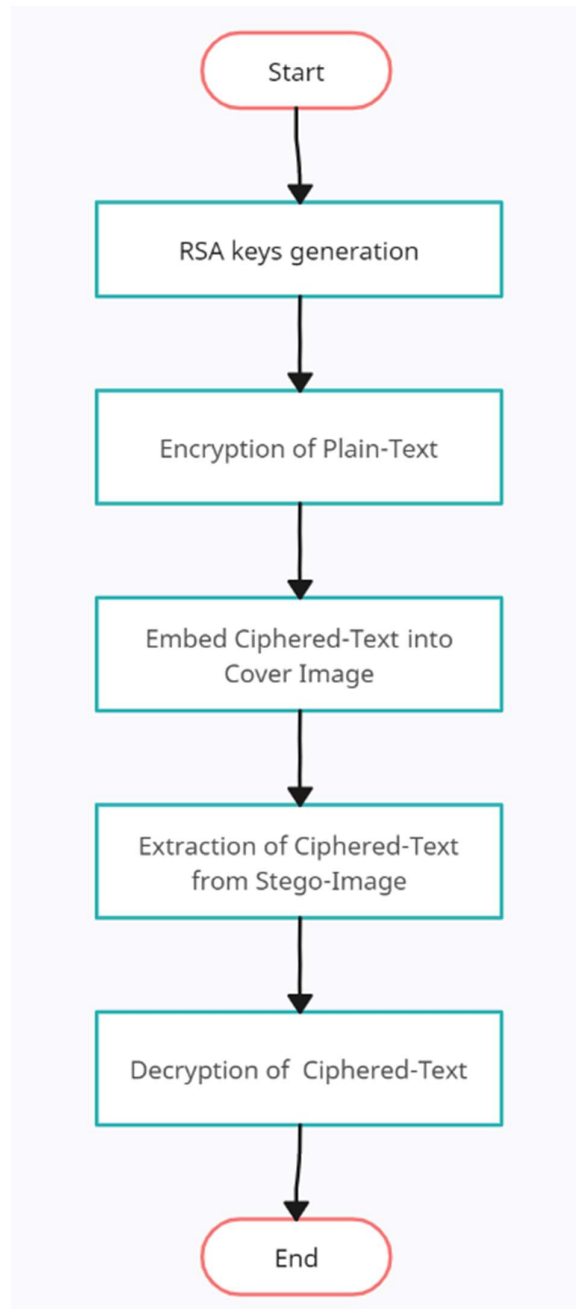


Figure-6 Block Diagram

Our project's block diagram is represented in Figure-6. The plain text and the cover image are given as the input to the model. RSA key generation takes place when two keys are generated. The two keys are the RSA public key and a private key. AES (Advanced Encryption Standard) is a symmetric algorithm that generates a single key for the encryption and decryption process. It generates a session key which is randomly generated using salt and the passphrase. From the session, a symmetric key is generated, and the generated symmetric key is used to encrypt the plaint to the cipher text. The session key is encrypted using the RSA public key. To get the bit positions where the text has to be embedded a pseudo-random generator is used to get the random sequence. Then the encrypted image is taken into the numpy array. The encrypted message along with the session key is embedded into the cover image.

1.5 Organization of the Report

The organization of this study report revolves around the following mentioned sections in depth. The study is organized into chapters of information which are presented in a logical sequence with understanding of the topic being explored to maintain cohesion. Each chapter focuses on a particular component of the work, and the information is presented in a clear and concise manner to ensure that the reader can easily follow the information being presented.

1.5.1 Introduction: A gentle introduction which explains the motivation behind this study is presented. Along with that, the objective of this study is clearly stated and the methodology to satisfy it, along with a brief description of the block diagram.

1.5.2 Literature Survey: A literature survey is a comprehensive review of existing literature, research studies, and other sources of information related to a specific topic. Where relevant information from research articles, conference papers, and Journals was carefully analyzed with the aim of identifying common themes, trends, and insights related to image restoration techniques. A summary of the existing works, advantages, results, and drawbacks of the works have been outlined.

1.5.3 Proposed Method: It involves stating the problem statement and objectives of the project. The architecture diagram and modules are explained in-depth. To add software and hardware requirements were also specified.

1.5.4 Results and Discussions: After the implementation of the proposed method, results were analyzed subsequently, the discussion about the proposed method and results were discussed.

1.5.5 Conclusion and Future Enhancements: Based on the results obtained and through analysis, the conclusion is presented along with some future enhancements related to the proposed method.

1.1.6 Appendices: The source code of the project consisting of the main model architecture is attached in this section.

1.1.7 References: Proper referencing is given to the mentioned reference works to support claims and to provide the context for the proposed method to solve the problem identified.

CHAPTER 2

LITERATURE SURVEY

This chapter includes the description and summary of our project's current approaches, their advantages, results, and their shortcomings.

2.1 Summary of Existing Approaches

Abdelkader Moumen et al., [1] proposed a version that completed version for Image Encryption using the Steganographic LSB Method, AES, and RSA algorithm. Their approach leverages the rate of symmetric encryption with the safety of uneven encryption and LSB methods. It combines the AES, RSA, and LSB techniques via encrypting the name of the game message using the AES set of rules and encrypting the AES secret key using the RSA algorithm, and the encrypted picture is hidden within the cipher picture the usage of the LSB techniques. Experimental consequences show that the Asymmetric set of rules guarantees key security and resilience towards renowned assaults. This approach is designed to ensure the powerful transmission of touchy statistics throughout essential sectors.

Krishnakant Tiwari et al. [2] have advanced an Image Steganography technique that employs a Pixel Locator Sequence combined with AES encryption. The process makes use of PLS all through each encoding and decoding of the LSB. The experimental results have confirmed applicable values of Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR), indicating minimum modifications to the image and lower distortion. In this method, data bits are allotted pseudo-randomly across the picture pixels, making the interpreting method hard without understanding of the authentic pixel series. Additionally, the technique complements protection by using encrypting the name of the game photograph earlier than encoding, thereby increasing the general level of protection.

Esther Hannah M et al., [3] developed an Android-based approach for Data Communication Security using LSB and Password-based Encryption for Image Steganography. This model contains symmetric encryption, asymmetric encryption, and steganographic methods. In this approach, the key, data, and original image were given as input to the steganographic System, the picture was encrypted initially using a Symmetric technique. Subsequently, the encrypted image was embedded inside another image using the LSB Steganographic method, following the encryption of the secret key using an asymmetric

algorithm. To develop an Android-based application, to hide secret information inside the safe photographs., suggesting the stego image was highly indistinguishable from the original image. The Application provided a user-friendly interface. The work strengthened the data security, applied robust encryption techniques, and steganographic methodologies together with interface enhanced the data security.

Vikas Singhal et al.,[4] worked on Image Steganography combined with Advance Encryption Standard (AES) using SHA-256 for security. The proposed work used steganography to hide secret information and cryptography to secure textual data. It focused on securing the key in Advanced Encryption Standard by employing hashing. In this process, the key was encrypted via the hashing technique SHA 256. This involved the generation of the hash key using SHA-256, a hashing technique that helps in encrypting the data using AES-256. The experimental results depicted that there was not much difference between the original image and the stego image, there were no changes in the dimensions of the image. Confidentiality and covertness of the crucial data remained intact by inventing an impermeable system. AES encryption with SHA-256 hashing ensured strong data concealment and integrity verification for enhanced security.

Zheyi Zhang et al.,[5] proposed a Chaotic color multi-image compression-encryption/LSB data type steganography scheme for NFT transaction security. Chaotic sequences are taken from the chaotic map, at the same time as a couple of mystery snapshots are compressed by the usage of CS and fused into a bigger mystery picture. A large Secret Image is embedded into a couple of cover pics through steganography, hiding statistical alterations. These cover images sell type steganography by adding a texture to the 3D version. Increased key space and steganographic capabilities were demonstrated by the results of performance and stimulation. Improved security, hiding success facts, resilience to attacks, environmentally friendly processing, and true worldwide applicability. utilizing steganography, cutting-edge encryption, and anti-counterfeiting techniques to guarantee NFT transaction security. This model secures the NFT image transactions and connects 2D to 3D textures by offering new thoughts for steganography.

Kevin Wijaya et al.,[6] Introduced a Time-Based Steganography Image with Dynamic Encryption Key Generation. The above approach focused on improving the security levels of keys by integrating steganography and encryption. It uses two keys: a normal string and one based on the insertion time. The Insertion process produces the time key at the time of insertion.

Both these keys are combined to form another key namely the third key which is used for the encryption of the secret data. Security of the data transmission increases after combining these two keys. The encryption keys are generated dynamically using the Physical factors. Experimental results showed the PSNR values of 55-81 dB for various cover images. Encryption technique based on time controls the decryption process for secret data only to specific periods. Encryption and Decryption time takes place in less than a minute.

Pooja Jha et al., [7] Study has focused on the Analysis and Implementation of Image Steganography by using the AES algorithm. In the above methodology, the secret data is encrypted using the AES algorithm and incorporated into the digital photographs. The model combines LSB Steganography with the AES encryption method, the above model is developed using Python Programming Language. The secret information is encrypted using the AES encryption method and then embedded into the cover image using LSB Steganography. The experimental results stated that the cover image is imperceptible to the human eyes after subtracting the original image from the encrypted image the resulting image formed is black indicating not much difference between them making the image imperceptible to human eyes and helps in making it difficult for the attackers to retrieve the information. Speeds of the encryption and decryption processes were 0.264 Mbps and 0.0631 Mbps, respectively.

Oluwakemi Christiana Abikoye et al.,[8] presented a Study focused on the Analytical Study of the LSB-based Image Steganography Approach. The Study focused on improving the performance of the Least Significant Bit (LSB) Steganography by proposing a modified LSB called circular shift LSB. The Study exhibited the LSB Steganography technique and estimated the results by using multiple cover images like .png, .bmp, and .jpeg. The prospective research surpasses the past techniques by acquiring the PSNR of 84.46 and MSE of 0.0002124. .bmp image format is advisable in the above approach for secure data transmission due to its high PSNR and low MSE values. This indicates there wasn't much distinction between the original image and the encrypted image by using LSB Steganographic techniques. The circular shift LSB an outcome of the LSB technique outperformed the past approaches.

D. Arul Suresh et al., [9] proposed a new technique to safeguard medical images through a hybrid technique called the Stegano-Crypto technique. This approach provides double-layer security by combining steganography and cryptography. The cover medium used here is X-ray images. Patient data is secured by attaching it to the cover image. The implementation includes different Steganographic techniques LSB, BHA, and HSA,

Encryption techniques like AES, DES, and RSA. By Integration of LSB and DES techniques, the system achieves an average PSNR value of 32.10. Therefore, the Secret message initially undergoes encryption with DES, and further DES is encrypted using RSA. The resultant cipher text is embedded into the cover image using the LSB technique. It provides an additional layer of security by making the decryption process difficult and reducing the encryption time.

Hasi Saha et al.,[10] The main idea of the research was to Improve LSB Steganography using Random pixel Selection, 1D Logistic Map, and AES Encryption. This research utilizes the irregular pixel selection through LSB Steganography to hide a secret message inside the cover image. LSB Steganography embeds the hidden message bits into the cover image bits without modifying the properties of the cover image. In the following approach, a .bmp image is utilized for embedding the secret message due to its uncompressible and convenient file format. The hidden text is embedded into the LSB bits of the cover image after the encryption of the secret image to increment the security of the hidden text by converting it into cipher text. The secret text is embedded into the LSB bits of the blue channel of RGB pixel values. A 1D Logistic map is used to determine pixel positions to embed the text. Provides better PSNR and lower distortion. Even if the cover image's cipher were cracked, the attacker would still require the encryption password for decryption. The output of the program was similar to the original image.

Youmin Xu et al.,[11] developed a robust method for hiding secret images within a container image while minimizing distortion (Robust Invertible Image Steganography). This advanced framework comprises 3 parts flow-based network to model high-frequency component distribution, with a conditional Normalizing Flow (CNF), a Container Enhancement Module (CEM) for robust reconstruction, and a Distortion-Guided Modulation (DGM) that helps in modifying parameters for distortion levels, to make an adaptable model. It is a difficult task to hide the Images with large secret messages in the small cover image because of distortions and jpeg image format. Robust invertible image Steganography overcomes this issue by introducing a Container Enhancement Module (CEM) and a CANP module to prevent the distortions towards the cover image. Experimental results indicated that this approach offers high imperceptibility and high capacity. Along, with it also protects from the distortions.

Rohan Bhangale et al.,[12] presented the work on the Encryption of EXIF Metadata for the Protection of Photographic Images of Copyright Piracy. In this method, EXIF metadata

is encrypted using the Advanced Encryption Standard or AES-128-bit algorithm, then divided into small blocks and placed at the end of the image file. Metadata is extracted from the image as plain text and then encrypted and transformed into cipher text and then embedded into the image. The above method was tested using JPEG/JPG image format and it was observed that change in the pixel color and metadata. Experimental results indicated the size of the image pixel remained unchanged but the file size changed by -25.15% due to metadata conversion, there was no change in the image color and pixel and there were no changes in the hexadecimal values of the encrypted and original image. It significantly protects the image copyright by encrypting the metadata present in the image, preserving the quality of the image and proof of ownership.

Reddy Madhavi K et al.,[13] developed a method using . In this Steganography, a color image is embedded inside another image using deep neural networks. Deep neural networks were trained simultaneously to hide and reveal images. The training was conducted on the random images taken from the ImageNet data, and various metrics were employed for evaluation. Unlike, other steganographic methods it spreads the representation of the secret message all over the image. The model automatically correlates and combines the relevant features from the data by enhancing the speed of the steganography. Successfully conceals images with low reconstruction errors, outperforming traditional methods. The approach offers flexible concealment with high-quality results, applicable to a wide range of images.

Shahid Rahman et al.,[14] proposed a method based on Huffman coding, the HSI color model, the Multi-Level Encryption Algorithm (MLEA), the Magic matrix, and the Least Significant Bit (LSB) substitution to embed messages securely with the colorless part of the picture (HC-LSBIS-MLE-AC). In this approach, the secret image is converted from an RGB image into an HSI Image, It is encrypted using MLEA encryption and Huffman Encryption. Huffman coding is more robust than other methods. Experimental results provided a high-quality image with minimal distortion and generated high-quality stego images with high adaptability and efficiency. Embedding the secret message at the I-plane of the HSI image rather than the RGB image enhanced security and processing time, it achieved a PSNR value of 79.29 dB over 165 standard images and showcased maximum efficiency and excellency over related works.

Surya Prakash Yalla et al.,[15] proposed a system that implemented a GUI application image steganography in the spatial domain using the Least Significant Bit (LSB)

Steganography and AES Encryption. After the steganography process, the capacity of the cover image gets increased. The modified cover image undergoes Discrete Wavelet Transform (DWT) and then gets encrypted through AES to enhance security and resistance against steganalysis. Experimental results indicated that there wasn't much distortion between the stego image and the transformed cover image, it also preserves the data quality. The Mean Square error (MSE) of the transformed cover image and the stego image was 1.53. As a result, this technique provided a high-quality image, high imperceptibility, less computational complexity, enhanced speed and security, and protection from steganalysis attacks and unauthorized access. It also provides double-layer security by encrypting the first and sending it to the receiver.

Masumeh Damrudi et al.,[16] employed a system that executes LSB Steganographic technique and AES, RSA, DES, 3DES, and Blowfish as encryption algorithms to ensure secure transfer of information. Initially the plain text is converted into the ASCII code, RSA algorithm generates public and private keys which are used for the encryption of the secret text then the cover image is converted to grayscale image, both the message and image are given input to the LSB algorithm. Output of the LSB algorithm is the visible image with encrypted data. The experimental results indicate that decryption time is less than encryption time because key generation takes place in the encryption phase. SNR values are between 66-67, PSNR values are between 72-73 and MSE value is 0.0039. High PSNR and low MSE values indicate there is no distinction between the original and the encrypted image, it produces a high quality stego image. Histogram analysis showed that the message was not easy to detect.

Mustafa M. AbdZaid et al.,[17] built an integrated method that utilizes RC4 as encryption technique and LSB as Steganographic technique. In the process, initially a plain text is given as input to the system, then this text is converted into ASCII code, it is encrypted using RC4 Algorithm and it is placed into the cover image using LSB technique. In the decryption process the encrypted message is taken from the cover image and decryption using RC4 and converted to plain text. Experimental results indicate that encryption and decryption time increased with the increase in the size of the image. The detection of the encrypted message was difficult due to the integration of steganography with cryptography. Integrated approach of steganography with cryptography provides a dual layer security and restores the quality of the stego image.

Prof. Sakshi Shejole et al.,[18] proposed a system that enhanced secret communication Using Multi-image Steganography with Face Recognition and OTP System. In the above

methodology, a single message is divided into multiple messages, and each message is embedded in a different Cover image, using facial recognition which extracts the facial features and verifies the person. LSB steganography is used to encrypt the secret message using the AES-256 algorithm to embed the secret message into the cover image. In the above model, otp system is also used to provide an extra layer of security to the system. This three-layer security makes the above system robust and protects the system from unauthorized attacks. The above system provides safe and secure transmission of information between military parties, multi-image steganography makes the system robust and difficult for hackers, otp and face recognition add extra layer of security.

Table-1 Summary of Existing Approaches

Reference No.	Objective	Methodology	Result	Significance	Limitations
[1]	To implement image encryption method using steganographic LSB method, AES, and RSA algorithm.	The asymmetric algorithm ensures key security and resilience against renowned attacks	The asymmetric algorithm ensures key security and resilience against renowned attacks	This method effectively secures sensitive image data across crucial sectors by addressing major data protection challenges.	Optimizing real-time scalability for efficient handling of large image datasets is a research gap for the proposed encryption method.

[2]	To implement LSB steganography using pixel locator sequence with AES	PLS during the encoding and then decoding process of LSB	PLS disperses data bits across image pixels pseudo-randomly, making decoding impossible without the exact pixel sequence.	PLS boosts security by scattering data bits randomly across pixels, making decoding reliant on having the original sequence	Exploring efficiency in data retrieval and enhanced resistance to pattern detection for the PLS approach.
[3]	To develop an Android-based application, to hide secret information inside the safe photographs.	Symmetric encryption to encrypt the picture, Steganography is used to place the image in LSB bits, and asymmetric encryption to encrypt the key.	The Application provided a user-friendly interface.	The experimental results indicated that Peak-Signal-To-Noise-Ratio values surpassed the threshold.	The data carrying capacity of the cover image has to be increased to incorporate more data into the image.

[4]	To secure secret information by Steganography combined with Advance Encryption Standard (AES) using SHA-256 for security.	securing the key in Advanced Encryption Standard by employing hashing. In this process, the key was encrypted via the hashing technique SHA 256.	The experimental results depicted that there was not much difference between the original image and the stego-image.	AES encryption with SHA-256 hashing ensured strong data concealment and integrity verification for enhanced security.	Encryption and LSB embedding can be resource-intensive, causing lag on devices with limited processing power.
[5]	Chaotic color multi-image compression-encryption/ LSB data type steganography scheme for NFT transaction security.	Chaotic sequences are taken from the chaotic map, at the same time as a couple of mystery snapshots are compressed by the usage of CS and fused into a bigger mystery picture	Performance and Stimulation consequence s indicated more key space and steganographic capability	This model secures the NFT image transactions and connects 2D to 3D textures by offering new thoughts for steganography.	large-scale NFT transaction networks are impacted by the combined compression-encryption, steganography, and anti-counterfeiting measures

[6]	To implement a Time-Based Steganography for images with Dynamic Encryption Key Generation.	It uses two keys: a normal string and one based on the insertion time. The Insertion process produces the time key.	Experimental results showed the PSNR values of 55-81 dB for various cover images.	encryption technique based on time controls the decryption process for secret data only to specific periods	Susceptibility to time-based attacks or interception during key exchange.
[7]	To perform an Analysis and Implementation of Image Steganography by using the AES algorithm.	The secret information is encrypted using the AES encryption method and then embedded into the cover image using LSB Steganography.	Speeds of the encryption and decryption processes were 0.264 Mbps and 0.0631 Mbps, respectively	cover image is imperceptible to the human eyes after subtracting the original image from the encrypted image the resulting image formed is black.	Encryption increases image size, suggesting possible concealed information.

[8]	To Conduct an Analytical Study of the LSB-based Image Steganography Approach.	Improving the performance of the Least Significant Bit (LSB) Steganography by proposing a modified LSB called circular shift LSB.	The prospective research surpasses the past techniques by acquiring the PSNR of 84.46 and MSE of 0.0002124.	.bmp image format is advisable in the above approach for secure data transmission due to its high PSNR and low MSE values.	This approach degrades the Stego-image quality.
[9]	Develop a novel double-layered security approach for medical images through a hybrid Stegano-Crypto technique.	The proposed method offers dual-layer security by leveraging the strengths of Cryptography and Steganography.	The hybrid Stegano-Crypto algorithm, combining LSB and DES, achieves an average PSNR value of 32.10.	It provides an additional layer of security by making the decryption process difficult and reducing the encryption time	Increased computational complexity due to the combination of both steganography and cryptography techniques.
[10]	To Improve LSB Steganography using Random pixel Selection, 1D Logistic Map, and AES Encryption.	This research utilizes the irregular pixel selection through LSB Steganography to hide a secret message inside the cover image.	Provides better PSNR and lower distortion. The output of the program was similar to the original image.	Even if the cipher were cracked, the attacker would still require the encryption password for decryption.	Decrease in the amount of data that can be embedded within the image using this approach.

[11]	To develop a robust method for hiding secret images within a container image while minimizing distortion.	Conditional Normalizing Flow along with a Container Enhancement Module and Distortion-Guided Module.	Experimental results indicated that this approach offers high imperceptibility and high capacity	RIIS offers high capacity, imperceptibility, and robustness against distortions	The size of the images can be large sometimes
[12]	To Encrypt the EXIF Metadata for the Protection of Photographic Images of Copyright Piracy.	EXIF metadata is encrypted using the Advanced Encryption Standard or AES-128-bit algorithm, then divided into small blocks and placed at the end of the image file.	Experimental results indicated the size of the image pixel remained unchanged but the file size changed by -25.15% due to metadata conversion.	Protects image copyright by encrypting metadata, preserving image quality, and providing proof of ownership.	May require additional computational resources, and change in file size due to metadata conversion.
[13]	To develop a method using deep neural networks to conceal full-size color images within other images, advancing steganography.	A color image is embedded inside another image using deep neural networks. Trained simultaneously to hide and reveal images	Successfully conceals images with low reconstruction errors, outperforming traditional methods.	Flexible concealment with high-quality results, applicable to a wide range of images.	Performance may degrade with images outside the training dataset, and training deep networks can be intensive

[14]	To Develop a method that Combines Huffman coding, the HSI color model, the Multi-Level Encryption Algorithm (MLEA), the Magic matrix, and Least Significant Bit (LSB) substitution.	The secret image is converted from an RGB image into an HSI Image, It is encrypted using MLEA encryption and Huffman Encryption.	achieved a PSNR value of 79.29 dB over 165 standard images	Enhanced security with Huffman coding and MLEA. High image quality with minimal distortion.	Limited message embedding capacity of about 20 kb.
[15]	To develop a GUI application image steganography in the spatial domain using the Least Significant Bit (LSB) Steganography and AES Encryption.	The modified cover image undergoes Discrete Wavelet Transform (DWT) and then gets encrypted through AES to enhance security and resistance against steganalysis	The Mean Square error (MSE) of the transformed cover image and the stego image was 1.53.	a high-quality image, high imperceptibility, less computational complexity, enhanced speed and security	parameter specifications and level of accuracy can be improved.

[16]	To employ a system that executes LSB Steganographic technique and AES, RSA, DES, 3DES, and Blowfish	the plain text is converted into the ASCII code ,RSA algorithm generates public and private keys which are used for the encryption of the secret text then the cover image	The experimental results indicate that SNR values are between 66-67 ,PSNR values are between 72-73 and MSE value is 0.0039	Histogram analysis showed that the message was not easy to detect.	.Low Robustness and limited payload capacity.
[17]	To built an integrated method that utilizes RC4 as encryption technique and LSB as Steganographic technique.In the process	a plain text is give as input to the system,then this text is converted into ASCII code ,it is encrypted using RC4 Algorithm	Experimental results indicate that encryption and decryption time increased with the increase in the size of the image.	Integrated approach of steganography with cryptography provides a dual layer security and restores the quality of the stego image.	Vulnerable to statistical attacks and limited capacity.

[18]	To propose a system that enhanced secret communication Using Multi-image Steganography with Face Recognition and OTP System	a single message is divided into multiple messages, and each message is embedded in a different Cover image, using facial recognition which extracts the facial features and verifies the person	This three-layer security makes the above system robust and protects the system from unauthorized attacks	The above system provides safe and secure transmission of information between military parties	High complexity and computational overhead..
------	---	--	---	--	--

2.2 Summary: Drawbacks of the Existing Approaches

In most of the recent approaches optimizing real-time scalability for efficient handling of large image datasets is a research gap for the proposed encryption method. Exploring efficiency in data retrieval and enhanced resistance to pattern detection for the PLS approach. The data carrying capacity of the cover image has to be increased to incorporate more data into the image. Encryption and LSB embedding can be resource-intensive, causing lag on devices with limited processing power. large-scale NFT transaction networks are impacted by the combined compression-encryption, steganography, and anti-counterfeiting measures Susceptibility to time-based attacks or interception during key exchange. Encryption increases image size, suggesting possible concealed information. This approach degrades the Stego Image quality. Increased computational complexity due to the combination of both steganography and cryptography techniques.

Decrease in the amount of data that can be embedded within the image using this approach. The size of the images can be large sometimes May require additional computational resources and change in file size due to metadata conversion. Performance may degrade with images outside the training dataset, and training deep networks can be computationally intensive Limited message embedding capacity of about 20 kb. parameter specifications and level of accuracy can be improved. It includes vulnerability in RC4 encryption which can limit the amount of data to be embedded. They can be used as cryptographic algorithms to encrypt a message before applying steganographic algorithms The drawback of using the DES (Data Encryption Standard) algorithm for cryptography is that its key size of 56 bits is too short for adequate security in this day and age as it can be brute forced quite easily with the right resources.

CHAPTER 3

PROPOSED METHODS

3.1 Problem Statement

As mentioned earlier in the previous chapters this work focuses on hiding confidential information or messages in the images using encrypting algorithms. Out of them, AES-256 and RSA algorithms were selected for this work due to their high efficiency in securing the data and usage and adoption due to their performance in hiding the data and safeguarding from unauthorized users and making it unnoticeable. Basically, encrypting and hiding data is a critical task since various factors have to be taken into consideration due to the advancement in steganalysis tools which are used to retrieve the information easily from the images in which the information is hidden. But employing traditional methods does not scale the demands of various fields using different mediums for transmission of confidential information. So, hence it needs robust methods for securing the data from unauthorized users. In order to withstand the the evolving methodologies used for the retrieval of data by unauthorized users. There is a demanding need for robust technologies in order to not only encrypt data but also conceal it with seemingly harmless files to prevent detection by unauthorized parties.

AES-256 plays a key role in this project by providing the additional security and for strong encryption before its embedded in the image. The high level of security ensures that without the correct key, they cannot understand the data or decrypt the cipher text. Basically, this encryption method transforms the original text into a complex code that is very difficult to decipher it. This technique also makes extremely secure against different attacks and introduces randomness in the embedding process. The initial value is taken as the sum of dimension of the image from where the embedding has to be done. Without knowing the initial random number, it becomes impossible to anyone to detect the hidden information. Key management is monitored using RSA encryption required for both encrypting and decrypting the message. In this approach the RSA's public key allows us to safely transmit these keys over an insecure channel.

Coming to the architecture of AES-256, it is easy to implement and it consumes less memory and uses keys of 128,192 or 256 bits. The required number of rounds depends on the key size. In this process it consists of 14 rounds of transformations. AES has four steps such as SubBytes, ShiftRows, mixed column and add round key. The only step which is responsible to

create confusion in the data is byte sub, the remaining steps are nonlinear. By adding AddRoundKeys it performs the XOR with each byte of the state with the corresponding byte of the next round key. The SubBytes replace each byte in the with its corresponding bytes from the S-box. ShiftRows circularly shift each row of state to the left by an offset. Then transform each column of the state using a fixed matrix multiplication to mix the bytes within each column. Next, repeat the AddRoundKey operation again until the number of rounds specifies by the round key length. The AES also supports different methods for encryption which are as follows: EBC, CBC, OFB, CFB and CTR.

Asymmetric encryption is not suitable for images because the computation is too long. But it is more secure than symmetric encryption because it removes the exchange of the secret key. To take the advantage of the speed of symmetric encryption and security of asymmetric encryption and steganographic methods. We have proposed combination of AES-256, RSA and LSB steganography method to achieve more secure transmission of information through images. AES is utterly designed for allowing parallel processing and hardware acceleration, which improves its performance and also in environments requiring high speed encryption and decryption.

The main objective of the project is to Integrate AES-256, RSA and LSB steganography method to build a robust and secure system.

3.1.1. Objectives of the Project

- To add additional security layer by encrypting the message:

This project adds an additional security layer implemented by encrypting the messages with a symmetric key. For each encryption session, symmetric keys are derived from session keys used to make sure that unique keys are employed. Consequently, this symmetric key encodes the confidential message into a cipher text. The cover image encompasses the least significant bits (LSBs) of its three-color channels modified to carry the cipher text within it. By adopting this method, it is made sure that the secrecy of message is not easily leaked out. Therefore, using a derived symmetric key makes it hard for unauthorized parties to decrypt messages without access to the session key. Because of this every single cover image provides an exclusive and safely encrypted message to show how unique they are from each other. This technique blends cryptography with steganography so as to have strong defense against any sensitive information being leaked out.

- To improve resistance to steganalysis tools and undetectable data embedding:

To assure that the information can't be seen by the naked eye and is hard for steganalysis tools to discover and use Randomized Least Significant Bit (LSB) technique of embedding the information. This method ensures that the hidden data isn't detectable thus preserving the cover image's integrity. Furthermore, incorporate compression and random distribution of data within the image so as to make analysis more difficult. Compression and random distribution of the encrypted data reduces the chances of it being detected by analytical tools. The use of these techniques together strengthens the overall stealthiness and security of concealed data, making it almost impossible for any unauthorized person to identify or extract them.

- To incorporate increased security through randomness:

To add unpredictability in the data embedding method, involve a number generator that is arbitrary. The initial random number needed for discovering the embedded data pattern also reinforces security. It becomes very hard to decode the pattern into which information has been concealed due to this kind of haphazardness unless one has the beginning seed. The confidentiality of the hidden message is therefore made more secure by not following any set rules during embedding processes. Anyone trying to extract this information illegally will experience difficulties since they will have no idea on what grounds to start from when trying such a move. As a result of these measures, unauthorized persons cannot easily break through such systems because there are extra layers which they need break first before getting access into them.

3.2 Architecture Diagram

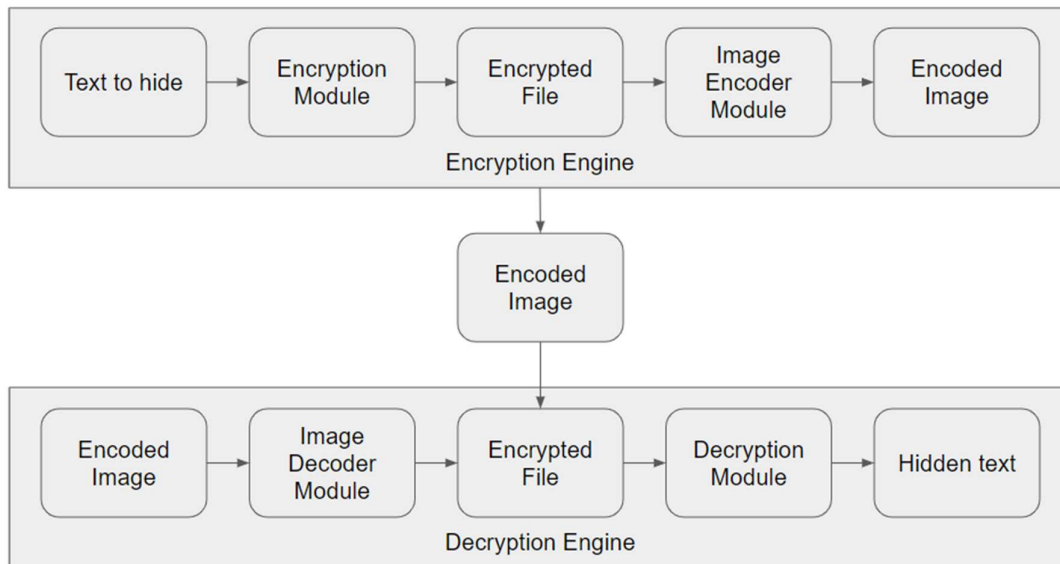


Figure-7 Architecture Diagram

The above architectural diagram(Figure-7) demonstrates a model that helps hide secret information within a cover image through the integration of AES-256, RSA, and randomized LSB. RSA encryption algorithm generates a public and a private key. These two generated keys are used in encryption and decryption techniques. The AES-256 algorithm is used for the generation of the session key. A pseudo-random number generator (PRNG) generates the bit positions where the data must be embedded. In the encryption process, generations of RSA, AES, bit positions, and placing the data into the cover image take place. while, in the decryption, the extraction of the secret message takes place.

In the encryption engine, plain text and public key are fed as input to the encryption module. In the encryption module, public key encrypts the session key generated from AES. A symmetric key is derived from a session key it is used to encrypt the data. Public key ensures safe exchange of session key between sender and receiver. In the encryption module public key encrypts the plain text into the encrypted text. The encrypted text is then given as input to the image encode module. In the image encoder module, the encrypted text is then converted into such a format so that it is suitable to fit in the image. In the image encoder module, the encrypted text is placed inside the LSB bits of the pixels in the image without degrading the image quality. Before sending the image, error correction codes are added to the encoder module to ensure that the encrypted text is recovered correctly. The

encrypted image file contains the image it seems to be normal image, which contains the encrypted text.

In the decryption module, the encrypted image file is given as input to the decryption module. The encoded image is extracted from it which contains the encrypted text. The encoded image is further given as input to the decoding module where the encrypted text is extracted from the image. This module in all likelihood reverses any ameliorations applied throughout the Encoding Module within the encryption stage. For instance, if error correction codes had been delivered before embedding, the Decoding Module could get rid of them at this level. Here, the extracted text and private key is fed into the decryption module. The private key decrypts the session key and the symmetric key is derived from the session key's bits are extracted from the image and they are reconstructed to form the original text. The symmetric key derived is used to decrypt the encrypted message. This process takes place in the decryption module. It outputs the original text message.

3.3 Modules-Connectivity Diagram

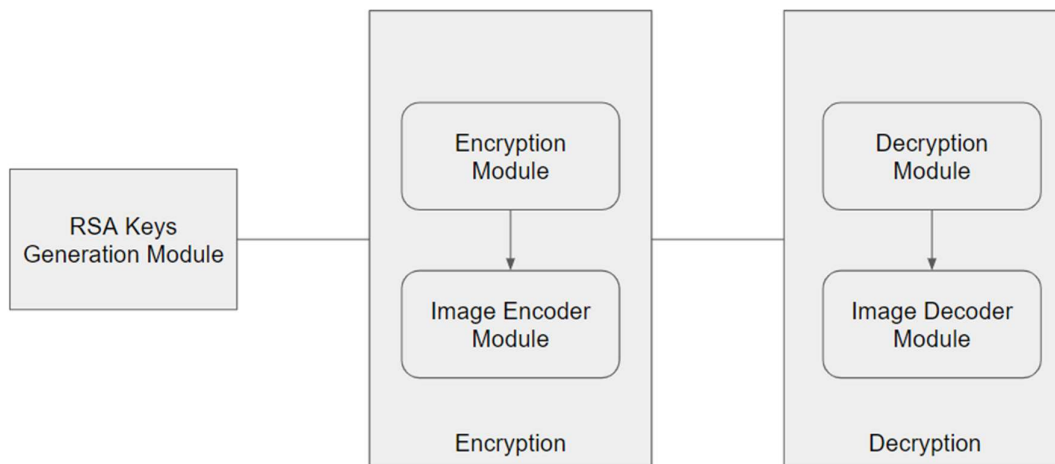


Figure-8 Modules-Connectivity Diagram

The module connectivity diagram as represented in Figure-8 shows how the usage of RSA encryption interacts with and data the actions between noteworthy components of a reliable photo processing utility. Every module within the machine has a completely unique characteristic that provides to the device's basic defense and functionality. The module's interconnectivity ensures a regular and untrained manner for picture encryption and decryption.

The RSA Keys Generation Module operates as the foundation of safety via imparting the vital keys, whilst the Decryption and Encryption components cope with the actual exchange of the information. This modular layout now not simplest greatly enhances safety but additionally permits adaptability and expansion of the equipment.

The RSA encryption module, which calls for the advent of both public and private keys, is the first module inside the modularity diagram. Public key and disturbed lock are comparable. Thus, the text can be encrypted through anybody. However, not everyone could have get entry to the name of the game key. The secret textual content is encrypted using the public key during the encryption technique. The encryption manner makes use of a personal key. It offers the gadget additional double layer protection. The steady key trade between sender and recipient is made possible with the aid of the RSA encryption technology.

The encryption system contains two modules encryption module and image encoder module. This two modules work in the coordinated manner. Encryption module is used to secure data by using RSA public key for the exchange of data. RSA public key is used to encrypt the secret text. After the encryption process the plain text is converted into the cipher text. It is used to strengthen text data by encrypting it. Another module is the Image Encoder module it formats the image in the form of binary or byte stream to make the image suitable for data encryption here the data is embedded into the image. The image is sent as input to the encryption module for the encryption of data using public key. Image Encoder module is used for formatting of the image. It takes raw data and formats the image suitable for the encryption. Image data is given input to Encryption module to encrypt the image data. Any steady facts processing device ought to have an encryption module, in particular while dealing with touchy facts like photos. The Encryption Module and the Image Encoder Module collaborate within the unique system architecture to safely encrypt image information using RSA encryption.

The decryption system consists of a decryption module and an image decoder module. The encrypted image data and private key are sent as input to the Decryption module to decrypt the image data. In the decryption module private key converts the encrypted image data to its decrypted form. Outputs the decrypted image data to the image decoder module. In the Image decoder module, the decrypted image data transformed back to its original form it is used for formatting the data. It converts data from the byte format to original text format without disturbing the meaning of the data. An important part of a secure records processing gadget is the Decryption Module, which transforms encrypted statistics (ciphertext) again into plaintext.

Together with the encryption module, this module makes certain that information is stable all through storage and transmission.

The workflow of the above system is First RSA key pair generation where public and private keys are generated. Then the public key is given to the encryption module and private key is given to the decryption module. Then input is sent as the raw image data to the Image encoder module here the image data is encoded to its suitable format. Then the encoded image data is sent to the encryption module where the public key encrypts the encoded image data to encrypted image data. For decryption process, the encrypted image data is sent to the decryption module encrypted image data is transformed to decrypted image data using a private key. Finally, it is sent as input to the image decoder module to convert data to its original form. This structure guarantees that the photo data stays secure all through transmission or garage using RSA encryption, making it hand handiest to parties possessing the best personal key for decryption. The Decryption Module is required to opposite the encryption procedure and repair encrypted records to their authentic, readable country. It makes use of the RSA private key to effectively decrypt the information. This is an in-depth evaluation of the duties, skills, and workflow of the Decryption Module. Data recovery: Gets the unique plaintext records by way of decrypting the ciphertext.

3.4 Software and Hardware Requirements

Software Requirements:

- Operating System: windows, macOS or Linux
- Development Environment: Python 3.8 or later
- Integrated Development Environment: Pycharm, Visual studio code, Jupyter Notebook
- Libraries and Packages:
 - Cryptography library (e.g. PyCryptodome or cryptography)
 - Image processing library. (e.g. Pillow)
 - Numpy for numerical operations.
 - PyQt5 or Tkinter for GUI development.

Hardware Requirements:

- Processor: I5
- Memory: 8GB or more

- SSD with atleast 1 GB
- Stable internet connection
- High-resolution monitor for better visualization of images
- Hardware Security module for secure key storage
- External storage device for backup

3.5 Modules and their Description

3.5.1 RSA keys Generation Module

Generation of RSA keys is a fundamental process that take place in the concept of cryptographic systems aimed at safe communication. It involves creating a pair of keys, a distinctive couple of keys in which one of them is categorized as the public key for encipherment while the other is the personal key desirable for decipherment. The initial step involves the generation of two large prime numbers p and q which are to be confidential. These primes are then multiplied to create n , and the modulus incorporated in both the keys shown in the above formula is used to restrict and contain the set range of the integer n . Subsequently, Euler's totient function is defined as $\phi(n) = (p-1)(q-1)$. This function plays an extremely significant role in identifying the public and the private exponents.

The public exponent e is then selected. It has to be a number more than 1 and less than $\phi(n)$ and this number must not share any factors with $\phi(n)$ except 1. (i.e., $\gcd(e, \phi(n)) = 1$) The standard choices for e are 3, 17, or sometimes also fewer like say 1 due to their balance between security and efficiency. From this, the private exponent d gets calculated. In this case, d is the modular multiplicative inverse of e modulo $\phi(n)$ where $(d \times e) \bmod \phi(n) = 1$. This allows to link the public key from above with a private key, so only someone who knows the corresponding private can access the original message (from #1) and decrypt it. The final key pair to be used in public will become (e, n) and in private (d, n) . The public key is available for everyone and treats the data, while the private key decrypts ensuring only intended readers should read it.

3.5.2 Encryption Module

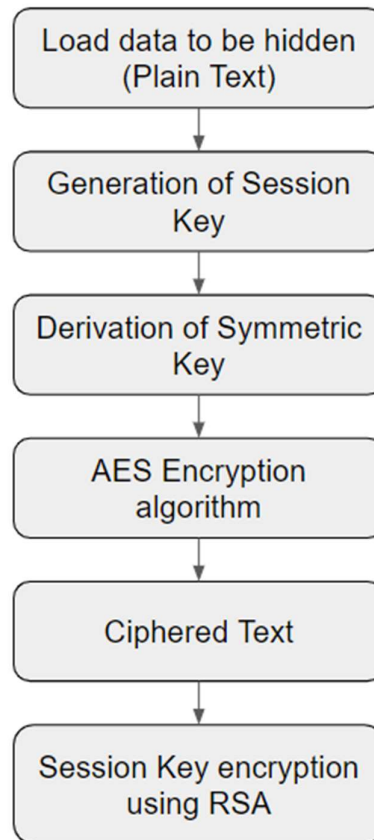


Figure-9 Encryption Module

In the Encryption module as Figure-9 represents, the AES-256 algorithm is used for the generation of the session key which is used to encrypt the data in the encryption module. From this, a symmetric key is generated which is used to encrypt the data. The AES session key is the transient symmetric key. If the plain text is encrypted in different sessions, then the resultant cipher text is different because of the randomness of the session key it is uniquely generated for each session. A Session key is generated securely through the Random Number Generator (RNG) or a cryptographic library. In the next phase, A symmetric key is derived using the Password-Based Key Derivation Function 2(PBKDF2) function. The user provides a passphrase or password as the input, for the given input a pseudo-random function along with a salt is applied to the input. Using the salt function a new random value is added to the password at each iteration so that for the given same password at each iteration a new key is generated. These iterations are repeated many times to make it difficult for the brute force attacks.

Advanced Encryption Standard (AES) is a symmetric algorithm that uses the same key for both encryption and decryption processes. Where Cipher Block Chaining is used for the modes of operation to apply on cipher blocks to encrypt the blocks. In the above model, the AES algorithm is integrated with Cipher Block Chaining (CBC) to strengthen the encryption process. Initially, plain text with a fixed size of 128 blocks is taken and it is converted into multiples of data by using padding (PKCS7). A random Initialization Vector (IV) is a random value that is used in encryption algorithms to make sure that identical plain text blocks are encrypted using different cipher blocks using the same key it is unique and different for each session to prevent attacks. An IV of a size similar to the block size of the cipher is taken, and the first plain text block is XORed with IV. Then the resulting block is encrypted using AES and a symmetric key. Further, the previous cipher blocks are XORed with the next fixed plain text blocks.

In the last phase, the Encryption of the session using the recipient's public key. Where the speed of the symmetric algorithm and the security of the asymmetric algorithm are integrated to form a robust model. A session key is generated from the Random number generator for symmetric encryption of data. The RSA public key is obtained from the recipient's public key by the sender. It is used for the encryption of the session key. The session key is transformed into cipher text using the public key it can only be transformed to its original form using the corresponding private key. It is sent to the receiver with encrypted data. These hybrid encryption algorithms ensure key exchange between sender and receiver. Ensuring the intended recipient who has the private key can only decrypt the session key. It adds dual-layer security to the data.

The above process ensures the safe transmission of information between the sender and receiver and makes the system powerful by integrating symmetric encryption with asymmetric encryption. Involves the generation of the random session using the AES-256 algorithm followed by the derivation of the symmetric key from the corresponding session key used for the encryption of data by encrypting data with CBC to ensure powerful transmission of data along with the IV vector to ensure that the unique plain text is encrypted using different cipher plain text blocks, at last Encryption of the session key using RSA public key to leverage the security of the asymmetric algorithm and speed of the symmetric algorithm.

3.5.3 Image Encoder Module

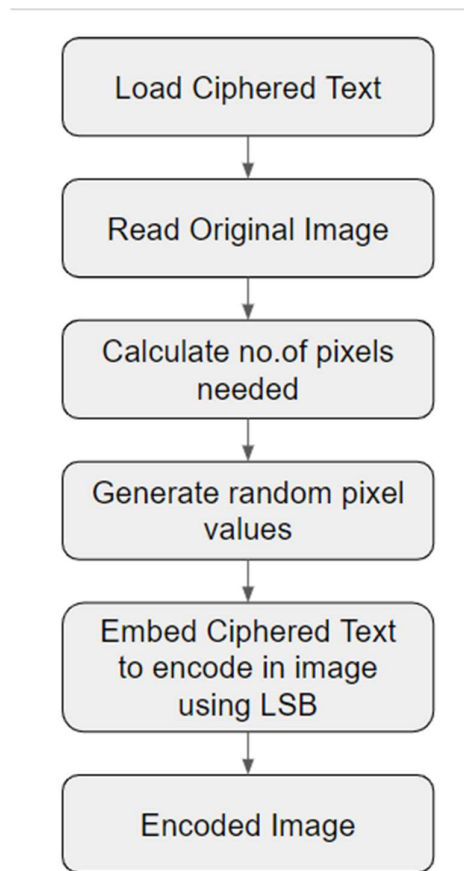


Figure-10 Image Encoder Module

Image Encoder Module is represented in Figure-10. In first step, the sum of the image dimensions is taken to get the initial random start for the randomization to be consistent. The sum is taken to initialize the random number generator (RNG) to ensure consistent embedding of data into the image. If the image contains height and weight as the dimensions they are added to get the sum. This sum is used to initialize the RNG to generate the random sequence, it has to be the same for all the sum values generated each to make the process reproduced. The last step ensures that the above randomization is reversible at the decryption stage to promote consistent randomization. Using the Sum of the Image Dimensions to Generate a Random Initial Start for Consistent Randomization. By the usage of a steady seed derived from the picture itself, you make sure that the randomization technique may be reproduced exactly. This is important for efficiently retrieving the embedded facts.

Reading original image and ensuring it's in a suitable format (RGBA) for data embedding. First loading the image from the image library using PIL. Then check the image

for correct depth and alpha channel for efficient transmission of data. Check the image format if it is in RGBA or the other format for proper modeling it is in another format convert it to RGBA. The image dimensions are for the quality embedding of data. In the next step, image pixels are manipulated for embedding data using the NumPy array. The data bits are embedded into the image using the LSB technique. Save the encoded image and send it back to the drive.

To read a file you need to open it in the binary mode, once the file is read into the memory it has to be compressed to place more data into the file. Compression is done through zlib. To embed more data into the image file its size is reduced. If the size is small, it promotes faster processing of data. It is difficult to understand the compressed data. Encrypting the data includes installing the cryptography library to import several modules for key generation and encryption process. Here the salt is added to the password input given by the user it is a random value added to the password to ensure each time a different value is generated with the same password. A random Initialization Vector (IV) is a random value that is used in encryption algorithms to make sure that identical plain text blocks are encrypted using different cipher blocks using the same key it is unique and different for each session to prevent attacks. An IV of a size similar to the block size of the cipher is taken, and the first plain text block is XORed with IV. Then the resulting block is encrypted using AES and a symmetric key. Further, the previous cipher blocks are XORed with the next fixed plain text blocks.

Embedding file size and each bit of data (encrypted session key, salt, IV, encrypted data) in the image using LSB matching. Embedding data into the image by modifying the image's LSB bits by the image bits. In the next step, the data is converted into binary format data including metadata, salt, and IV, encrypted data. To embed the data into the image file first, we need to load the image, flatten the image embed the data into the image, and reshape the data and then the encoded image is saved. After embedding the data into the image using LSB matching the image has to be saved to protect the image it is done by using the save method. At last, transforming the modified numpy array back to the file and saving it. The above steps ensure consistent randomization and embed the secret text message into the image by formatting the image and transferring the image to the suitable format without disturbing the image quality and dimensions, by ensuring the data quality throughout the exchange between sender and receiver.

3.5.4 Image Decoder Module

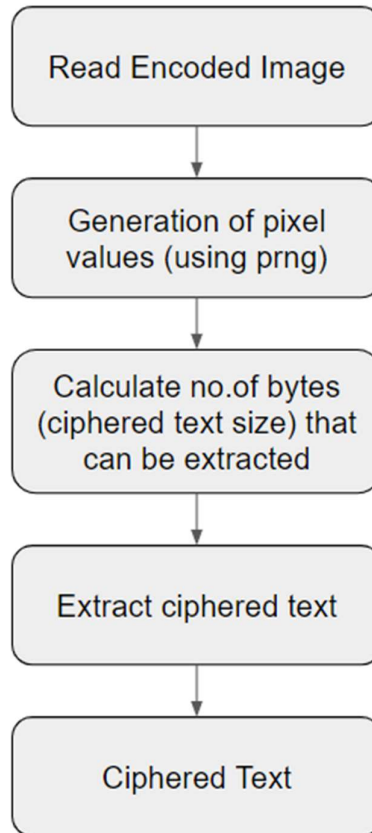


Figure-11 Image Decoder Module

Image Decoder Module is represented in Figure-11. Reading the encoded image involves extracting the encoded image file from the disk which contains the encrypted message. Then check whether the file is in the correct format for extraction or not. If not convert the file to the required format for extraction and processing. The first step is to load the encoded image file from the disk using the Pillow library (PIL) used for opening and manipulating the file. To extract the data from the file the image has to be in a suitable format in this model it is in the RGBA format provides pixel manipulation, and the alpha channel shows the transparency in the picture. To perform manipulations on the data it has to be converted into the numpy array which is a powerful tool for mathematical computations. In this step image is loaded, the format of the image is verified, and the data is converted into the numpy format for manipulation.

Extracting the hidden data from the image. For extracting the encrypted text from the image finding the starting position where the text can be extracted is very crucial for the purpose the sum generated using a pseudo-random generator can be used for both encoding

and decoding purposes. It provides the starting point where the text can be extracted. To extract the binary data from the image you need to start from the generated starting point to do that the 2D image has to be converted into the 1D image the image has to be flattened iteration starts from the starting point it goes throughout the pixel by taking the LSB values of the RGB pixels and storing the values into a binary string. The accumulated data must be similar to embedded data in order it include file size, salt, initialization vector, and encrypted data. A file size of 32 bits is used to store the binary string. It is also required to store the IV, salt, and encrypted data. In this, the extracted data is interpreted.

Decrypting the data using a key. The key is extracted from the password and salt. The key is generated by the password by making use of salt to add a random number to the password. To get the key, firstly the salt has to be extracted from the encrypted data. To get the key Derivation Function (KDF) is applied to the password similar to the PBKDF2. To decrypt the data, the key has to be used with IV the cipher text. The cipher (CBC) and mode (CFS) have to be the same as in the encryption. Encrypted has to be processed with the cipher (CBC) to get the decrypted data. Data has to be decompressed if it is compressed in the encryption process. And it has to return to its original format. The process starts with the extraction of keys from the password and salt. IV with the derived key is used to decrypt the encrypted data to get the original data. At, last data is decompressed.

3.5.5 Decryption Module

Decryption Module is represented in Figure-12. The decryption of the session key using the RSA private key. RSA is an asymmetric encryption technique, that involves the generation of public and private keys. Encryption of the session key is done using the RSA public key and decryption of the session key using the RSA private key. The public key can be shared with anyone, but the private key cannot be shared with anyone. The session key is encrypted in the encryption which is actually used to encrypt the data. In the image decoding process, along with the encrypted data the session key is also extracted. The session key is decrypted using the RSA Private Key which is used for encrypting the data in the encryption process.

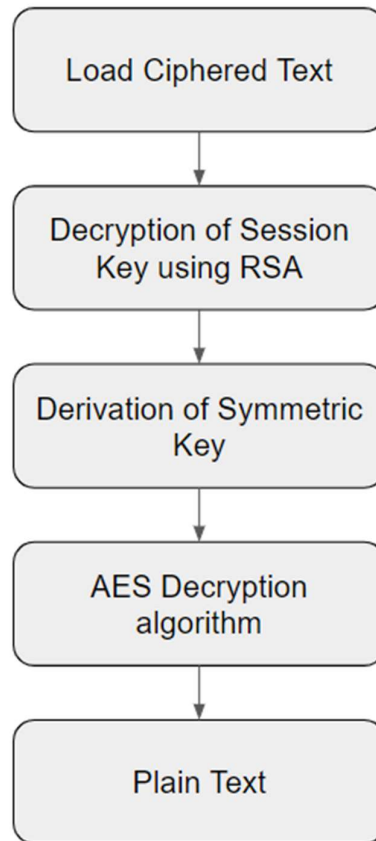


Figure-12 Decryption Module

Derivation of Symmetric Key using a key derivative function (PBKDF2-HMAC). In this process after decryption of the session key it is used to derive a symmetric key from using PBKDF2-HMAC. Input in the PBKDF2-HMAC is password, salt, and iterations. The decrypted session key generated using the RSA private key, salt, and other parameters are given as input to the PBKDF2-HMAC the function outputs a symmetric key. KDF is used to strengthen the session key even if it is derived from the weak passphrase.

When the symmetric is derived from the session key it is used to decrypt the data. For the decryption of data AES algorithm is used which operates on 128 bits of fixed block size. An IV is used to ensure that repeated plain text blocks do not generate repeated cipher text blocks for repeated plain text blocks it generates different cipher text blocks. The AES cipher is initialized using the IV and the derived symmetric key. AES cipher is used to process the encrypted data to produce the original data. The session key is decrypted using the RSA public key, derivation of the symmetric key from the session key, and decryption of the encrypted data using the symmetric key.

3.6 Requirements Engineering

3.6.1 Functional Requirements

The functional requirements mentioned here will state what the planned work should do in terms of function:

- a. AES-256 and RSA encryption: whereby AES-256 is employed as symmetric key for encrypting data and RSA asymmetric keys for storing keys and secret information.
- b. Randomized LSB Embedding: Through using random least significant bits (LSB) method to embed private data into images covertly without any human detection.
- c. User-defined key input: to allow users enter their own passphrases for generating AES-256 and RSA key pairs hence making it flexible.
- d. Compression and distribution of data: such that it distributes data within the images to manage storage, improve durability against detection.
- e. Command line interface: which offers an intuitive CLI for users to interact with on encryptions facilities and key management.
- f. Performance optimization: Optimize algorithms that handle large files while maintaining high levels of simultaneousness between encryption accuracy and embedding size

3.6.2. Non-Functional Requirements

The functional requirements mentioned here will state what the planned work should do in terms of function:

- a. Security: to make sure there is strong encryption and key management
- b. Performance: to ensure that it works faster with large files
- c. Reliability: to make sure it works all the time well without any raising any errors.
- d. Compatibility: make sure it works on different operating systems like windows, linux and macOS.
- e. Data Integrity: make sure that data which is hidden in images stays safe.
- f. Maintainability: to keep it easy to be used by everyone and ensuring error handling if something goes wrong.

3.7 Analysis and Design through UML

3.7.1 Class Diagram

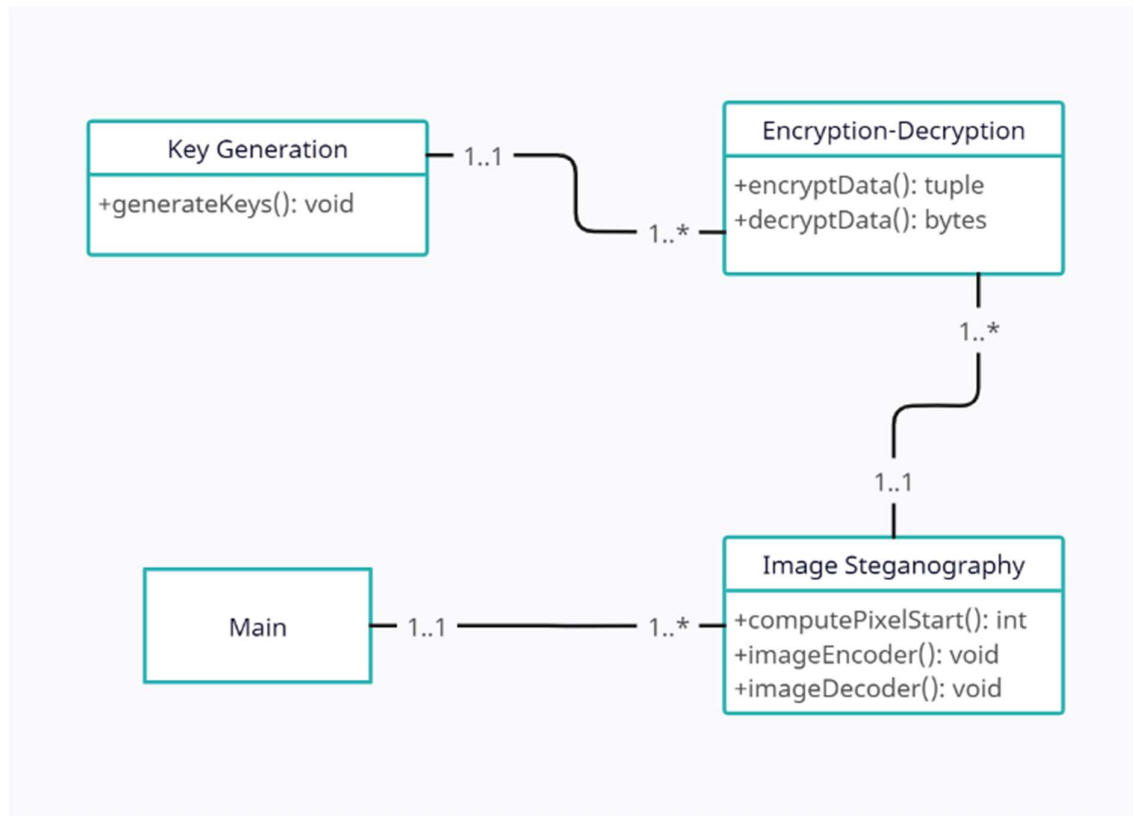


Figure-13 UML Class Diagram

The given model proposes a uml class diagram shown in figure-13. It illustrates classes employed in the design of the proposed model. These classes relate to each other in order to complete the task of hiding data into PNG image files by using AES-256, RSA algorithm, and Randomized LSB Steganography. The four parts of this system model are Key Generation, Encryption-Decryption, Image Steganography and Main. This generateKeys() method produces keys for use by the Key Generation class. The Encryption-Decryption class encrypts data and decrypts it with decryptData(). The Image Steganography class hides their messages or secrets in form of pictures (imageEncoder()) while afterward they can retrieve that information again (imageDecoder()). The Main class starts embedding data on pixels; this is controlled through repeated computePixelStart() calls aimed at coordinating key generations; encryption/decryption processes while hiding or extracting steganographic information within different sets belonging to these categories respectively. In multiple instances of each of these classes which is useful in aiding processes like generation keys; encryption/decryption of

message texts not limited steganographic embeddings thus Main class controls all interactions between them.

3.7.2 Sequence Diagram

For Encoding:

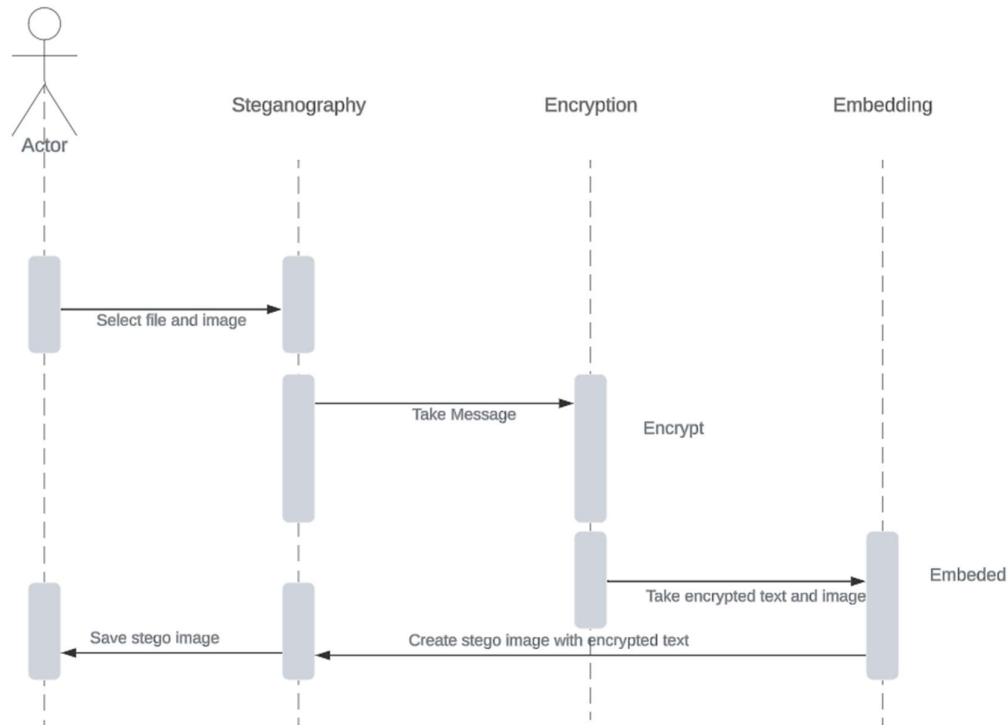


Figure-14 UML Sequence Diagram for encoding

The above figure-14 shows the flow of encoding the message into a image. To get started, an actor will need to choose a file and an image. The chosen message is then sent for encryption. After that, the encrypted text is passed on to the embedding component together with the image. This particular component embeds this information into the picture thereby creating what we call a stego image. Lastly, the stego image which contains the concealed encrypted text is saved by the actor. The flow from selecting initial files up until saving final stego image can be seen in the diagram above which also shows how steganography interacts with encryption as well as embedding components.

For Extracting:

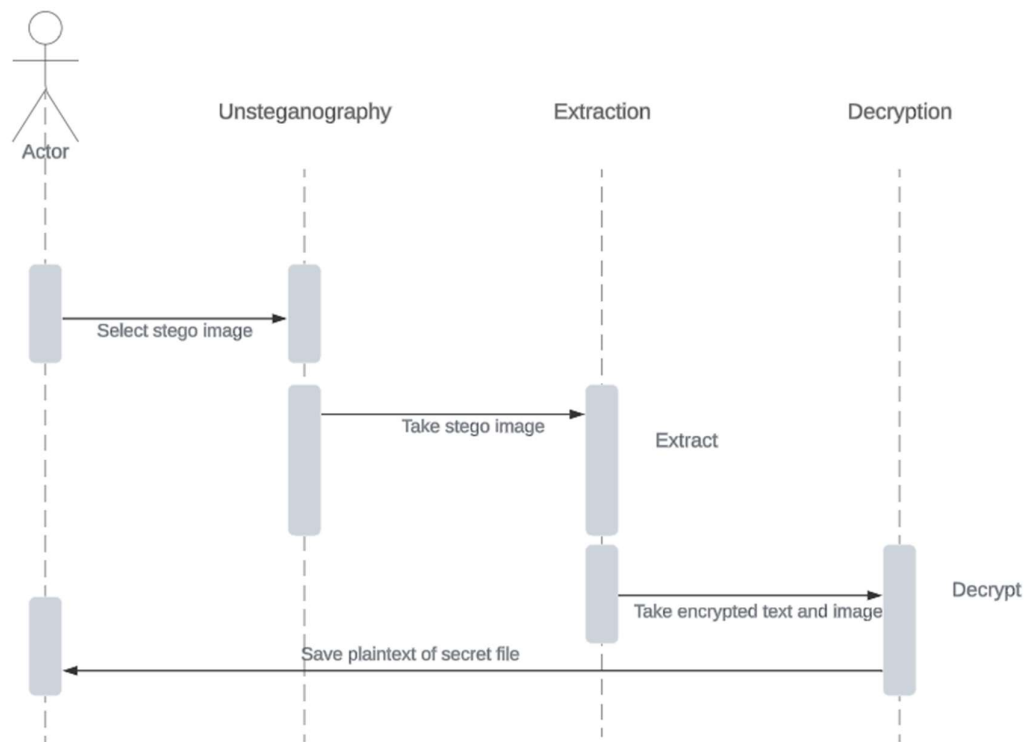


Figure-15 UML Sequence Diagram for Extracting

The above sequence diagram(Figure-15) shows the process of extracting and decrypting the hidden data from an stego image. An actor starts by choosing a stego image, then they send this image to the unsteganography component. The stego image is received by the extraction device from there which extracts the hidden encrypted text in the image. The decrypted original plain text message is now with the decryption component which also takes along the image. Lastly, the secret file's plaintext is saved by the performer. This map illustrates how one goes about selecting a stego image up until saving their decrypted secret file whilst demonstrating interactions between different components such as unsteganography, extraction and decryption ones.

3.7.3 Use Case Diagram

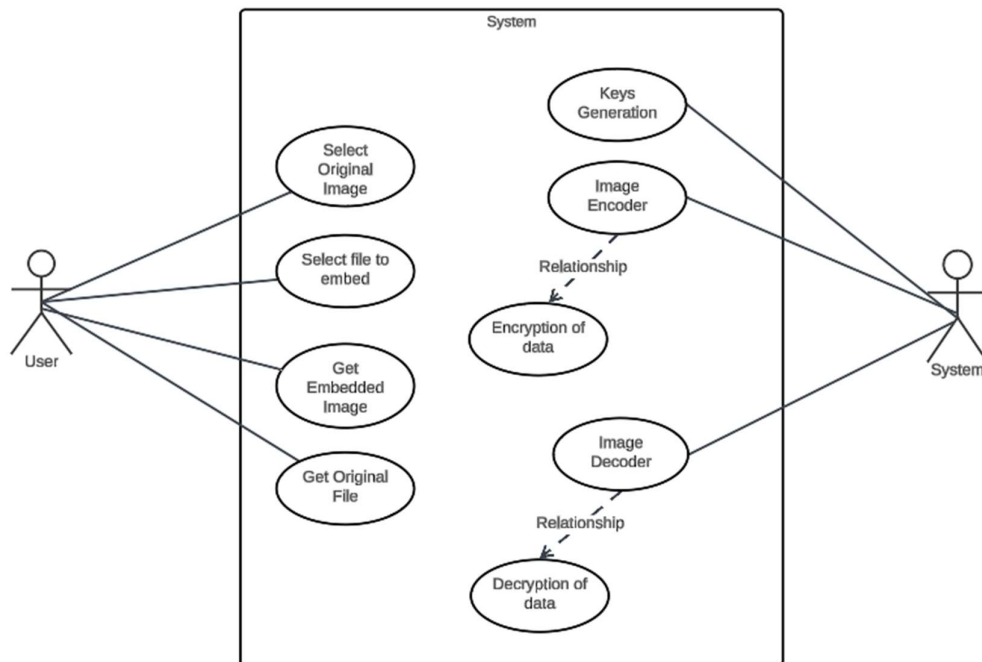


Figure-16 UML Use Case Diagram

The UML use case diagram as figure-16 represents an interaction whereby a user triggers a system that has been designed for encrypting and hiding data in images. Initially, the chooser prompts by pointing at the cover image and the stegano file. After this action, key generation for security is done by the steganography system. Then, the original image is encoded such that through the image encoder component, the embedded (encrypted) data is written into the image. When the sender wants to communicate some data secretly, the recipient must get the steg image containing the secret message. To recover the original document, a request is sent from the user to this application. Furthermore, the UML use case diagram depicts decryption process which is triggered by the retrieval of data during information extraction phase thus ensuring integrity and security.

3.7.4 Activity Diagram

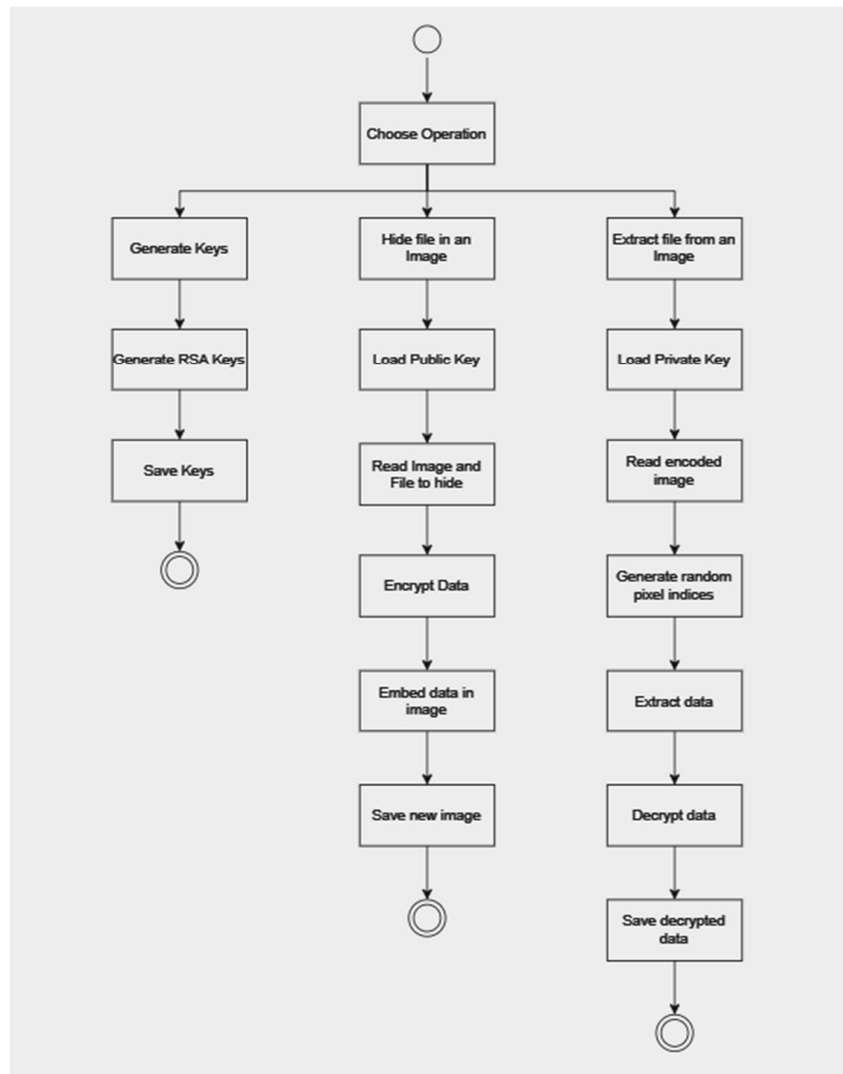


Figure-17 UML Activity Diagram

As the above figure-17 represents, The stage initially requires the client to pick an operation:

Generate Keys: Selecting this option will begin the process of creating a new RSA key pair. The keys that are created will then be saved into a file.

Hide file in an Image: This option enables you to hide a file inside an image. The steps involved include: The program asks you to load the public key for encryption. You also have to load the image in which the file will be hidden. Then, load the file you want to hide. A list of random pixel indices is generated within the image. Encrypt the file using the loaded public key. Embed the encrypted data into the image using the generated random pixel indices. A new image is

saved with the file hidden inside it. The program asks you to load the public key for encryption. You have to load the image in which the file will be hidden. Load the file you want to hide, then it encrypts it using the loaded public key. A list of random pixel indices is generated within the image embed the encrypted data into the image using the generated random pixel indices. The system then saves a new image with the file hidden inside it.

Extract file from an Image: With this option, you can retrieve a hidden file from an image. This is done as follows: The program will ask you to load the private key for decryption. Also, you need to load the image with the hidden file. Then the system reads the encoded image. The information hidden in the image is extracted. Finally, decrypt the extracted data using the loaded private key and save it as a new file. Click on the “Load Private Key” to start the decryption process ensure you have also uploaded the image file containing the secret information The system will then read the coded picture to extract the hidden data inside the image. Use the loaded private key to decrypt the extracted information and save the resulting plaintext as a new document.

3.8 Testing

Image steganography is the process of hiding the images and information in such a way that one cannot notice such a change when they look at the original image. In order to achieve the requisite security of the concealed data as well as the quality of the picture, several testing methodologies are used. One of the simplest methods is visual comparison, where the cover and the stego-images are compared *visa -vis* in case of any distortions or irregularities detected easily. Other superior quantitative measurements are the Peak Signal to noise Ratio (PSNR) and Mean Squared Error (MSE). PSNR compares the quality of the stego image with the original one, greater values representing better quality, while MSE indicates the average squared difference between the images; the smaller the better.

In the analysis section, additional qualitative analysis is done through histogram analysis of the pixel values. By comparing the histograms of the original and stego images distortion can be observed in areas relevant to the statistical measures of the stego image that can indicate the existence of concealed information. An additional method based on Structural Similarity Index (SSIM) is used to measure the structural similarity of the original and the stego image where SSIM values approximate to 1 express a high similarity and changes are minimal. Further, the Bit Error Rate (BER) determines the efficiency of the extracted hidden data against

the embedded information; the BER has low values hence indicating the efficiency of the process.

There is also the aspect of robustness testing through which one tries to attack the stego image through methods such as image compression, image cropping as well as other deformation that can be applied on stego images and the ability of extracting the actual data from the stego image must be tested. This guarantees that the steganographic method will be resilient to conditions of the real world where images may be distorted. Finally, verifying the payload capability makes it possible to define how much information can be hidden while achieving the best visual quality, meeting the obvious requirement of stealth while not compromising on the picture's realism.

In our project, we have used PSNR for testing the steganographed image. Peak Signal-to-Noise Ratio is a measure of quality. More than 30 dB is recommended to contain little to no distortion.

- 30 dB to 40 dB: These ranges normally imply that the stego image is almost distorted in some way and such changes are normally negligible and almost invisible.
- 40 dB and above: PSNR values of the images within this range indicate very high quality and the amount of distortion is practically negligible.
- 20 dB to 30 dB: Although images in this range are decent, they are not ideal, meaning that there will be some distortion apparent but it may be acceptable for most uses based on the application that is needed.
- Below 20 dB: This range typically points to large deviations, which, in turn, would mean that the hidden information might be easily discernible and the steganography compromised.

According to the PSNR, the goal for almost all image steganography techniques is to have a PSNR that is greater or equal to 30 dB so that the hidden information will not be easily detected and distinguished from the actual quality of the carrying media.

Chapter 4

Results and Discussions

4.1 Detailed Explanation of the Experimentation Results

This chapter displays the results of this steganography project, which embeds data into images by altering between the red and blue channels. To evaluate how effectively the embedding process performed, two methods are used: visual examination and quantitative measurements (Peak Signal-to-Noise Ratio).

We took our data for experiment from websites: pexels, pixabay, vecteezy

4.1.1 Data Embedding Results

Embedding Procedure

The red and blue channels of the image pixels were used to conceal compressed and encrypted files during the data embedding procedure. Steps involved:

Step-1: Pillow library (PIL) was used to read the original pictures.

Step-2: The data that needed to be hidden was encrypted with a public key in order to guarantee security.

Step-3: The encrypted data was compressed to reduce its size.

Step-4: The compressed and encrypted data was alternatively placed into the image pixels of red and blue channels. To assure consistency, embedding procedure used a pseudo-random number generator (PRNG) with the sum of the image dimensions

Example

A sample file of size 76 KB, was embedded using an sand picture (sand.png) with 257 x 183 dimensions. After embedding was completed successfully, the results were assessed as follows:

- A Visual comparison
- To evaluate the image quality objectively, PSNR value was computed and it was 63.24

Visual Comparison

When compared visually, the steganographed and original photos were nearly the same. It appears that the image quality was not considerably affected by embedding procedure because there are no observable distortions or artifacts.



Figure-18 Sand Image(Courtesy: Source [21])

Quantitative Measurement

Peak Signal-to-Noise Ratio (PSNR): To measure the difference between the original and steganographed images, the PSNR value was computed.

$$\text{PSNR} = 20 * \log_{10}(\text{maxPixel}) - 10 * \log_{10}(\text{mse})$$

Where, maxPixel is the maximum possible pixel value (255 for 8 – bit images).

mse is the mean squared error between the original and steganographed images.

For the sand image above, the computed PSNR value was 63.24 dB. This remarkably high PSNR value suggests that the embedding procedure did a very good job of maintaining image quality.

4.1.2 Data Extraction Results

Extraction Procedure

The Steganographed image's concealed data had to be extracted during the extraction procedure. Steps involved:

Step-1: Pillow library was used to read the steganographed image.









Step-2: The red and blue channels of the picture pixels were used to extract the data in turn. To locate the embedded data accurately, the extraction method employed the same PRNG start as was used during the embedding procedure.











Step-3: The matching private key was used to decrypt the extracted data. The original file was recovered by decompressing the decrypted data.

Example

To extract the secret file, the identical sand image (sand.png) was utilized. The file that was extracted was exactly the same as the original file before embedding, indicating that the extraction procedure was effective.

Table-4.1 Result

Cover Image	Stego Image	Image Metrics
		PSNR: 65.19
		PSNR: 65.06
		PSNR: 64.80
		PSNR: 64.78

		PSNR: 64.55
		PSNR: 64.67
		PSNR: 64.37
		PSNR: 64.34
		PSNR: 63.24

4.2 Significance and Advantages

Significance

The main idea of this job is to stop secret facts in digital pictures from being seen. This is done by these ways. The first one is to change strong codes like AES-256 and RSA. Another way is to mix the Least Significant Bit (LSB) with something random. This is to make sure that it can be both scrambled and hidden inside the photo easily. There are two layers of safety. To

break into these extra ways, people who are not allowed to get in will find it almost impossible to discover what is there anytime they do. In areas like financial institutions, governments or even medical care centers where keeping things private is very important this becomes even more necessary.

In addition it increases secrecy of the embedded data and makes the information more immune to attacks known as steganalysis. In this method, the least significant bit is replaced by a randomly generated number to ensure that no modifications can be detected by human eyes. Therefore, since preservation of image quality is crucial in practical applications, such systems are highly effective. Also, distribution of secret bits within the cover object is made random so that extracting them or identifying where they are becomes very difficult hence this enhances security against steganalysis.

Again, the importance of error detection and correction mechanisms as well as possible integration of block chain technology further enhances this project. Error detection and correction ensures that even in circumstances where the image goes through noise or distortions during transmission, the embedded data is still kept intact and accurate. This makes the steganographic system more authentic and credible by providing a tamper-proof audit trail. The hidden data security is combined with technologies that can be used within other fields associated with high levels of decency in databases like forensic investigation and secure communication systems to give a comprehensive approach.

Advantagaes

- Enhanced invisibility: When you hide data in pictures by using the least significant bit (LSB) method, you should randomize it, so that human eyes cannot easily detect any change. The secret message is placed into each pixel's intensity where the least significant bit is swapped with a bit from the private data. This technique maintains image quality because secret bits have been spread throughout all LSBs causing them not to follow any set pattern; hence, making them seem unrelated to any underlying information. Furthermore, this type of distribution greatly reduces the chances of detection which is what enhances invisibility in steganography or other methods used for concealing information within digital files.
- Scalability and Flexibility: The suggested system can adjust for different circumstances and forms of details even for delicate biometric data like fingerprints. This implies that it can find applications in various areas ranging from forensics to secure

communications where a variety of things can be done with it depending on what is necessary.

- Error Detection and Correction: It's important to include error detection and correction formats like Reed-Solomon codes or CRC in order for the information to be transmitted despite having been distorted or affected by noise. This is very important as far as truthfulness of practical data is concerned.
- Versatility: Modern cryptographic solutions are very versatile and flexible in terms of their application areas that include file encryption, secure communications and data storage. To cater for diverse security needs, these solutions have introduced different modes of operations including CBC (Cipher Block Chaining). They also facilitate their implementation in various programming languages and operating systems through being cross-platform. Besides, they can be easily incorporated into existing protocols and systems without any complications.

CHAPTER 5

Conclusion and Future Enhancements

After exploring various applications of image steganography and emphasizing on the transmission of data through embedded mediums in different formats, we came to know that it has become crucial to secure the information from unauthorized parties. It is understood that preventing the illegitimate access of confidential information in different fields has become rising demand for providing trust to the users who are utilizing the services provided by organizations. Therefore, to manage, maintain and secure the information there is need for advancements in steganographic methods and also expanding its capability for adopting it in different scenarios.

Our project mainly focuses on innovating the digital steganography by combining the AES-256, RSA, and DCT. This one not only improves the data security but also ensures that embedded information stays hidden within the images only. By using the strong encryption techniques like AES-256 and RSA, we can protect the hidden data from unauthorized access. Using DCT, it also helps to embed data safely while keeping the image looking normal. Adding a random number generator it also enhances security by making it impossible to guess the pattern of hidden data without the correct starting number.

Rather than using LSB, AES-256, RSA and DCT (discrete cosine transformations) can be combined to improve current digital steganography techniques. This innovative technique seeks to hide sensitive information in images with a greater degree of concealment and security. In using strong encryption approaches such as AES-256 and RSA. The privacy of the obscured content is also guaranteed. Additionally, DCT can be used for efficient data hiding processes in steganographic applications that maintain imperceptibility. While incorporating a random number generator would make this process even more secure by rendering predictable extraction impossible without the right key. This well-rounded approach not only enhances data protection but it also toughens up the resistance of steganography system against detection and analysis.

Other steganographic techniques can be explored to make further improvement on this project and enhance the strength of the existing strategies. Another technique is replacing Randomized Least Significant Bit (LSB) with Discrete Cosine Transform (DCT) Steganography. This makes the hidden data even more difficult for detection algorithms by

modifying its frequency coefficients. Also, embedding processes that take into account image characteristics may benefit from machine learning algorithms that adjust dynamically. By blending these sophisticated techniques with AES and RSA encryption, higher levels of confidentiality and data integrity are possible in the system. To this effect, continuous testing and optimization using large datasets drawn from various sources will help refine the model and ensure its effectiveness and reliability in real world applications.

To effectively integrate this project with fingerprint data from crime scenes which is to be stored secretly several key steps must be followed. First, the steganographic techniques in the project such as AES, RSA, and either Randomized Least Significant Bit (LSB) or Discrete Cosine Transform (DCT) steganography will be modified to allow for the embedding of fingerprint data onto digital images. This process will see that sensitive biometric information like fingerprints can be concealed secretly within an image file. The integration will involve working together with forensic experts in order to develop guidelines for capturing and encoding fingerprint data at crime scene level. Therefore, specialized software tools will be created that automate the embedding process so as to ensure consistency and security throughout handling of confidential information. In addition, AES and RSA encryption are used in securing these fingerprints when embedded therein hence ensuring privacy even if the images are intercepted without proper authority or accessed by unauthorized persons.

Chapter 6

Appendices

6.1 Source Code

```
import argparse

import sys

import os

import random

from PIL import Image

import cv2

import numpy as np

# from skimage import data, img_as_float

# from skimage.metrics import structural_similarity as ssim

from cryptography.hazmat.primitives.padding import PKCS7

from cryptography.hazmat.primitives import serialization

from cryptography.hazmat.primitives.asymmetric import padding

from cryptography.hazmat.primitives import hashes

from cryptography.hazmat.primitives.kdf.pbkdf2 import PBKDF2HMAC

from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes

from cryptography.hazmat.backends import default_backend

import zlib

from getpass import getpass


def encryptData(data, publicKey):

    # Generate a random session key

    sessionKey = os.urandom(32) # 32 bytes for 256-bit key
```

```

# Derive a symmetric key from the session key

salt = os.urandom(16) # 16 bytes for 128-bit salt

kdf = PBKDF2HMAC(

    algorithm=hashes.SHA256(),

    length=32,

    salt=salt,

    iterations=200000, # Increased iterations for added security

    backend=default_backend()

)

key = kdf.derive(sessionKey)


# Encrypt the data with AES

iv = os.urandom(16) # 16 bytes for 128-bit IV

cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=default_backend())

encryptor = cipher.encryptor()

padder = PKCS7(algorithms.AES.block_size).padder()

paddedData = padder.update(data) + padder.finalize()

encryptedData = encryptor.update(paddedData) + encryptor.finalize()


# Encrypt the session key with RSA

encryptedSessionKey = publicKey.encrypt(

    sessionKey,

    padding.OAEP(

        mgf=padding.MGF1(algorithm=hashes.SHA256()),

        algorithm=hashes.SHA256(),

```

```

        label=None
    )
)

return encryptedSessionKey, salt, iv, encryptedData

def decryptData(encryptedSessionKey, salt, iv, encryptedData, privateKey):
    # Decrypt the session key with RSA
    sessionKey = privateKey.decrypt(
        encryptedSessionKey,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )

    # Derive the symmetric key from the session key
    kdf = PBKDF2HMAC(
        algorithm=hashes.SHA256(),
        length=32,
        salt=salt,
        iterations=200000, # Increased iterations for added security
        backend=default_backend()
    )
    key = kdf.derive(sessionKey)

```

```
# Decrypt the data with AES

cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=default_backend())

decryptor = cipher.decryptor()

decryptedPaddedData = decryptor.update(encryptedData) + decryptor.finalize()

unpadder = PKCS7(algorithms.AES.block_size).unpadder()

decryptedData = unpadder.update(decryptedPaddedData) + unpadder.finalize()


return decryptedData
```


REFERENCES

- [1] Abdelkader Moumen, Hocinen Sissaoui, "Images encryption method using steganographic LSB method, AES and RSA algorithm". Department of mathematics, LANOS laboratory, Badji Mokhtar University(2016). <http://dx.doi.org/10.1515/nleng-2016-0010>
- [2] Krishnakant Tiwari, Sahil J Gangurde, "LSB Steganography Using Pixel Locator Sequence with AES" ABV-Indian institute of information technology and management, Gwalior (4th December 2020).
- [3] Esther Hannah M, J. Jerusalin Carol, " An Android-based Image Steganography Approach to Data Communication Security using LSB and Password-based Encryption" International Journal on Recent and Innovation Trends in Computing and Communication, VOL. 11 NO .10 (2023). <https://doi.org/10.17762/ijritcc.v11i10.8708>
- [4] Vikas Singhal, Yash Kumar Shukla, Navin Prakash. "Image steganography embedded with advance encryption standard(AES) securing with SHA-256". International Journal of innovative technology and exploring engineering, volume 9, June 2020. <http://dx.doi.org/10.35940/ijitee.H6442.069820>
- [5] Zheyi Zhang, Yinghong Cao, Hadi Jahanshahi. "Chaotic color multi-image compression- encryption/LSB data type steganography scheme for NFT transaction security". Department of mechanical and manufacturing engineering, University of Manitoba, Winnipeg, Canada.(5th November 2023). <https://doi.org/10.1016/j.jksuci.2023.101839>
- [6] Kevin Wijaya, Bryan Lansky, Cecilia Ariani Dewia, Rojali, Ghinaa Zain Na, "Time-Based Steganography Image with Dynamic Encryption Key Generation", 2023, 8th International Conference on Computer Science and Computational Intelligence. <https://doi.org/10.1016/j.procs.2023.10.521>
- [7] Riya Kedia, Biresh Kumar, Pallab Banerjee, Pooja Jha, Tannisha Kundu, Mohan Kumar Dehury, "Analysis and Implementation of Image Steganography by using AES algorithm", 2023, Proceedings of the 7th International Conference on Trends in Electronics and Informatics (ICOEI 2023). <http://dx.doi.org/10.1109/ICOEI56765.2023.10125790>

- [8] Oluwakemi Christiana Abikoye, Roseline Oluwaseun Ogun-Dokun, Sanjay Misra, Akasht Agrawal, "Analytical Study on LSB-based Image Steganography Approach", 2022, IN Book: Computational Intelligence in Machine Learning (pp.451-457). <http://dx.doi.org/10.1109/ICIP.2001.958299>
- [9] D. Arul Suresh, J. Jude Moses Anto Devakanth, Dr. R. Balasubramanian, "A novel double layered security for medical images using hybrid Stegano-Crypto technique", 2022, International Journal of Health Sciences, 6(S10), 267–284. <https://doi.org/10.10072Fs12553-021-00602-1>
- [10] Hasi Saha, MST. Rafia Chowdhury, G C Saha, Masum Billah, "Random Pixel Selection Based Improved LSB Image Steganography Method Using 1D Logistic Map and AES Encryption Algorithm", 2022, International Journal of Innovative Science and Research Technology ISSN No:-2456-2165
- [11] Youmin Xu, Chong Mou, "Robust Invertible Image Steganography", Shenzhen Fundamental Research Program, No.GXWD20201231165807007-20200807164903001. <https://doi.org/10.1109/CVPR52688.2022.00772>
- [12] Hendro Wijayanto, Yudi Prayudi, Imam Raidi, "Encryption EXIF Metadata for Protection Photographic Image of Copyright Piracy", IJRCCT, Vol 5, Issue-5, May 2016.
- [13] Reddy Madhavi K, K. Pranitha, "Image Steganography Using Deep Neural Networks", Information and Knowledge Systems. <https://doi.org/10.52305/YKAF1880>
- [14] Shahid Rahman, Jamal Uddin, Hameed Hussain, Aftab Ahmed, "A Huffman code LSB based image steganography technique using multi-level encryption and achromatic component of an image", (2023). <http://dx.doi.org/10.21203/rs.3.rs-2579014/v1>
- [15] Surya prakash yalla, Archana Uriti, Abhishek Sethy, "GUI implementation of modified and secure image steganography using Least Significant Bit substitution", International Journal of safety and security engineering, Vol 12, No 5, October 2022, pp. 639-643. <https://doi.org/10.18280/ijssse.120513>
- [16] Mustafa M.Abd Zaid, Ahmed ali Talib Al-khazaali, Ahmed Abed Mohammed, "LSB steganography using Dual layer for Text crypto-stego", BIO web conferences 97, 00069 (2024), ISCKU. <https://doi.org/10.1051/bioconf/20249700069>

- [17] Malavika prabhakar, Aiswarya krishnan, Lavanya Nandanasabapathi, Mrinalini Majumdar, D.Saveetha, "Secret messages in social media using LSB and AES algorithms", International journal of engineering and technical research, ISSN: 2321-0869 (O) 2454-4698 (P) Volume-8, Issue-10, October 2018.
- [18] Masumeh Damrudi, Kamal Jadidy Aval, "Image Steganography using LSB and encrypted message with AES,RSA,DES,3DES,and Blowfish", International Journal of Engineering and Advanced Technology (IJEAT) Volume-8 Issue-6S3, September 2019.
- [19] "steganography-in-tokenization", [Online]. Available: <https://www.hsc.com/resources/blog/steganography-in-tokenization/>
- [20] "zlib-vs-gzip-vs-zip", [Online]. Available: <https://www.baeldung.com/cs/zlib-vs-gzip-vs-zip>
- [21] "beach-sand-dunes-landscape-7153932", [Online]. Available: <https://pixabay.com/photos/beach-sand-dunes-landscape-7153932/>

<1%

● 15% Overall Similarity



Top sources found in the following databases:

- 12% Internet database1% Publications
- Crossref Posted Content database10%
- database
- Submitted Works database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1

cse.griet.ac.in

Internet

2%

2

fastercapital.com

Internet

1%

3

degruyter.com

Internet

<1%

4

mdpi.com

Internet

<1%

5

ijeat.org

Internet

<1%

6

Kwame Nkrumah University of Science and Technology on 2022-07-16

Submitted works

<1%

7

Queensland University of Technology on 2024-06-02

Submitted works

<1%

8

erpublication.org

Internet

<1%

9

ijraset.com

Internet

<1%