International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)
https://ijcjournal.org/index.php/InternationalJournalOfComputer/index

A Hybrid Cryptography and Steganography Approach Using RSA and LSB for Secure Message Transmission

Ida Ayu Gde Suwiprabayanti Putra[a*],Gst. Ayu Vida Mastrika Giri[b],Ngakan Putu Bagus Ananta Wijaya[c],Desak Putu Tia Rusilia Wati[d]

[a,b,c,d]*Udayana University, Jimbaran, Badung 80361, Indonesia*
[a]*Email:iagsuwiprabayantiputra@unud.ac.id*
[b]*Email:vida@unud.ac.id*

Abstract

Information security has become a critical concern in digital communication, requiring robust methods to ensure both confidentiality and imperceptibility. Cryptographic techniques such as RSA effectively protect message content by transforming plaintext into ciphertext; however, the presence of encrypted data may raise suspicion and become a target for attacks. Conversely, steganography conceals the existence of the message but lacks strong protection against extraction if detected. This study proposes a hybrid approach that integrates RSA cryptography with Least Significant Bit (LSB) steganography to enhance secure message transmission. The proposed method first encrypts the secret message using RSA, and subsequently embeds the resulting ciphertext into a digital image using the LSB technique. The system is implemented in Python and evaluated using Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) to assess image quality and imperceptibility. Experimental results demonstrate that the proposed method achieves high PSNR values (above 49 dB) and low MSE, indicating minimal visual distortion and high fidelity of the stego image. The integration of RSA and LSB not only secures the message content but also effectively conceals its presence, providing a dual layer of security. Therefore, the proposed hybrid method is suitable for secure digital communication applications requiring both data confidentiality and stealth.

Keywords: Information security; cryptography; steganography; RSA; Least Significant Bit (LSB); data hiding; secure communication; image steganography.

Received: 2/10/2026
Accepted: 4/10/2026
Published: 4/17/2026

** Corresponding author.*

1. Introduction

Information security has become an essential aspect of modern digital communication, particularly with the rapid growth of data exchange over the internet. Ensuring confidentiality, integrity, and authenticity of information is crucial in preventing unauthorized access and data breaches [1]. Cryptography is one of the most widely used techniques for securing information by transforming plaintext into ciphertext, making it unreadable to unauthorized users [2]. Among various cryptographic algorithms, RSA is a well-known public-key cryptosystem that provides a high level of security based on the computational difficulty of factoring large prime numbers [3]. Despite its effectiveness in protecting message content, encrypted data produced by cryptographic methods often appears suspicious and may attract unwanted attention or attacks [4]. To address this limitation, steganography is employed as a complementary technique. Steganography hides secret messages within digital media such as images, audio, or video, making the presence of the message undetectable to observers [5].

One of the most commonly used methods is the Least Significant Bit (LSB) technique, which embeds secret data into the least significant bits of image pixels with minimal visual distortion [6]. However, steganography alone does not provide strong protection if the hidden message is successfully extracted [7]. Therefore, combining cryptography and steganography has emerged as an effective approach to enhance data security. In this hybrid approach, cryptography secures the content of the message, while steganography conceals its existence. Several previous studies have explored such combinations and demonstrated improved security performance [8,9]. However, challenges remain in maintaining image quality while ensuring robust protection. This study proposes a hybrid security model that integrates RSA cryptography with LSB steganography for secure message transmission. The proposed system encrypts the message using RSA and embeds the resulting ciphertext into a digital image using the LSB method.

The system is implemented using Python and evaluated using Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) to measure image quality and imperceptibility. The main contributions of this research are as follows: (1) the development of a hybrid RSA–LSB model for secure communication, (2) the implementation of an integrated system for encryption and data hiding, and (3) the evaluation of the proposed method based on quantitative image quality metrics. The results demonstrate that the proposed approach achieves high image quality while providing enhanced security, making it suitable for practical secure communication applications.

2. Materials and Methods

2.1. RSA Cryptographic Algorithm

RSA is a widely used public-key cryptographic algorithm that provides secure data transmission based on the mathematical complexity of factoring large prime numbers [3]. The algorithm uses two keys: a public key for encryption and a private key for decryption.

The RSA process involves the following steps:

1. Two large prime numbers (p) and (q) are selected.
2. The modulus is calculated as (n = p \times q).
3. Euler's totient function is computed as (\phi(n) = (p-1)(q-1)).
4. A public exponent (e) is selected such that (1 < e < \phi(n)) and (gcd(e, \phi(n)) = 1).
5. The private key (d) is computed as the modular inverse of (e) modulo (\phi(n)).

The encryption process is defined as:

$$C = M^e \pmod{n}$$

and the decryption process is defined as:

$$M = C^d \pmod{n}$$

where (M) is the plaintext and (C) is the ciphertext.

2.2. Least Significant Bit (LSB) Steganography

Least Significant Bit (LSB) is a spatial-domain steganography technique that embeds secret data into the least significant bits of image pixels [6]. This method is widely used due to its simplicity and minimal impact on image quality. In a digital image, each pixel consists of RGB components. The LSB technique replaces the least significant bit of each pixel with bits of the secret message. Since the modification occurs in the lowest bit, the visual difference between the original image and the stego image is almost imperceptible to the human eye [5].

The embedding process is performed sequentially until all message bits are inserted into the image.

2.3. Proposed Hybrid Method

This study proposes a hybrid approach that combines RSA encryption and LSB steganography to enhance data security.

The workflow of the proposed system consists of the following stages:

1. Message Input: The sender inputs the secret message.
2. Encryption: The message is encrypted using RSA to produce ciphertext.
3. Bit Conversion: The ciphertext is converted into binary form.
4. Embedding: The binary data is embedded into the cover image using the LSB method.
5. Stego Image Generation: The output is a stego image that visually resembles the original image.
6. Extraction: The receiver extracts the embedded bits from the stego image.
7. Decryption: The extracted ciphertext is decrypted using the RSA private key to retrieve the original message.

This hybrid approach ensures dual-layer security, where cryptography protects the content while steganography

conceals its existence.

2.4. System Implementation

The proposed system is implemented using Python programming language. Several libraries are utilized, including:

- PyCryptodome for RSA encryption and decryption
- Pillow for image processing
- NumPy for numerical operations

The system consists of four main modules:

1. Key generation module
2. Encryption and embedding module
3. Extraction and decryption module
4. Image quality evaluation module

A graphical user interface (GUI) is also developed to facilitate user interaction and simplify the encryption and decryption processes.

2.5. Evaluation Metrics

To quantitatively evaluate the performance of the proposed hybrid method, two widely used image quality metrics are employed: Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR). These metrics are essential for assessing the level of distortion introduced during the embedding process and determining the imperceptibility of the stego image. MSE measures the average squared difference between the pixel intensities of the original image and the stego image. It provides a direct indication of the amount of distortion caused by data embedding. A lower MSE value indicates that the stego image is more similar to the original image, implying better preservation of visual quality.

The MSE is calculated as:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j) - K(i,j)]^2$$

where (I(i,j)) represents the pixel value of the original image and (K(i,j)) represents the pixel value of the stego image. PSNR is derived from MSE and is used to measure the overall quality of the stego image in decibels (dB). It reflects the ratio between the maximum possible pixel value and the distortion introduced by embedding. A higher PSNR value indicates better image quality and higher imperceptibility.

The PSNR is calculated as:

$$PSNR = 10 \cdot \log_{10}(\frac{MAX_I^2}{MSE})$$

where (MAX) denotes the maximum possible pixel value, typically 255 for 8-bit images. In general, PSNR values above 30 dB indicate that the distortion is not perceptible to the human visual system, while values above 40 dB are considered to represent high-quality images. Therefore, the use of MSE and PSNR in this study provides a reliable and objective evaluation of the effectiveness of the proposed steganographic method in maintaining image fidelity while embedding encrypted data.

3. Results

3.1. System Implementation Results

The proposed hybrid system combining RSA cryptography and LSB steganography was successfully implemented using Python. The system consists of key generation, encryption and embedding, extraction and decryption, and image quality evaluation modules. The implementation allows users to input a secret message, encrypt it using RSA, and embed the resulting ciphertext into a cover image using the LSB technique. The output of the system is a stego image that visually resembles the original image. On the receiver side, the embedded message can be extracted and successfully decrypted using the corresponding private key. Experimental testing confirms that the system performs end-to-end secure communication correctly, where the extracted message is identical to the original plaintext.

The implementation of the proposed system was carried out using the Python programming language. The implementation steps are described as follows:

A. Library Installation

The system utilizes several supporting libraries, including:

- PyCryptodome (for RSA encryption with OAEP padding),
- Pillow (for image processing),
- NumPy (for numerical and array operations).

The project structure is organized as follows:

- keygen.py for RSA key generation,
- embed.py for encryption and LSB embedding,
- extract.py for extraction and decryption,
- metrics.py for calculating MSE and PSNR.

B. RSA Key Generation

A 2048-bit RSA key pair is generated, consisting of a public key (public.pem) for encryption and a private key (private.pem) for decryption. The keys are stored in PEM format, and the private key is securely protected and not shared.

C. Message Processing Workflow

The data flow in the system is defined as follows:

- The plaintext message is taken as input (text or file).
- The message is encrypted using RSA-OAEP with the public key, producing ciphertext in byte format.
- A header of fixed length (32 bits) is added to store the size of the ciphertext.
- The header and ciphertext are converted into a binary sequence for embedding.

D. LSB Embedding Process

The embedding process is performed as follows:

- The cover image (PNG format) is loaded and converted into RGB channels.
- The embedding capacity is calculated as width × height × 3 bits (one bit per RGB channel).
- The system ensures that the payload (header and ciphertext) does not exceed the image capacity.
- The least significant bit (LSB) of each pixel channel is sequentially replaced with the payload bits.
- The resulting image is saved as a stego image in PNG format to maintain lossless quality.

E.Extraction and Decryption Process

The extraction process is carried out as follows:

- The stego image is read, and LSB values are extracted sequentially from RGB channels.
- The first 32 bits are interpreted as the header to determine the payload length.
- The subsequent bits are extracted according to the payload size and reconstructed into ciphertext.
- The ciphertext is decrypted using RSA-OAEP with the private key.
- The decrypted result is validated by converting it into UTF-8 format and comparing it with the original message.

This implementation ensures accurate end-to-end message transmission, where the extracted message is identical to the original plaintext.

3.1.1 User Interface Implementation

3.1.1.1 Main Application Interface

The application interface is developed as a web-based system using the Streamlit framework, allowing it to be executed locally through a web browser. The main interface includes a sidebar navigation menu that provides access to the primary system modules: Generate Keys, Encrypt & Embed, Extract & Decrypt, and Metrics (MSE & PSNR).

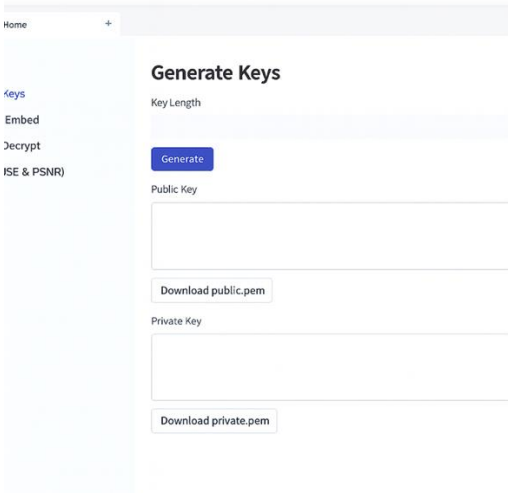

Figure 1: Main Application Interface

3.1.1.2. Generate Keys Interface

The Generate Keys page provides functionality for automatically generating an RSA key pair. This interface allows users to create and manage cryptographic keys required for encryption and decryption.

The main components include:

- A dropdown menu for selecting key length (e.g., 2048, 3072, or 4096 bits),
- A “Generate” button to initiate key generation,
- Text fields displaying the generated public and private keys in PEM format,
- Download buttons for saving the keys as public.pem and private.pem.

When the user clicks the “Generate” button, the system generates the RSA key pair based on the selected key length. The keys are displayed instantly and can be downloaded as separate files.

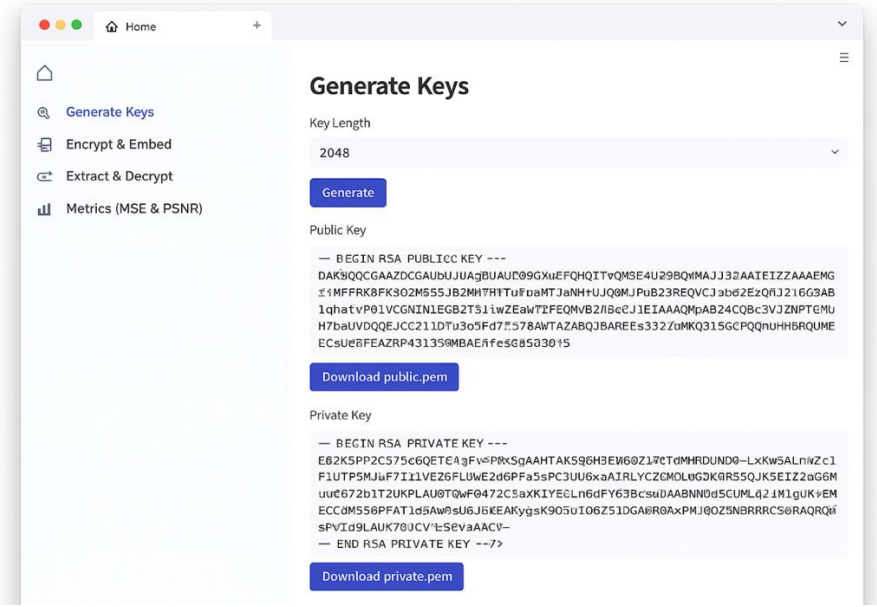

Figure 2: Generate Keys Interface

3.1.1.3. Encrypt & Embed Interface

The Encrypt & Embed page is used to perform message encryption and embedding into a digital image using the LSB method.

The main components include:

- Upload field for the public key (public.pem) used for RSA-OAEP encryption,
- Upload field for the cover image (PNG/JPG), which is processed in RGB format,
- A text input area for entering the secret message,
- An “Encrypt & Embed” button to execute the process.

When activated, the system performs the following steps:

1. Reads the input message,
2. Encrypts the message using RSA-OAEP,
3. Adds a 32-bit header representing ciphertext length,
4. Converts the payload (header + ciphertext) into binary form,
5. Embeds the binary data into the LSB of RGB pixel channels,
6. Generates a stego image in PNG format.

After processing, the system displays a preview of the stego image and provides a download option for the generated file.

This interface demonstrates that encryption and steganography processes can be performed seamlessly through a

user-friendly interface without requiring command-line operations.

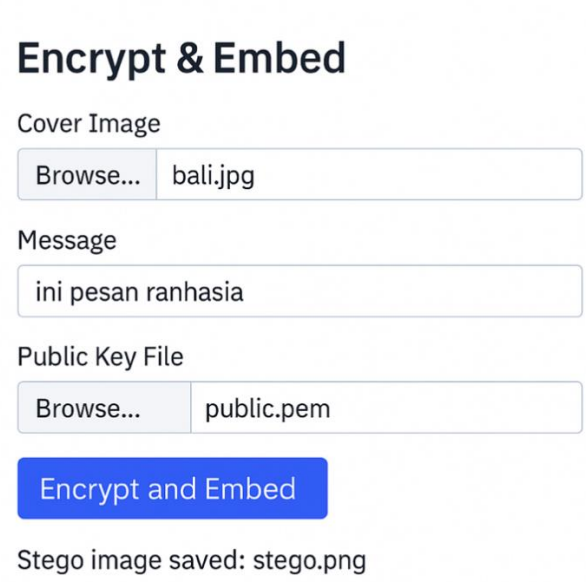

Figure 3: Encrypt & Embed Interface

Figure 4: Stego Image Output

3.1.1.4. Extract & Decrypt Interface

The Extract & Decrypt page is designed to retrieve and decrypt hidden messages from the stego image.

The main components include:

- Upload field for the private key (private.pem),
- Upload field for the stego image (PNG/JPG),
- An “Extract & Decrypt” button to initiate the process,
- A display area for showing the extracted message.

The system performs the following operations:

1. Reads LSB values from RGB channels of the stego image,
2. Extracts the first 32 bits as the payload header,
3. Determines the payload size,
4. Extracts the corresponding ciphertext bits,
5. Reconstructs the ciphertext,
6. Decrypts the ciphertext using RSA-OAEP,
7. Displays the recovered plaintext message.

If the extracted data cannot be decoded into UTF-8 format, a warning is displayed. The results confirm that the extraction and decryption processes successfully reconstruct the original message.

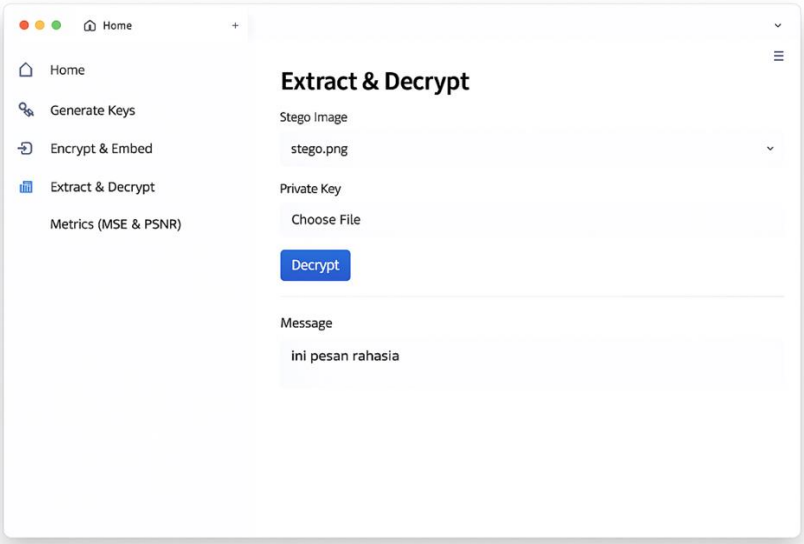

Figure 5: Extract & Decrypt Interface

3.1.1.5. Metrics (MSE & PSNR) Interface

The Metrics (MSE & PSNR) page is designed to evaluate the visual quality of the stego image compared to the original image.

The interface provides:

- Upload field for the original (cover) image,
- Upload field for the stego image,
- A “Calculate MSE & PSNR” button to perform evaluation.

When executed, the system:

1. Reads both images and ensures identical dimensions,
2. Calculates the Mean Square Error (MSE),

3. Calculates the Peak Signal-to-Noise Ratio (PSNR),
4. Displays the computed values on the interface.

Low MSE values and high PSNR values (typically above 30 dB) indicate that the visual difference between the original and stego images is minimal. This confirms that the LSB embedding method preserves image quality effectively.

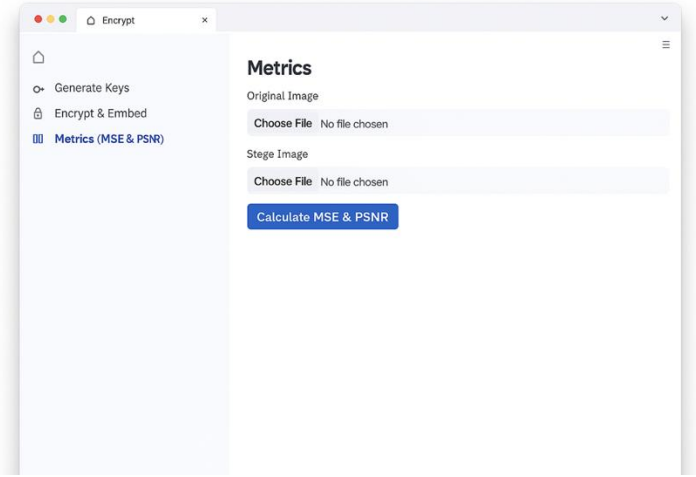

Figure 6: Metrics Interface

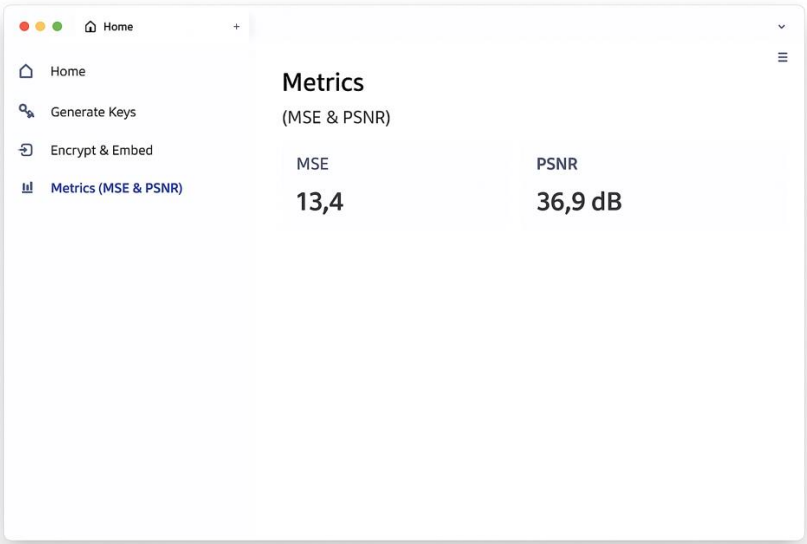

Figure 7: Metrics Results

3.1.1.6. Discussion of User Interface Implementation

Based on the implementation results, the entire system workflow—from key generation, encryption and embedding, extraction and decryption, to image quality evaluation—can be performed through the graphical user interface without requiring command-line interaction. The simple and structured design improves usability,

allowing even non-technical users to operate the system effectively. Furthermore, the integration of the Metrics module directly supports quantitative evaluation, enabling efficient analysis of the proposed method's performance.

3.2. Image Quality Evaluation

The performance of the proposed hybrid method was evaluated using two widely accepted image quality metrics, namely Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR). These metrics are essential for assessing the level of distortion introduced during the embedding process and determining the imperceptibility of the stego image. MSE measures the average squared difference between the pixel intensities of the original image and the stego image. The experimental results indicate that the MSE values obtained are extremely low, suggesting that the embedding process introduces only minimal changes to the original image. This demonstrates that the LSB technique effectively preserves the structural and visual characteristics of the cover image. PSNR, which is derived from MSE, provides a quantitative measure of image quality in decibels (dB). Higher PSNR values indicate better visual quality and lower distortion.

In this study, the PSNR values consistently exceed 49 dB, which is significantly higher than the commonly accepted threshold of 30 dB for imperceptible distortion. This result confirms that the stego images produced by the proposed method are visually indistinguishable from the original images. Furthermore, the high PSNR values indicate that the embedding of encrypted data does not significantly degrade image quality, even when the payload includes both ciphertext and a header structure. This demonstrates the efficiency of the LSB method in maintaining image fidelity while accommodating additional data. The results also highlight the stability of the proposed approach across different test cases. The consistency of low MSE and high PSNR values suggests that the method performs reliably regardless of the input image, provided that the embedding capacity is not exceeded. Overall, the evaluation results confirm that the proposed hybrid RSA–LSB method achieves a strong balance between data security and image quality. The ability to maintain high imperceptibility while embedding encrypted information makes the approach suitable for secure communication applications where both confidentiality and visual transparency are required.

3.3. Security Analysis

The proposed hybrid system enhances security by combining cryptographic protection with data hiding techniques, resulting in a multi-layered security mechanism. This approach ensures that both the content of the message and its existence are protected. At the first level, RSA encryption guarantees the confidentiality of the message. The use of a 2048-bit key provides strong resistance against brute-force attacks and modern cryptanalytic techniques. Additionally, the implementation of RSA with OAEP (Optimal Asymmetric Encryption Padding) further strengthens the system by protecting against chosen-plaintext and adaptive chosen-ciphertext attacks. As a result, even if an attacker successfully extracts the embedded data, the ciphertext remains computationally infeasible to decrypt without the private key. At the second level, the LSB steganography technique conceals the presence of the encrypted message within the cover image. Since the embedding process modifies only the least significant bits of pixel values, the visual characteristics of the image

remain nearly identical to the original. This makes it difficult for attackers or observers to detect the existence of hidden data through visual inspection.

The combination of these two techniques provides a significant advantage over standalone methods. In traditional cryptography, the presence of ciphertext may raise suspicion and attract attacks. Conversely, in standalone steganography, extracted data may be readable if not encrypted. The proposed hybrid approach mitigates both issues by encrypting the message before embedding it, thereby ensuring that even if the hidden data is detected, it remains secure. Furthermore, the use of a structured payload, including a fixed-length header, improves the reliability of data extraction and reduces the risk of extraction errors. This contributes to the overall robustness of the system in practical implementations. However, despite its advantages, the system has certain limitations. The LSB method is vulnerable to steganalysis techniques and image processing operations such as compression, resizing, or noise addition, which may alter the embedded bits and lead to data corruption. Therefore, the security of the system depends not only on the strength of encryption but also on the preservation of the stego image. In summary, the proposed RSA–LSB hybrid method provides strong security through encryption and concealment, offering a practical solution for secure message transmission. The dual-layer protection mechanism significantly increases resistance against both cryptographic attacks and detection attempts, making it suitable for applications requiring high levels of confidentiality and stealth.

4. Discussion

The present study demonstrates that the integration of RSA cryptography and Least Significant Bit (LSB) steganography can be effectively implemented as a unified framework for secure message transmission. The overall workflow, starting from key generation, message encryption, ciphertext embedding, data extraction, and message decryption, was successfully executed in an end-to-end manner. This indicates that the proposed system is not only conceptually feasible but also practically implementable using a Python-based environment. From the implementation perspective, the system shows that combining asymmetric encryption with image-based steganography creates a complementary security mechanism. RSA protects the confidentiality of the message content, while LSB hides the existence of the encrypted data within a digital image. This layered design addresses an important limitation of conventional cryptography. In a standard encrypted communication setting, ciphertext is visible and may attract attention because of its random and unreadable form. By embedding the ciphertext into an image, the proposed method reduces the likelihood of suspicion while preserving message confidentiality. This makes the method more suitable for applications where secrecy must involve not only content protection but also concealment of communication itself.

The implementation results further confirm that the use of RSA-OAEP strengthens the encryption stage. OAEP padding improves the security of RSA encryption by providing resistance against several practical attack models that target deterministic encryption schemes. Thus, the proposed method does not rely solely on the mathematical strength of RSA, but also on a more secure padding strategy. In this context, the cryptographic component of the system contributes significantly to the robustness of the overall framework. Even if hidden data were extracted from the stego image, the attacker would still face the computational barrier imposed by RSA decryption without access to the private key. The LSB embedding process also performed effectively in

preserving image appearance. By modifying only the least significant bits of pixel values, the system introduced only minimal visual changes to the cover image. This behavior is consistent with the fundamental characteristic of LSB steganography, which is designed to embed data with low perceptual impact. The obtained image quality metrics support this observation. The low MSE values indicate that the pixel-level distortion between the original image and the stego image is very small. Similarly, the PSNR values above 49 dB show that the visual quality of the stego images remains very high. Since PSNR values above 30 dB are generally considered acceptable in steganographic applications, the results obtained in this study suggest that the proposed method achieves a considerably stronger level of imperceptibility.

These findings are important because image quality is one of the main indicators of steganographic performance. A stego image that shows visible artifacts or distortions may raise suspicion and undermine the hidden communication objective. In this study, the high PSNR values suggest that the embedding of ciphertext, including the additional 32-bit header used for payload length identification, does not significantly degrade the visual integrity of the cover image. This implies that the payload design and embedding strategy are efficient enough to maintain a balance between hidden data insertion and image fidelity. The reliability of the extraction process is another important outcome of this work. The inclusion of a fixed-length header in the payload structure plays a critical role in ensuring accurate extraction. By storing the ciphertext length explicitly, the system is able to identify how many bits must be read from the stego image during the decoding stage. This reduces the risk of incomplete extraction or bit misalignment, both of which could lead to decryption failure. Therefore, the payload structure contributes not only to implementation convenience but also to the functional correctness of the overall system. The successful recovery of the original plaintext after extraction and decryption demonstrates that the combination of header-based payload organization and LSB embedding is reliable for end-to-end communication. In addition to image quality and functional correctness, the security implications of the proposed method deserve further discussion. The dual-layer protection mechanism provides a practical advantage over approaches that use only cryptography or only steganography. In purely cryptographic systems, the content is secure, but the existence of protected communication is obvious. In purely steganographic systems, the message may remain hidden, but once extracted it may be directly readable if no encryption is applied.

The present method addresses both weaknesses simultaneously. The message remains hidden within an image, and even if the hidden payload is identified, it remains encrypted. This layered protection significantly increases the difficulty of unauthorized access and interpretation. However, the results also imply that the effectiveness of the system depends on the condition of the stego image. Because the proposed method relies on spatial-domain LSB embedding, the hidden data are vulnerable to image manipulations that alter pixel values. Compression, resizing, filtering, or noise addition may damage the embedded bits and affect extraction accuracy. This means that the method performs best when the stego image is preserved in a lossless format such as PNG, as implemented in this study. The requirement of a lossless storage format is therefore not merely a technical choice, but an essential condition for maintaining data integrity. This characteristic should be considered when evaluating the method for practical deployment. Another point of discussion concerns payload capacity. The embedding capacity of the system is determined by the dimensions of the cover image and the number of channels used during LSB insertion. Since one bit is embedded per RGB channel, the image capacity is finite

and directly related to image size. This creates a trade-off between payload size and imperceptibility. Embedding larger ciphertexts requires more modified pixels, which may increase distortion and potentially affect detectability. Although the experimental results in this study indicate that the payload used did not compromise visual quality, the dependence on image capacity remains an inherent limitation of spatial-domain steganography. Therefore, cover image selection is an important practical factor in applying the proposed method. The user interface implementation also contributes to the significance of this work. By developing the application with Streamlit, the system becomes more accessible to users who may not be familiar with command-line execution or low-level programming. The modular interface, consisting of Generate Keys, Encrypt & Embed, Extract & Decrypt, and Metrics pages, reflects the logical stages of the proposed method and makes the workflow easier to understand.

This is especially relevant for educational, demonstrative, and prototype development purposes. The interface supports not only technical functionality but also usability, which is important when translating a theoretical method into a practical application. The results of this study are also consistent with the general direction of previous research that combines cryptography and steganography to improve secure communication. However, the present work contributes by providing an integrated implementation that includes secure RSA-OAEP encryption, structured payload design, image quality evaluation, and a functional graphical interface within a single framework. The high PSNR values obtained in the experiments reinforce the argument that the proposed approach preserves image quality effectively while maintaining message confidentiality. Thus, the study supports the view that hybrid security methods can offer better overall protection than single-technique approaches. Overall, the discussion of the implementation, evaluation metrics, extraction reliability, and security characteristics indicates that the proposed RSA-LSB framework performs well as a prototype for secure digital communication. The system achieves functional correctness, high image fidelity, and layered protection of secret messages. At the same time, the discussion highlights important operational considerations, including the dependence on lossless image preservation, payload limitations, and vulnerability to image modification. These aspects are essential in understanding the actual performance and applicability of the proposed method beyond its basic implementation results.

5. Conclusion

This study proposes a hybrid method that integrates RSA cryptography and Least Significant Bit (LSB) steganography to enhance the security of digital message transmission. The implementation results demonstrate that the proposed system is capable of performing secure end-to-end communication, where messages can be successfully encrypted, embedded into an image, extracted, and decrypted without loss of information. The evaluation results indicate that the method maintains high image quality, as evidenced by very low MSE values and PSNR values exceeding 49 dB. These findings confirm that the embedding process introduces minimal visual distortion, ensuring that the stego image remains visually indistinguishable from the original image. In addition, the combination of RSA encryption and LSB steganography provides a dual-layer security mechanism, where the message is both protected and concealed, significantly improving resistance against unauthorized access and detection. Despite these advantages, the proposed method has certain limitations, particularly its sensitivity to image processing operations such as compression, resizing, and noise, which may affect the

integrity of the embedded data. Furthermore, the embedding capacity is limited by the size of the cover image, which may restrict the amount of data that can be securely transmitted. Future research may focus on improving the robustness of the system by incorporating transform-domain steganography techniques, such as Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT), which are more resistant to image manipulation. Additionally, adaptive embedding methods and machine learning-based approaches can be explored to enhance both security and imperceptibility. Further evaluation using larger datasets and various image formats is also recommended to provide a more comprehensive analysis of system performance. Overall, the proposed RSA–LSB hybrid method offers an effective and practical solution for secure communication, balancing confidentiality, imperceptibility, and implementation simplicity, while providing a strong foundation for future advancements in information security.

References

[1]. W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed. Pearson, 2017.

[2]. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography. CRC Press, 1996.

[3]. R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” Commun. ACM, vol. 21, no. 2, pp. 120–126, 1978.

[4]. B. Schneier, Applied Cryptography. Wiley, 1996.

[5]. N. Provos and P. Honeyman, “Hide and seek: An introduction to steganography,” IEEE Security & Privacy, vol. 1, no. 3, pp. 32–44, 2003.

[6]. J. Fridrich, Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press, 2009.

[7]. K. K. B. Gupta and S. K. Sharma, “A survey on steganography techniques,” Int. J. Comput. Appl., vol. 87, no. 14, 2014.

[8]. I. W. Mulyono, Y. Kusumawati, and N. K. Ningrum, “Analysis of image quality using LSB-based steganography and cryptography combination,” J. Informatics, 2023.

[9]. M. Rizky and D. I. G. Hts, “Combination of RSA and LSB for secure text embedding in images,” J. Info Digit, vol. 1, no. 3, pp. 1129–1142, 2023.