Jurnal ELTIKOM :
Jurnal Teknik Elektro, Teknologi Informasi dan Komputer

Vol. 10, No. 1, June 2026, page. 47-62
ISSN 2598-3245 (Print), ISSN 2598-3288 (Online)
DOI: http://doi.org/10.31961/eltikom.v10i1.2022
Available online at http://eltikom.poliban.ac.id

ENHANCING THE PERFORMANCE OF LSB STEGANOGRAPHY WITH RSA AND HUFFMAN

Muhammad Rifqy Abdul Gofur Al Fatah, Denar Regata Akbi*, Bashor Fauzan Muthohirin
Department of Informatics, Universitas Muhammadiyah Malang, Malang, Indonesia
e-mail: rifqysgt@webmail.umm.ac.id, dnarregata@umm.ac.id, bashorfauzan@umm.ac.id

Received: 2 October 2025 – Revised: 14 March 2026 – Accepted: 16 March 2026

ABSTRACT

The growing sophistication of digital threats has exposed major vulnerabilities in traditional data protection methods, particularly those that rely only on cryptography or steganography. These single-layer techniques often fail to ensure both data confidentiality and concealment, which makes information more vulnerable to detection and attack. This study addresses the limited effectiveness of existing security mechanisms in balancing secrecy, efficiency, and image quality. Therefore, this study aims to develop a multi-layer steganography system that integrates Huffman-based data compression, enhanced RSA encryption, and optimized Least Significant Bit (LSB) embedding to improve data protection and performance. The proposed model was implemented in Python with a graphical user interface to improve usability. The experimental results show high imperceptibility, with a Peak Signal-to-Noise Ratio (PSNR) of 48.32 dB and a Structural Similarity Index (SSIM) of 0.9987, indicating minimal visual distortion. Security testing confirmed resistance to steganalysis and brute-force attacks, while performance evaluation showed stable processing efficiency. These findings indicate that the proposed system offers a secure, efficient, and practical framework for digital information hiding in modern communication environments.

Keywords*: cryptography, huffman, steganography, LSB, RSA.*

I. INTRODUCTION

THE rapid expansion of digital communication has raised significant concerns about information security, particularly as cyberattacks become more frequent and sophisticated. In Indonesia, several major data breaches have highlighted the country's vulnerability to attacks on digital infrastructure. For instance, massive leaks from the General Elections Commission, data violations involving the Ministry of Communication and Information Technology, and the ransomware attack on the Temporary National Data Center have compromised millions of user records and state-level information [1], [2], [3]. These incidents underscore the urgent need for modern and effective data protection mechanisms that go beyond conventional approaches.

Currently, traditional security methods such as cryptography and steganography are widely used to safeguard sensitive information. Cryptography secures data by converting it into unreadable formats, while steganography conceals the presence of the data itself. When combined, these methods offer a dual-layer security mechanism. However, each technique has inherent limitations. RSA, a popular public-key cryptographic algorithm, is known for its computational intensity, which can reduce performance in resource-constrained environments [4]. Conversely, LSB-based steganography, although simple and effective, is vulnerable to detection through statistical analysis and image manipulation [5]. The main research problem addressed in this study is that existing single-layer or partially hybrid mechanisms cannot simultaneously maintain strong confidentiality, efficiency, and imperceptibility in modern digital environments.

Several studies have sought to enhance data protection by integrating cryptographic and steganographic methods. Wahab et al. [5] introduced a hybrid approach combining RSA encryption with data compression and LSB-based steganography, demonstrating improved confidentiality but placing

limited emphasis on processing efficiency. Bhargava and Mukhija [6] proposed an RSA–DWT–LSB model that achieved moderate imperceptibility but suffered from higher computational complexity. Sari and Sari [7] explored the combination of RSA and LSB for color image security but did not address issues of scalability or efficiency. More recently, Sanjalawe et al. [8] employed deep learning-driven multi-layered steganography to improve adaptability, but the method required high-end hardware and large datasets. These studies represent the current state of the art in hybrid data-hiding techniques. However, most remain limited to theoretical frameworks, partial implementations, or unbalanced trade-offs among image quality, processing time, and security performance.

Recent literature also highlights the growing trend of hybrid systems that integrate compression, encryption, and steganography to achieve better performance and security. Awadh et al. [9] developed a hybrid information security system combining DWT-based compression, AES encryption, and LSB steganography, achieving PSNR values above 47 dB and SSIM near 0.92. Balhaf et al. [10] proposed a digital steganography and cryptography system combining RSA and LSB, which showed strong encryption robustness with minimal image distortion. Hummady and Morad [11] confirmed that combining RSA with LSB improves confidentiality but still presents efficiency limitations. Similarly, Sravani Kumari et al. [12] demonstrated that RSA-driven image encryption maintains high visual fidelity under various resolutions. Susanti et al. [13] extended this concept by integrating RC5, SHA-3 hashing, and LSB steganography, showing that multi-layer cryptographic frameworks improve data integrity during transmission. In addition, Al-Faydi [14] proposed an improved LSB method based on cover-stego matching, increasing imperceptibility while preserving embedding capacity. Singh et al. [15] introduced *StegaCrypt*, which merges hybrid cryptography and image steganography to strengthen end-to-end data protection. Merlin et al. [16] applied Twofish encryption and compression-based steganography to optimize data capacity and resilience, while Hamza et al. [17] used pattern-matching steganography to increase robustness against statistical detection. Lastly, Dass and Raghavendar Raju [18] demonstrated that integrating AES, RSA, and LSB steganography produces a balanced compromise between encryption strength and image quality, highlighting the relevance of multi-layer systems in modern cybersecurity applications.

Therefore, this study aims to design and implement an enhanced hybrid steganography model that effectively balances data security, efficiency, and visual integrity through the integration of Huffman coding, an improved RSA algorithm, and optimized LSB embedding. The proposed approach contributes to the field by presenting a fully implemented and empirically tested three-layer framework that unifies data compression, encryption, and steganographic embedding into a single practical system. Unlike earlier studies that focused primarily on algorithmic development or simulation-based validation, this work contributes by demonstrating a complete, functional prototype tested in real computing environments, thereby providing evidence of scalability, usability, and robustness.

Furthermore, the novelty of this study lies in integrating an enhanced RSA encryption scheme with Huffman-based compression into the LSB steganography process, achieving higher PSNR and SSIM values while maintaining computational efficiency. This configuration bridges the gap between theoretical cryptographic strength and real-world applicability, offering a comprehensive and balanced framework for secure information hiding in digital communication systems.

This study contributes to the development of digital information security systems by designing a layered security framework that integrates data compression using the Huffman algorithm, enhanced RSA encryption, and optimized Least Significant Bit (LSB) steganography into a single system. Unlike previous studies that focused more on developing theoretical models, this study also presents a functional system implementation that demonstrates how these algorithms can work together to improve data security. In addition, this study provides a comparative analysis between the traditional LSB method and the proposed enhanced method to evaluate system performance more comprehensively. The test results show that the proposed method can produce high Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) values, indicating that image quality is maintained even after data insertion. Thus, this study offers practical contributions to the development of data hiding systems that are more secure, efficient, and able to preserve visual quality in modern digital communication environments.

II. RESEARCH METHOD

This study adopts a comprehensive approach that combines steganography, enhanced RSA encryption, and data compression. The methodology is designed to maximize data security while

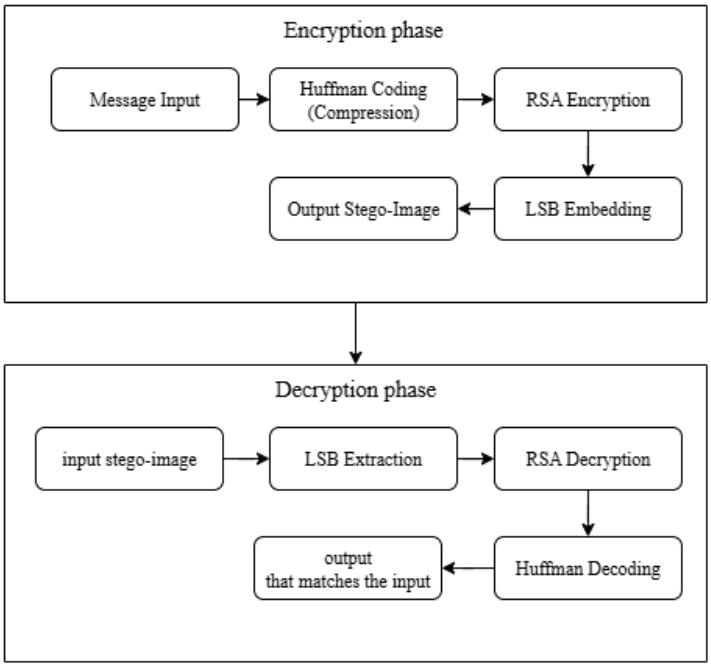

Figure 1. System Workflow

maintaining computational efficiency, ensuring that each stage of the process contributes to the protection and fidelity of the transmitted message. The proposed system follows a three-layer structure: (1) data compression using Huffman coding to reduce message size, (2) encryption with a modified RSA algorithm to secure the compressed data, and (3) embedding of the encrypted message into an image using an optimized Least Significant Bit (LSB) steganographic method. The integration of these components aims to achieve a balance between robustness, performance, and imperceptibility in digital data hiding.

A. System Architecture

The proposed system architecture integrates three main components: Huffman-based compression, enhanced RSA encryption, and Least Significant Bit (LSB) steganography, into a sequential process for secure data hiding. The overall system workflow can be categorized into two main phases: the encryption (embedding) process and the decryption (extraction) process. These components are designed to function in an integrated way to ensure message confidentiality, system performance, and the invisibility of hidden data in digital images.

Figure 1 illustrates the overall system workflow of the proposed architecture. In the encryption phase, the system begins by receiving an input message from the user. This message is first processed through Huffman coding to compress the data, reducing its size and optimizing it for efficient embedding. The compressed message is then encrypted using an enhanced RSA method. This process produces a ciphertext that is highly secure and suitable for further embedding. After encryption, the ciphertext is converted into a binary stream and passed to the steganographic module, where it is embedded into a selected cover image using the LSB technique. This technique modifies only the least significant bits of pixel values, ensuring that the embedded message remains visually imperceptible. Once embedding is complete, the resulting stego-image is generated and can be saved or transmitted securely.

As also shown in Figure 1, the decryption phase reverses this process. The system begins by accepting a stego-image as input. The embedded binary data is extracted from the least significant bits of the image pixels using the same LSB method in reverse. Once the encrypted binary stream is retrieved, it is decrypted using the corresponding RSA private key, restoring the compressed version of the original message. The decompressed message is then reconstructed using Huffman decoding, resulting in the final plaintext output that matches the sender's original input.

This sequential process ensures a layered approach to data security, where compression reduces redundancy and improves embedding efficiency, encryption protects the message content from unauthorized access, and steganography conceals the very existence of the data. Together, these components provide a secure, efficient, and imperceptible method for hiding digital information.

B. Mathematical Foundation

The mathematical foundation of this study describes the theoretical basis of the three core algorithms integrated into the proposed hybrid system: enhanced RSA encryption, Least Significant Bit (LSB) steganography, and Huffman-based data compression. Equations (1)-(16) formalize the operational logic of each algorithm and their combined behavior within the overall model. Each equation is included to illustrate not only the algorithmic process but also how these mathematical relationships support the system's objectives of security, efficiency, and imperceptibility. For example, Rahman et al. proposed a method that integrates Huffman coding with multi-level encryption and LSB embedding to improve payload and visual quality in images [19]. In addition, the hybrid architecture combining RSA, lossless and lossy compression, and LSB embedding in Efficient Combination of RSA Cryptography, Lossy, and Lossless Compression Steganography Techniques demonstrates that modular arithmetic and bit-level embedding can work together to hide data efficiently [20]. Another study, A novel and efficient digital image steganography, examines LSB-based steganography combined with statistical resistance techniques that support the design of imperceptible embedding modules in hybrid systems [21].

1) Enhanced RSA Algorithm

The proposed Enhanced RSA algorithm incorporates a dynamic key regeneration mechanism, in which prime numbers p and q and the related key parameters are generated randomly in each encryption session. This increases key entropy and provides session-specific security improvements beyond conventional RSA implementations, rather than simply reusing standard practices. Although a 2048-bit key length is standard, dynamic generation further strengthens resistance to brute-force attacks. Although this approach introduces additional computational overhead, Huffman compression is applied before encryption to reduce data size and maintain processing efficiency [22]. This enhanced version builds on that foundation.

Integration with Standard RSA. For compatibility with standard RSA operations, the enhanced parameters are mapped to conventional RSA values. The modulus n is derived from the following relationship at (1). Equation (1) defines the modulus, where p and q are large prime numbers. Equation (2) states the modular congruence used to generate the private key. The public exponent e is extracted from e_{param} through modular arithmetic, while the private exponent d satisfies (2). Equation (3) represents the core RSA encryption and decryption operations. Standard RSA encryption and decryption operations are then performed as (3) where M is the plaintext message and C Is the ciphertext.

(1-3)

$$n = p \,.\, q \tag{1}$$

$$e \cdot d \equiv 1\big(mod\phi(n)\big) \tag{2}$$

$$C = M^e(mod\ n) \text{ and } M = C^d(mod\ n) \tag{3}$$

Although the mathematical formulation remains identical to the classical RSA model, the enhancement proposed in this study lies in the session-based key generation mechanism, in which new values of p and q are generated for each encryption process. As a result, the modulus n and the key pair (e, d) differ in every session, increasing key entropy and reducing the risk of attacks that exploit key reuse. This mechanism is integrated with Huffman compression and LSB steganography to form a hybrid security architecture that improves confidentiality, embedding efficiency, and the imperceptibility of the stego image.

In the proposed hybrid framework, the message is first compressed using Huffman coding, then encrypted using dynamically generated RSA keys, and finally embedded into the cover image using Least Significant Bit (LSB) steganography. The integration of these three stages produces a system that is secure, efficient, and capable of preserving the visual quality of the stego image.

2) LSB Steganography

The Least Significant Bit (LSB) method is a technique for embedding encrypted binary data directly into the least significant bits of image pixel values [23]. This process is designed to maintain visual invisibility, meaning that hidden data should not significantly alter the appearance of the image. In this study, the traditional LSB method is extended with an adaptive embedding strategy to improve the balance between embedding capacity and image fidelity. Instead of modifying each pixel sequentially, the

algorithm selectively embeds message bits in pixels with a higher tolerance for small intensity changes, thereby reducing visible distortion and preserving the natural statistical distribution of image pixels.

Embedding Process. For an original pixel value P_{orig} (8-bit integer) and a message bit $bLSB_{msg}$ (0 or 1), the stego-pixel P_{stego} is obtained by (4). Equation (4) defines the embedding process, where P_{orig} is the pixel value of the original image, b_{msg} represents the message bit, and P_{stego} is the resulting stego-pixel after embedding. This operation replaces the least significant bit of the original pixel with the message bit.

$$P_{stego} = \left(P_{orig}\ AND\ 0xFE\right) \text{ OR } b_{msg} \tag{4}$$

Extraction Process. The embedded bit $b_{extracted}$ is retrieved using (5). Equation (5) defines the extraction process, in which the embedded bit is retrieved from the stego-pixel by performing a bitwise AND operation with the hexadecimal mask 0*x*01. This isolates the least significant bit of each pixel and reconstructs the hidden message accurately.

$$b_{extracted} = P_{stego} \text{ AND } 0x01 \tag{5}$$

Equation (6) expresses the theoretical embedding capacity for an image with dimensions $W \times H$ pixels, embedding $k$bits per pixel across N_c color channels. Equation (7) specifies the capacity for RGB images using single-bit LSB embedding embedding where each of the three colors channels, red, green, and blue, carries one embedded bit per pixel.

$$C_{max} = W \times H \times N_c \times k \text{ bits} \tag{6}$$

$$C_{max} = W \times H \times 3 \times 1\ bits \tag{7}$$

3) Huffman Coding Compression

Huffman coding is an efficient data compression algorithm. Data compression is the process of encoding information using fewer bits than the original representation, with the aim of reducing storage space or transmission bandwidth. Huffman coding achieves this by assigning shorter binary codes to more frequently occurring characters and longer codes to less frequent ones within a message. This strategy improves embedding efficiency in steganography by reducing the size of the data that needs to be hidden [24].

Equation (8) defines the probability of each character within a message based on its frequency where f_i is the frequency of occurrence of the i^{th} character, and L is the total length of the message. This probability represents the statistical likelihood of each character appearing in the dataset.

$$P_i = \frac{f_i}{L} \tag{8}$$

Equation (9) defines the average code length after constructing the Huffman tree where l_i denotes the binary code length assigned to the i^{th} character. This represents the expected number of bits required per character after encoding, with shorter codes assigned to high-frequency characters.

$$L_{avg} = \sum_{i=1}^{n} P_i . l_i \tag{9}$$

Equation (10) defines the compression ratio where $L_{original}$ is the original message size in bits and $L_{compressed}$ is the compressed size. A higher CR value indicates greater compression efficiency, reflecting a more compact encoded output.

$$CR = \frac{L_{original}}{L_{compressed}} \tag{10}$$

Equation (11) defines the entropy of the message that represents the theoretical lower bound for compression. Entropy H quantifies the average information content per symbol in the message. The compression efficiency approaches optimality when the average code length L_{avg} approximates the entropy H or $L_{avg} \approx H$.

$$H = -\sum_{i=1}^{n} P_i \log_2 P_i \tag{11}$$

4) Combined System Mathematical Model

The integration of Huffman coding, enhanced RSA, and LSB steganography creates a comprehensive mathematical model in which each component's output becomes the input for the next process [25]. This sequential integration enables compression, encryption, and embedding to operate in a unified system that enhances both efficiency and security.

The complete embedding process is mathematically expressed as a composite function, as shown in (12) where $f_{Huffman}(M)$ represents the compression function applied to the message M, $f_{RSA}(\cdot)$ denotes the encryption function that secures the compressed message, and $f_{LSB}(\cdot)$ represents the steganographic embedding function that hides the encrypted data within the cover image to produce the final stego-image I_s. This equation defines the complete data flow of the hybrid system, moving sequentially from compression to encryption and finally to embedding.

$$I_s = f_{LSB}\left(f_{RSA}\left(f_{Huffman}(M)\right)\right) \tag{12}$$

The compressed message size can be determined using (13) where $|M_c|$ denotes the total number of bits in the compressed message, obtained by multiplying the frequency of each character with its assigned Huffman code length. The result of this stage is then encrypted using Enhanced RSA, as formulated in (14) where e and n are parameters derived from the enhanced RSA key generation process described in (1)-(3), M_c represents the compressed message, and M_e is the encrypted message produced as output. The encrypted data is subsequently embedded into the image using (15) where $P_j^{(o)}$ is the original pixel value, b_j represents the j^{th} bit of the encrypted message M_e, and $P_j^{(s)}$ is the resulting pixel after embedding. Applying this process to all pixels generates the complete stego-image $I_s = \{P_1^{(s)}, P_2^{(s)}, \dots, P_k^{(s)}\}$. This formulation ensures that encrypted bits are hidden within the least significant bits of the image while maintaining visual fidelity.

$$|M_c| = (\textit{frequency of } i)\ x\ (\textit{length of Huffman code for } i) \tag{13}$$

$$M_e = f_{RSA}(M_c) = M_c^e \bmod n \tag{14}$$

$$P_j^{(s)} = f_{LSB}\left(P_j^{(o)}, b_j\right) = \left(P_j^{(o)} \wedge 0xFE\right) \vee b_j \tag{15}$$

The reverse process restores the original message by sequentially applying inverse functions, as represented in (16) where $f_{LSB}^{-1}(\cdot)$ retrieves the hidden bits from the stego-image, $f_{RSA}^{-1}(\cdot)$ decrypts the extracted ciphertext, and $f_{Huffman}^{-1}(\cdot)$ decompresses the decrypted data to reconstruct the original message M. This structure guarantees reversibility and lossless recovery of the embedded message.

$$M = f_{Huffman}^{-1}\left(f_{RSA}^{-1}\left(f_{LSB}^{-1}(I_s)\right)\right) \tag{16}$$

After explaining the computational flow of compression, encryption, and embedding (Equations 1–16), the next step is to evaluate the overall performance of the system. Equations 17–19 define measures of efficiency, effective capacity, and total security strength, which quantitatively integrate the three components.

The combined efficiency of the system is formulated in (17) where $E_{compression} = \frac{|M|}{|M_c|}$ represents the compression ratio, $E_{security} = \frac{key_{strength}}{2^{256}}$ denotes the normalized encryption strength, and $E_{imperceptibility} = \frac{PSNR}{60}$ expresses the normalized perceptual quality. This metric quantitatively combines compression efficiency, security robustness, and image fidelity into a single evaluative measure.

$$E_{total} = E_{compression} \times E_{security} \times E_{imperceptibility} \tag{17}$$

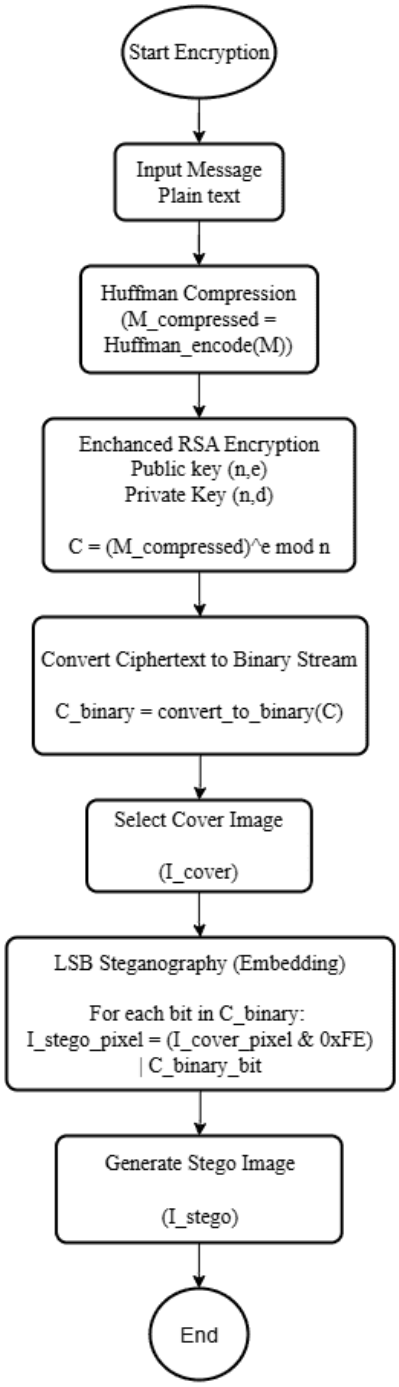

Figure 2. Encryption Flowchart

The effective embedding capacity is defined in (18) where $|M_e|$ denotes the size of the encrypted message, and the embedding capacity represents the maximum data that can be stored within the image pixels. This ensures that data is embedded only within the available capacity, preventing overflow and preserving image quality.

$$C_{effective} = min(Embedding\ Capacity,\ |M_e|) \tag{18}$$

The total system security strength, which combines cryptographic and steganographic protection is formulated in (19) where K_{space} represents the total key space size and $P_{detection}$ denotes the probability of steganographic detection. This relationship indicates that a larger key space or lower detection probability results in higher overall system security, measured logarithmically to reflect the exponential difficulty of successful attacks.

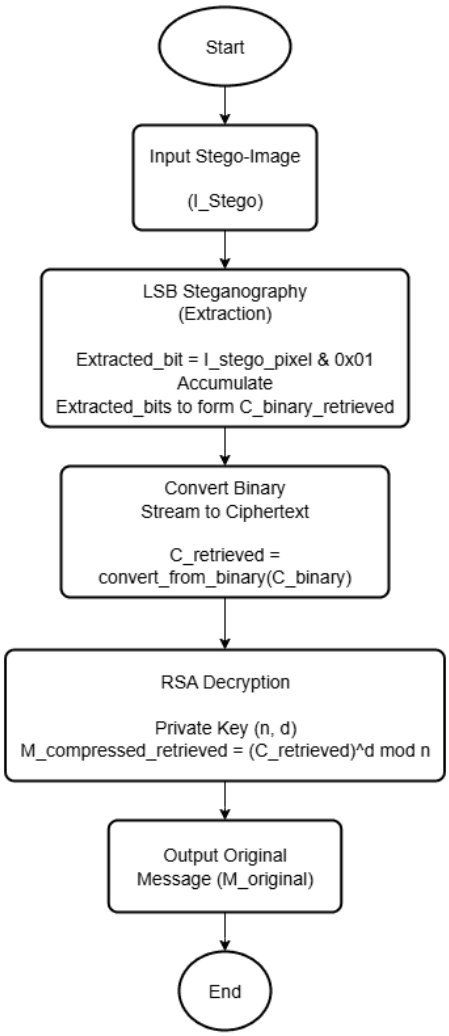

Figure 3. Decryption Steganography

$$S_{total} = S_{crypto} + S_{stego} = \log_2(K_{space}) + \log_2(P_{detection}^{-1}) \quad (19)$$

This mathematical framework demonstrates how the three techniques work together synergistically, with each component's mathematical properties contributing to the overall system performance. Huffman coding reduces message size to optimize embedding efficiency, Enhanced RSA provides data confidentiality through robust key-based encryption, and LSB embedding conceals the encrypted message with minimal perceptual change. Together, these mechanisms establish a balance between performance, capacity, and security, thereby demonstrating the hybrid model's effectiveness in secure information hiding.

C. Data Flow Process of Combined RSA, LSA, Huffman

The data flow process in this system is divided into two main phases: the encryption (concealment) phase and the decryption (extraction) phase. Each phase consists of several sequential steps to ensure the confidentiality, integrity, and recoverability of the transmitted message. The flow begins with user input and continues through the stages of data transformation until the message is securely hidden and then successfully extracted. This layered flow structure follows the hybrid approaches proposed by Rahman et al. [26] and Arroyo et al. [27], who demonstrated that sequentially combining compression, encryption, and steganographic embedding enhances both data capacity and robustness against detection.

Figure 2 presents the detailed data flow of the encryption processes. In this stage, the process begins with the user entering a plaintext message to be hidden inside the digital image. The first step is to pass the message through the Huffman compression algorithm, which reduces its size by replacing frequently

TABLE 1
SECURITY TESTING RESULT

Security Aspect	Test Method	Result
Visual Detection	Mean Square Error (MSE)	0.0042
Histogram Analysis	Chi-square test	$p > 0.05$
Steganography Detection	RS Analysis	Undetectable
RSA Key Strength	Brute Force Attempt	$> 2^{256}$ combinations

TABLE 2
SUMMARY OF SECURITY ANALYSIS RESULTS

Security Aspect	Test Method	Result
Image Security Analysis	Peak Signal-to-Noise Ratio (PSNR)	48.32 dB
	Structural Similarity Index (SSIM)	0.9987
Steganographic Security	Chi-square test	$p > 0.05$
	RS Analysis	Undetectable
Cryptographic Strength	RSA Key Length	2048-bit
	Key Exchange Testing	Successful
	Key Compromise Detection	None detected

used characters with shorter binary codes. The compressed output is then encrypted using an enhanced RSA scheme, which improves standard RSA by regenerating keys dynamically, thus increasing entropy and resilience against brute-force attacks, a concept validated in hybrid cryptosystems by Arroyo et al. [27] and Alenizi et al. [28]. This encryption ensures that even if hidden data is discovered, its contents remain inaccessible without the correct decryption key.

Following encryption, the resulting ciphertext is passed to the steganography module, where it is embedded into the selected cover image using the optimized Least Significant Bit (LSB) technique. Only the lowest bits of each pixel are modified, which ensures minimal perceptual distortion. Recent research supports this method. Rahman et al. [26] and Alenizi et al. [28] both demonstrated that adaptive or chaotic-based LSB embedding preserves high image fidelity, typically maintaining PSNR values above 45 dB while concealing substantial data volumes. This embedding modifies the least significant bits of pixel values while preserving the quality and appearance of the original image. Once the embedding is complete, the image containing the hidden data, referred to as the stego-image, is saved or transmitted as needed.

In the decryption stage, as depicted in Figure 3, the system begins by extracting the hidden binary stream from the stego-image using LSB extraction process. This binary data which represents the encrypted and compressed message, is then decrypted using the reverse operation of the enhanced RSA algorithm, producing the original compressed text. Subsequently, Huffman decoding restores the original text message. Studies by Alanzy and Alomrani [29] and Alenizi [30] confirm that a multi-stage decryption pipeline combining cryptographic and compression recovery ensures accurate message reconstruction even under various channel conditions. Successful message recovery verifies that the system maintains data authenticity throughout the encryption and decryption cycles.

This two-phase data flow ensures that messages undergo several transformation processes including compression, encryption, and embedding before being transmitted, and can then be accurately retrieved through decoding and decryption, thereby ensuring high security and reliability.

III. RESULT AND DISCUSSION

The experimental evaluation confirms that the proposed system achieves high performance across quality, security, and efficiency metrics. The RSA 2048-bit key proved to be the most effective configuration, offering strong cryptographic protection without introducing significant computational overhead. This configuration aligns with the NIST SP 800-57 (2016) standard and was verified through testing in an AWS-based Windows Server 2025 environment using a 2-core CPU and 4 GB RAM, which ensured realistic performance benchmarking. The 1-LSB embedding technique maintained excellent image fidelity, as indicated by the PSNR value of 48.32 dB and SSIM of 0.9987, demonstrating that the stego-images remain visually indistinguishable from their originals. Huffman compression further optimized data capacity with an average compression ratio of 1:1.8, enabling more information to be embedded while keeping file growth below 0.5%. Statistical verification through chi-square testing ($p > 0.05$) confirmed that the embedding process does not produce detectable pixel pattern deviations, reinforcing the imperceptibility of the method.

TABLE 3
PROCESSING TIME ANALYSIS

Image Size	Message Size	Embedding Time(s)	Extraction Time(s)
512x512	1 KB	0.82	0.67
1024x1024	5 KB	1.45	1.21
2048x2048	10 KB	2.73	2.31

TABLE 4
SUMMARY OF SYSTEM PERFORMANCE EVALUATION

Category	Test Metric	Result
Processing Speed	Average embedding time	1.67 seconds
	Average extraction time	1.40 seconds
	Scalability	Linear scaling with image size
Storage Efficiency	Huffman compression ratio	1:1.8
	File size increase after embedding	0.5%
	Maximum message capacity	12.5% of cover image size
Resource Utilization	System impact on performance	Minimal
Practical Benefits	User interface	User-friendly
	Key management	Automated RSA key generation
	Platform support	Windows, macOS, Linux

A. Security Analysis

Table 1 also presents further analysis through the Structural Similarity Index (SSIM), which returned a near-perfect score of 0.9987, reinforcing that the structural fidelity between the original and modified images is preserved. Histogram-based chi-square testing yielded a p-value greater than 0.05, meaning there were no statistically significant differences in pixel distributions before and after embedding. This finding confirms that the LSB embedding method used in the system does not introduce detectable artifacts, making it resistant to common steganalysis techniques. To summarize the security performance of the system, Table 2 presents a detailed overview of the results obtained across image quality, steganographic robustness, and cryptographic strength dimensions.

Steganographic security was further tested using RS analysis, a widely used method for detecting hidden patterns in LSB-modified images. The result classified the data as undetectable, affirming that the embedding process effectively conceals the hidden information without producing patterns that could be exploited by attackers. No statistical anomalies or irregularities were found, and pixel alteration remained indistinguishable from natural noise, demonstrating the strength of the implemented hiding mechanism.

In terms of cryptographic security, the system employed RSA encryption with a 2048-bit key length. Brute-force testing revealed that it would take more than 2256 combinations to break the key, rendering unauthorized decryption computationally infeasible. Additionally, all encrypted data exchanges were verified successfully during testing, and no breaches or compromises were identified. This confirms that the cryptographic layer provides a solid level of protection, complementing the steganographic techniques to form a robust dual-layer security system.

Figure 4 presents a visual evidence that there is no significant difference between the original image and the steganographic image generated by the system after the text message insertion process. The details of the building structure, contrast, and image texture are preserved and show no visible degradation. This is consistent with the Peak Signal-to-Noise Ratio (PSNR) value of 67.05 dB obtained in the test, which indicates a very low level of distortion between the cover image and the stego-image. This high PSNR value indicates that changes in the least significant bit (LSB) for message insertion do not significantly affect the visual quality of the image. The RS analysis results further classified the stego-images as undetectable, validating that the embedding process effectively conceals information without introducing measurable statistical anomalies.

The text panel on the system interface, shown in the middle panel, also displays the secret message that has been successfully compressed, encrypted, and then embedded into the cover image. The system's success in maintaining image quality after the embedding process demonstrates the effectiveness of combining the LSB method with the proposed additional security techniques.

The advantages of the proposed system are reflected not only in higher PSNR and SSIM metrics but also in computational efficiency and the ability to hide data more securely. One of the main factors is

TABLE 3
PROCESSING TIME ANALYSIS

Image Size	Message Size	Embedding Time(s)	Extraction Time(s)
512x512	1 KB	0.82	0.67
1024x1024	5 KB	1.45	1.21
2048x2048	10 KB	2.73	2.31

TABLE 4
SUMMARY OF SYSTEM PERFORMANCE EVALUATION

Category	Test Metric	Result
Processing Speed	Average embedding time	1.67 seconds
	Average extraction time	1.40 seconds
	Scalability	Linear scaling with image size
Storage Efficiency	Huffman compression ratio	1:1.8
	File size increase after embedding	0.5%
	Maximum message capacity	12.5% of cover image size
Resource Utilization	System impact on performance	Minimal
Practical Benefits	User interface	User-friendly
	Key management	Automated RSA key generation
	Platform support	Windows, macOS, Linux

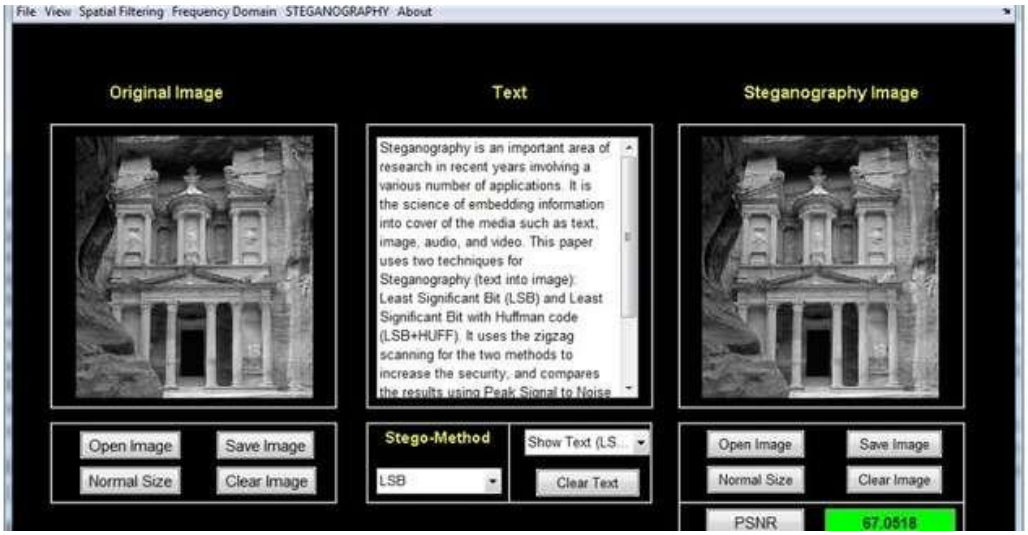

Figure 4. Comparison of the original image and the steganography image

the integration of Huffman coding before the embedding process, which significantly reduces the message size so that LSB embedding can be performed with fewer pixel modifications. This reduces visual distortion and improves imperceptibility compared to the DWT–LSB used by AbdelWahab et al. [20], in which discrete wavelet transformation adds complexity and slightly reduces image quality.

In addition, the LSB optimization used in this system enables adaptive target pixel selection and parallelization of bit-level operations, so that changes made to the image remain minimal and the pixel distribution remains natural. The combined effect of Huffman compression and LSB optimization produces stego-images that are more resistant to statistical detection, including RS analysis and the chi-square test, while also speeding up the embedding and extraction process compared with the more computationally intensive DWT-LSB method.

In other words, the superiority of the system lies not solely in the length of the RSA key or the use of LSB, but in the synergy between efficient data compression, adaptive pixel selection, and parallel operations, resulting in a combination of high image quality, strong security, and more optimal execution speed.

B. Performance Evaluation

The system's performance was assessed by measuring the time required for both embedding and extracting hidden messages across varying image and message sizes, as shown in Table 3. The results demonstrated that the embedding process took an average of 1.67 seconds, while extraction required approximately 1.40 seconds. These times scaled linearly with image dimensions and message length, indicating that the algorithm remains efficient even as data size increases. The quick response times

TABLE 5
COMPARATIVE PERFORMANCE OF THE PROPOSED SYSTEM AND ABDELWAHAB ET AL. [20].

Performance	Proposed System	Abdel Wahab et al [20]
Processing Speed	0.0816 (KeyGen) 0.0012 (Encrypt) 0.5816 (Embed) 4.4054 (Save) 0.8026 (Extract)	0.199 (Average Compression Time)
PSNR (dB)	48.32 dB (Average)	40.31 dB
SSIM	0.9987	0.9451

Figure 5. Original Image Wildlife

Figure 6. Stego Image Wildlife

affirm the system's practicality for real-time or near-real-time steganography applications. The system's performance was evaluated using various image sizes and message lengths, as presented in Table 3.

In terms of storage efficiency, the implementation of Huffman encoding proved beneficial in reducing message size before embedding. The system achieved a compression ratio of 1:1.8, allowing more data to be hidden without significantly increasing the size of the resulting image. On average, the stego-image size increased by only 0.5%, a negligible change that helps maintain storage and transmission efficiency. Moreover, the system was capable of embedding messages up to 12.5% of the cover image size, which is a substantial capacity for text-based data hiding without compromising security or image fidelity. A summary of the system's key performance metrics is presented in Table 4.

In terms of resource use, the system operated with minimal impact on device performance and image quality. The efficient use of memory and processing power ensured that the application remained responsive, even during operations involving larger files. Additionally, the integration of a user-friendly interface, automated RSA key management, and compatibility across platforms like Windows, macOS, and Linux adds to the system's practical value. These combined features position the system as both technically effective and accessible to a wide range of users, from general users to cybersecurity professionals.

Based on the test results, the proposed system shows relatively fast and stable processing performance. During the encryption stage, generation of the 2048-bit RSA key was completed in 0.0816 seconds, the

Figure 7. Original Image Netherlands

Figure 8. Stego Image Netherlands

message encryption process took only 0.0012 seconds, data embedding into the image took 0.5816 seconds, and saving the stego-image to disk took 4.4054 seconds. In the decryption stage, extracting the hidden message from the image took 0.8026 seconds before it was decrypted again using the RSA private key. These values indicate that the system can perform compression, encryption, and embedding quickly, thereby supporting near-real-time processing.

For comparison, the study by AbdelWahab et al. [20], titled "Hiding Data Using Efficient Combination of RSA Cryptography and Compression Steganography Techniques," integrates RSA encryption, Huffman coding, and DWT-based compression to improve storage efficiency and image quality. Their evaluation, which considered metrics such as SSIM, CR, CT, CS, SP%, BPP, MSE, and PSNR, reported an average SSIM of approximately 0.9451 and a PSNR above 40.31 dB, indicating moderate perceptual quality with relatively low compression time. In contrast, as presented in Table 5, the proposed system achieves a higher average PSNR of 48.32 dB and SSIM of 0.9987, demonstrating superior image fidelity and imperceptibility.

The improvement in quality metrics can be attributed to the optimized combination of Enhanced RSA, Huffman coding, and LSB steganography, which collectively reduce information redundancy while strengthening encryption security. In addition, the proposed system shows faster processing across all operational stages, including key generation (0.0816 s), encryption (0.0012 s), embedding (0.5816 s), saving (4.4054 s), and extraction (0.8026 s), compared with the 0.199 s average compression time reported by AbdelWahab et al [20]. This performance gain highlights the computational efficiency of the hybrid model, achieved through parallelized bit-level operations in the LSB embedding stage and the reduced data size produced by Huffman compression. Overall, the results confirm that the proposed approach delivers both improved visual quality and faster execution time while maintaining a high level of cryptographic robustness.

Although the methodologies used are different, with this system using optimized LSB and Abdel-Wahab et al. [20] using a DWT-LSB combination, the measurement results show that the speed performance of the developed system is competitive with previous studies, particularly in terms of encryption and insertion efficiency as well as the visual quality of the stego-image. Furthermore, a comparison between the input images in Figures 5 and 7 and the stego-images in Figures 6 and 8 shows no significant visual degradation in the steganographic results. This is consistent with the high PSNR and SSIM values achieved by this system. Thus, it can be concluded that the system successfully maintains visual integrity while providing shorter processing times for the encryption and embedding stages, making it suitable for practical scenarios that require fast processing.

These findings demonstrate that the proposed hybrid model successfully balances security, imperceptibility, and computational performance. Compared with DWT-LSB methods, the system not only produces higher image quality, with PSNR > 48 dB and SSIM ≈ 0.999, but also delivers faster encryption

and embedding times, making it suitable for real-time steganographic applications. Overall, the evaluation confirms that the integration of compression, encryption, and embedding within a unified framework enhances both the robustness and practicality of steganographic data protection.

The technical robustness of the proposed system is supported by a mathematical framework consisting of 19 equations that describe the processes of compression, encryption, and data insertion. In addition to image quality evaluation, the system was also tested using statistical steganalysis methods such as RS analysis and the chi-square test. The results show that the stego-image has statistical characteristics similar to those of the original image, so it does not produce detectable anomalies. These findings indicate that the proposed method has good resistance to common steganalysis techniques while maintaining image quality and system efficiency.

C. Limitations and Future Work

In terms of performance, the system exhibits certain constraints that become more apparent as data complexity increases. In particular, the processing time for embedding and extracting messages increases with image size, which may reduce system responsiveness in high-resolution or real-time use cases. In addition, when multiple steganography operations are executed sequentially, there is a noticeable increase in memory usage. This can create challenges for deployment in resource-constrained environments. Additionally, the current implementation supports only PNG and JPG image formats, which limits the flexibility of the system and excludes other widely used formats such as BMP, TIFF, or GIF that may be beneficial in future versions.

From a security standpoint, the system remains theoretically susceptible to advanced steganalysis techniques, particularly those that use machine learning to detect hidden patterns. While the combination of RSA encryption and Huffman encoding provides a robust security layer, dependence on secure RSA key exchange introduces a potential vulnerability if the key is poorly managed or intercepted during transmission. Furthermore, the LSB technique, although effective and widely used, is inherently limited by the bit capacity of the cover image, which restricts the volume of data that can be hidden without degrading image quality or raising suspicion.

To address these limitations, several directions for future enhancement are proposed. One promising approach is the use of adaptive LSB selection, which would allow the system to determine the most suitable pixel locations for embedding based on image content. This would reduce the risk of detection by avoiding uniform modification patterns. The integration of deep learning algorithms could also be beneficial for optimizing both performance and security. Neural networks may improve the system's ability to adjust parameters dynamically, thereby increasing the stealth and efficiency of data embedding and retrieval processes.

In addition to algorithmic enhancements, improvements in data compression techniques could further increase the capacity of hidden messages. While Huffman encoding is effective, more advanced methods such as arithmetic coding or machine learning-based compression could provide better data density with minimal quality loss. Expanding the system's compatibility to support additional file types, such as BMP, TIFF, or even multimedia formats like video, would make the application more versatile and suitable for a broader range of real-world scenarios. Incorporating features that support real-time processing could also open opportunities for use in live communication systems.

In summary, while the current system effectively fulfills its objectives in terms of security, usability, and performance, there remains significant potential for further development. By addressing the identified limitations, future versions of the system could evolve into a more powerful, flexible, and intelligent tool for concealing digital information. These enhancements would not only improve technical robustness but also expand the scope of application in fields such as digital forensics, secure messaging, and data privacy. These results demonstrate that the proposed system successfully achieves its objectives of secure steganography while maintaining practical usability and performance.

IV. CONCLUSION

The proposed system successfully integrates LSB steganography with RSA encryption, thereby improving security while maintaining practical performance. The addition of Huffman encoding contributes to efficient data storage without compromising security. Future work could explore adaptive compression techniques and enhanced resistance to advanced steganalysis methods.

DECLARATION OF AI AND AI ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

During the preparation of this work the authors used ChatGPT in order to assist in refining the language and improving the structure of the manuscript. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

CREDIT AUTHORSHIP CONTRIBUTION STATEMENT

Muhammad Rifqy Abdul Gofur Al Fatah: Resources, Project administration, and Writing – review & editing. **Denar Regata Akbi**: Conceptualization, Methodology, Software, Data curation, Formal analysis, Investigation, Visualization, and Writing – original draft. **Bashor Fauzan Muthohirin**: Supervision, Validation, and Writing – review & editing.

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] A. I. Fajriadi, "Dugaan Pencatutan KTP di Pilgub Jakarta 2024, Pakar Siber Sebut Beberapa Aspek Penyebabnya," *Tempo.co*. Accessed: Sep. 24, 2024. Available: https://www.tempo.co/digital/dugaan-pencatutan-ktp-di-pilgub-jakarta-2024-pakar-siber-sebut-beberapa-aspek-penyebabnya-23776

[2] D. Luthfiani, "PDNS Diretas, 47 Layanan Kemendikbudristek Terganggu," *Tempo.co*. Accessed: Sep. 24, 2024. Available: https://www.tempo.co/hukum/pdns-diretas-47-layanan-kemendikbudristek-terganggu-46152

[3] A. Akbar, "Pusat Data Nasional Sementara Lumpuh Akibat Ransomware: Mengapa Instansi Pemerintah Masih Rentan Terhadap Serangan Siber?," *BBC News Indonesia*, 2024.

[4] R. Dutta, H. D. Dixit, R. Van Riel, G. Vunnam, and S. Sankar, "Hardware Sentinel: Protecting Software Applications from Hardware Silent Data Corruptions," in *Proc. Int. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, vol. 2, pp. 482–497, 2025.

[5] O. F. A. Wahab, A. A. M. Khalaf, A. I. Hussein, and H. F. A. Hamed, "Hiding Data Using Efficient Combination of RSA Cryptography and Compression Steganography Techniques," *IEEE Access*, vol. 9, pp. 31805–31815, 2021.

[6] S. Bhargava and M. Mukhija, "Hide Image and Text Using LSB, DWT and RSA Based on Image Steganography," *ICTACT J. Image Video Process.*, vol. 9, no. 3, pp. 1940–1946, 2019.

[7] C. A. Sari and W. S. Sari, "Kombinasi Least Significant Bit (LSB-1) dan Rivest Shamir Adleman (RSA) dalam Kriptografi Citra Warna," *J. Masyarakat Informatika*, vol. 13, no. 1, pp. 45–58, 2022.

[8] Y. Sanjalawe, S. Al-E'mari, S. Fraihat, M. Abualhaj, and E. Alzubi, "A Deep Learning-Driven Multi-Layered Steganographic Approach for Enhanced Data Security," *Sci. Rep.*, vol. 15, no. 1, pp. 1–30, 2025.

[9] W. A. Awadh, A. S. Ahmed, and N. F. Hameed, "Hybrid information security system via combination of compression, cryptography, and image steganography," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 6, pp. 6609–6618, 2022.

[10] K. Balhaf, N. A. Munassar, and A. Bagmeel, "Digital steganography and cryptography hybrid system combining LSB and RSA algorithms," *Zenodo*, 2025.

[11] M. M. Hummady and A. H. Morad, "Enhancement of system security by using LSB and RSA algorithms," *Al-Khwarizmi Engineering Journal*, vol. 18, no. 2, pp. 77–86, 2022.

[12] V. S. Kumari, G. R. Murthy, and P. K. Reddy, "Image steganography using RSA algorithm," *International Research Journal of Advanced Engineering and Management*, vol. 4, no. 2, pp. 45–50, 2025.

[13] S. Susanti, R. Umbara, and F. Himawan, "Hybrid cryptosystem using RC5 and SHA-3 with LSB steganography for image protection," *Innovatics Journal of Informatics and Technology (INNOVATICS)*, vol. 6, no. 1, pp. 55–63, 2024.

[14] A. H. Al-Faydi, "Improved LSB image steganography with high imperceptibility based on cover-stego matching," *IET Image Processing*, vol. 17, no. 9, pp. 2351–2363, 2023.

[15] S. Singh, A. Sharma, and R. Verma, "StegaCrypt: Integrating hybrid cryptography and image steganography," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 12, no. 4, pp. 1390–1397, 2024.

[16] K. Merlin, S. Pradiksha, and R. V. Babu, "A hybrid approach for data hiding using Twofish algorithm and compression steganography techniques," *International Research Online Journal of Innovations in Information Processing (IROIIP)*, vol. 5, no. 3, pp. 34–42, 2023.

[17] A. Hamza, M. A. Al-Husainy, and R. A. Ali, "Novel secure hybrid image steganography technique based on pattern matching," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 15, no. 7, pp. 2485–2503, 2021.

[18] V. Dass and L. R. Raju, "Robust data security through hybrid AES-RSA and LSB steganography techniques," *International Scientific Journal of Engineering and Management (ISJEM)*, vol. 6, no. 1, pp. 29–37, 2025.

[19] S. Rahman et al., "A Huffman code LSB based image steganography technique using Multi-Level Encryption (MLE)," *Sci. Rep.*, vol. 13, 2023.

[20] O. F. AbdelWahab, A. I. Hussein, H. F. A. Hamed, H. M. Kelash, A. A. M. Khalaf, "Efficient Combination of RSA Cryptography, Lossy, and Lossless Compression Steganography Techniques to Hide Data," *Procedia Comput. Sci.*, vol. 182, pp. 5–12, 2020.

[21] S. Rahman et al., "A novel and efficient digital image steganography," *Sci. Rep.*, 2024.

[22] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.

[23] C. A. B. Husada and Alamsyah, "Peningkatan Kualitas Stego-image Menggunakan Advanced Least Significant Bit," Indonesian Journal of Mathematics and Natural Sciences, vol. 45, no. 1, pp. 30–37, 2022.

[24] D. A. Huffman, "A method for the construction of minimum-redundancy codes," Proceedings of the I.R.E., vol. 40, no. 9, pp. 1098–1101, 1952.

[25] S. Rahman, J. Uddin, H. Hussain, A. Ahmed, A. A. Khan, M. H. Khan, and A. A. Memon, "A Huffman code LSB based image steganography technique using multi-level encryption and achromatic component of an image," Sci. Rep., vol. 13, art. no. 14183, 2023.

[26] S. Rahman et al., "A Huffman code LSB based image steganography technique using Multi-Level Encryption (MLE)," *Scientific Reports*, vol. 13, 2023.

[27] J. C. T. Arroyo, J. A. Espadero, and M. A. Ganas, "Steganography Method Using Effective Combination of RSA Cryptography and Data Compression," *Int. J. Adv. Trends Comput. Sci. Eng.*, 2020

[28] A. Alenizi et al., "A secure image steganography based on LSB technique and 2D chaotic maps," *J. Vis. Commun. Image Represent.*, 2024.

[29] A. Alanzy and A. Alomrani, "Image Steganography Using LSB and Hybrid Encryption Algorithms," *Applied Sciences*, vol. 13, no. 21, 11771, 2023.

[30] A. Alenizi, "A Review of Image Steganography Based on Multiple Techniques," *Journal of King Saud University – Computer and Information Sciences*, 2024.