Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

Hybrid Cryptography-Steganography Scheme Based on Camellia-256 and LSB for Enhanced Security and Imperceptibility of Secret Messages

Imam Prayogo Pujiono[*1], Eko Hari Rachmawanto[2], Christy Atika Sari[3], Said Fachri Ariza[4], Isnaeni Kholifatun[5]

[1,4,5]Informatics, Faculty of Islamic Economics and Business, Universitas Islam Negeri K.H. Abdurrahman Wahid Pekalongan, Indonesia
[2,3]Informatics, Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia

Email: [1]imam.prayogopujiono@uingusdur.ac.id

Received : Sep 23, 2025; Revised : Jan 6, 2026; Accepted : Jan 12, 2026; Published : Jun 15, 2026

Abstract

The development of digital communications has increased the risk of message interception and manipulation, necessitating robust and multi-layered security solutions. This research designs, implements, and evaluates a multi-layered security scheme that integrates cryptography and steganography. The proposed method first encrypts the secret message using the Camellia-256 algorithm in Electronic Codebook (ECB) mode with PKCS#7 padding. The resulting ciphertext is then embedded into the cover image using the Least Significant Bit (LSB) steganography technique. From a practical standpoint, this design provides defense-in-depth for covert communication: encryption preserves confidentiality even if the hidden payload is detected, while steganography reduces the likelihood that the encrypted content is flagged during transmission. This combination mitigates LSB's weakness against statistical steganalysis by encrypting the payload into ciphertext, thereby reducing structured bit patterns that may otherwise facilitate statistical detection. System performance is quantitatively evaluated using two primary metrics: the Avalanche Effect to measure cryptographic strength and the Peak Signal-to-Noise Ratio (PSNR) to measure the visual imperceptibility of the stego-image. The experimental results demonstrate excellent cryptographic strength, evidenced by an average Avalanche Rate of 54.37%, indicating that minimal changes to the input result in significant changes to the output. Furthermore, the scheme exhibits excellent visual imperceptibility with an average PSNR of 75 dB, making the stego-image visually indistinguishable from the original cover image. It is concluded that the proposed hybrid scheme offers a robust and validated solution for secure message communication, combining content confidentiality through cryptography and message obfuscation through steganography, thus providing dual protection against cybersecurity threats.

Keywords : *Camellia-256, Hybrid Cryptography, LSB, Secret Message, Steganography.*

1. INTRODUCTION

In the current digital era, the volume of data exchanged over public networks such as the internet has reached an unprecedented scale [1]. Critical sectors, including healthcare, banking, military communications, and information technology corporations, are heavily dependent on the rapid and efficient transmission of digital data [2]. However, this convenience and speed are accompanied by a significant increase in security risks [3]. Transmitted data is vulnerable to a variety of threats, including eavesdropping, unauthorized modification, and data theft by unauthorized parties [4]. As the sophistication of cyberattacks continues to grow, the need for reliable, multi-layered data protection mechanisms has become increasingly urgent to guarantee the confidentiality, integrity, and authentication of information [5].

To address these challenges, two disciplines within information security (cryptography and steganography) have emerged as foundational pillars [6]. They offer distinct yet complementary

Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

approaches that can be integrated to create a comprehensive security system [7]. Cryptography is the science and art of securing messages by transforming them into an unreadable format known as ciphertext [8]. Its primary objective is to protect the content of a message, ensuring that only parties possessing the legitimate decryption key can access the original information [9]. Despite its effectiveness in preserving content confidentiality, cryptography has an inherent weakness: the very existence of an encrypted message can attract attention and suspicion [10]. Ciphertext, which appears as random data, may become a target for cryptanalysts to analyze or even destroy, even if its contents remain inaccessible [11].

On the other hand, steganography is the science and art of concealing a message within another medium (referred to as a cover medium), such as an image, audio, or video file, in such a way that the message's existence is not detected [12]. Unlike cryptography, which protects content, steganography protects the existence of the communication itself [7]. Its goal is to facilitate covert communication [13]. However, the principal weakness of steganography is that if the concealment method is detected, the secret message can be easily extracted and read by a third party or eavesdropper [14].

Recognizing the respective limitations of each technique, researchers have developed hybrid approaches that combine cryptography and steganography [15]. This integration creates a synergistic, multi-layered security system where the weakness of one technique is effectively mitigated by the strength of the other [16]. The conspicuous nature of cryptography, which signals the presence of valuable data, is camouflaged by the covert nature of steganography [17]. Conversely, steganography's vulnerability to detection is addressed by cryptography [18], even if the presence of a hidden message is detected, an attacker is still confronted with encrypted data that is computationally difficult to decipher [19]. Thus, a hybrid approach does not merely add a layer of security but establishes a qualitatively stronger security paradigm in which the layers mutually reinforce one another [20].

Recent studies show an increasing shift toward hybrid security frameworks that combine encryption with steganography to achieve both confidentiality and covert communication. Encryption-assisted embedding schemes continue to be explored to reduce exposure when steganalysis is successful [7], while recent systematic evidence confirms that classical spatial-domain LSB can be statistically detectable at higher payloads, motivating layered or more adaptive designs [14]. Researchers have also proposed variants such as cryptography with K-LSB in cloud settings [15] and two-phase/distributed embedding strategies to strengthen secrecy under realistic adversaries [16]. Notably, Malik et al. [21] introduced a hybrid framework using DCT-domain embedding combined with GAN-based generation to enhance secure data communication in big-data contexts; however, this line of work does not evaluate the baseline combination of Camellia-256 encryption with spatial-domain LSB embedding. Therefore, the present study positions Camellia-256 + LSB as an auditable and reproducible baseline, providing empirical benchmark metrics (Avalanche Effect and PSNR) that can be compared against more complex recent approaches, including transform-domain and learning-based frameworks [21].

In designing such a hybrid system, the selection of algorithms for each layer is of paramount importance. For the cryptographic layer, a robust symmetric block cipher is required. While the Advanced Encryption Standard (AES) is a global standard widely adopted by governments and industries [22], this research explores the Camellia algorithm as an alternative with equivalent capabilities [23][24][25]. The Camellia algorithm is a 128-bit block cipher jointly developed by Mitsubishi Electric and Nippon Telegraph and Telephone (NTT), supporting key sizes of 128, 192, and 256 bits [26][27]. It offers a level of security and performance comparable to AES and is efficient for implementation in both hardware and software [28][29]. One of Camellia's strategic advantages is its independence from standards mandated by a specific government entity, such as AES by the U.S. National Institute of Standards and Technology (NIST). This makes Camellia an attractive choice for applications requiring algorithmic diversification or seeking to avoid dependence on a single

standardization body, a consideration that carries geopolitical and philosophical weight in the global cybersecurity landscape where trust in decentralized standards is highly valued. Technically, Camellia employs a Feistel Network structure with 24 rounds for a 256-bit key, ensuring a high degree of diffusion and confusion [30].

For the steganographic layer, the Least Significant Bit (LSB) technique was selected. LSB is one of the most fundamental spatial domain steganography methods, wherein the bits of a secret message are embedded into the least significant bits of the pixel values of a cover image [31]. The primary advantages of LSB are its simplicity of implementation and high payload capacity. However, its main drawback is its vulnerability to statistical steganalysis attacks, such as Chi-Square and Regular-Singular (RS) analysis, which can detect the statistical anomalies introduced by LSB embedding [32]. The selection of LSB in this research is strategic, its known vulnerabilities actually strengthen the justification for applying the cryptographic layer. Encryption transforms the message into ciphertext that statistically resembles random noise, which inherently enhances resistance to certain forms of statistical analysis and provides a final line of defense if the steganography method is detected [21].

Based on the preceding background and literature review, the primary problem addressed in this research is the need for an empirically validated, multi-layered data security mechanism capable of providing both message confidentiality and concealment of the message's existence with measurable performance. While many studies have proposed hybrid schemes, they often rely on standard cryptographic libraries that function as "black boxes," which complicates in-depth analysis and the development of core algorithms. Furthermore, quantitative performance data for the specific combination of the Camellia-256 algorithm and the LSB technique remains exceedingly limited.

Therefore, this research has several primary objectives: (1) To design and implement a hybrid security system using a manual implementation of the Camellia-256 algorithm (ECB mode, PKCS#7 padding) and LSB in the Java programming language, without reliance on standard cryptographic libraries such as javax.crypto. (2) To empirically evaluate the cryptographic security level of the implemented system using the Avalanche Effect metric. (3) To measure the visual quality and imperceptibility of the resulting stego-image using the Peak Signal-to-Noise Ratio (PSNR) metric.

The main contributions of this research are threefold. First, it presents empirical performance data for the specific combination of the Camellia-256 algorithm and LSB, for which data was previously scarce. Second, through manual implementation, this research provides code that is more transparent, auditable, and extensible for optimization, as well as more flexible for integration with other system components. Third, the quantitative results of this study can serve as a benchmark for researchers and practitioners in the field of information security to compare and develop other hybrid security schemes in the future.

2. METHOD

2.1. Research Design

This research was designed as a quantitative experimental study. Its objective is to measure and analyze the performance of the developed hybrid security system within a controlled environment. The research process encompasses the design, manual implementation, testing, and evaluation of performance metrics.

The overall workflow of the proposed system is illustrated in Figure 1. The process begins with three primary inputs: a plaintext message, a secret key, and a cover image. The encryption process transforms the plaintext into ciphertext using Camellia-256. This ciphertext is then embedded into the cover image using the LSB technique to produce a stego-image. The reverse process, involving extraction and decryption, is performed to recover the original message. Performance evaluation is conducted at two critical points: analysis of the Avalanche Effect during the encryption stage to measure

Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

cryptographic strength, and analysis of the PSNR between the cover image and the stego-image to measure visual quality.

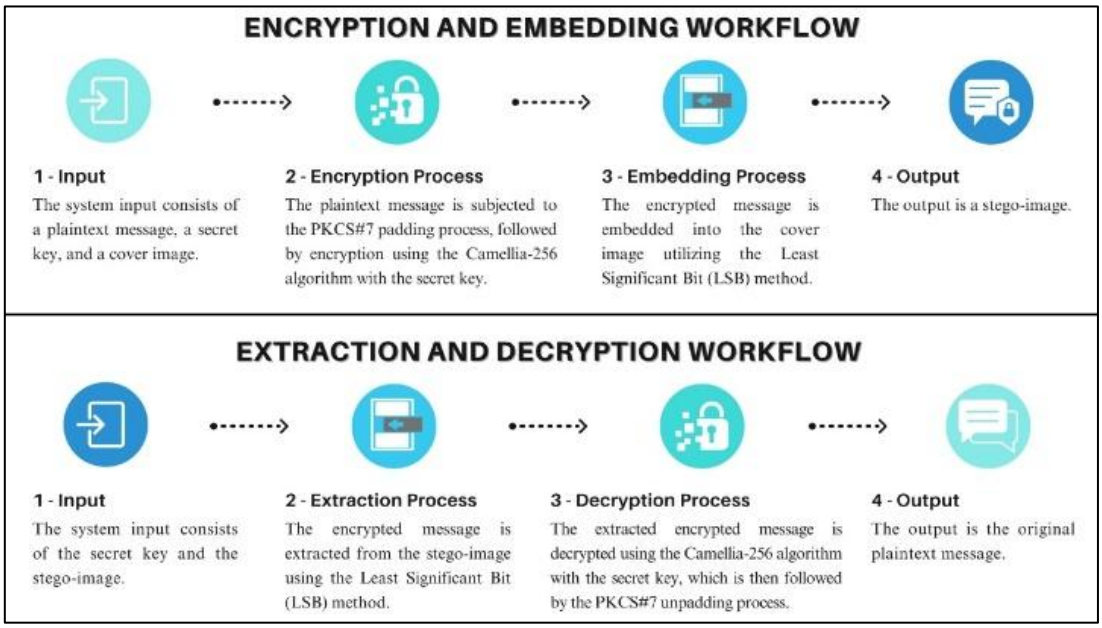


Figure 1. Hybrid Cryptography-Steganography System Workflow

As depicted in Figure 1, the system comprises two primary flows: the encryption and embedding flow, and the extraction and decryption flow. A detailed explanation of each flow is provided below.

Encryption and Embedding Flow:

1. Input: Plaintext message, Secret Key (256-bit), Cover Image.
2. Encryption Process (Camellia-256): The plaintext message undergoes PKCS#7 padding to ensure its length is a multiple of 128 bits (16 bytes). The padded message is then encrypted block by block using the Camellia-256 algorithm with the secret key.
3. Embedding Process (LSB): The ciphertext (in binary form) is embedded into the least significant bits of the pixels in the cover image.
4. Output: Stego-Image.

Extraction and Decryption Flow:

1. Input: Stego-Image, Secret Key (256-bit).
2. Extraction Process (LSB): Bits from the LSBs of the stego-image pixels are extracted sequentially to reconstruct the binary data.
3. Decryption Process (Camellia-256): The ciphertext is decrypted block by block using the same Camellia-256 algorithm and secret key. The decrypted data then undergoes PKCS#7 unpadding to remove the appended bytes.
4. Output: Plaintext message.

A fundamental aspect of this research is the manual implementation of the cryptographic and steganographic algorithms using the Java programming language, without reliance on built-in cryptographic libraries like javax.crypto. This approach facilitates a deep analysis of each component and the validation of its functionality from the ground up. The resulting code is more transparent and auditable, easier to extend or modify for optimization purposes, and more flexible for integration with other system components.

Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

2.2. Implementation of the Camellia-256 Algorithm

The Camellia algorithm is a symmetric block cipher with a fixed block size of 128 bits and support for key lengths of 128, 192, or 256 bits. In the Camellia-256 variant, which uses a 256-bit key, the encryption process employs 24 rounds based on a Feistel structure. This structure includes special FL/FL^{-1} layers inserted after the 6th, 12th, and 18th rounds, as well as 128-bit whitening operations (XOR) performed before the first round and after the final round [26][27]. The overall flow for the 192/256-bit variants is shown in Figure 2.

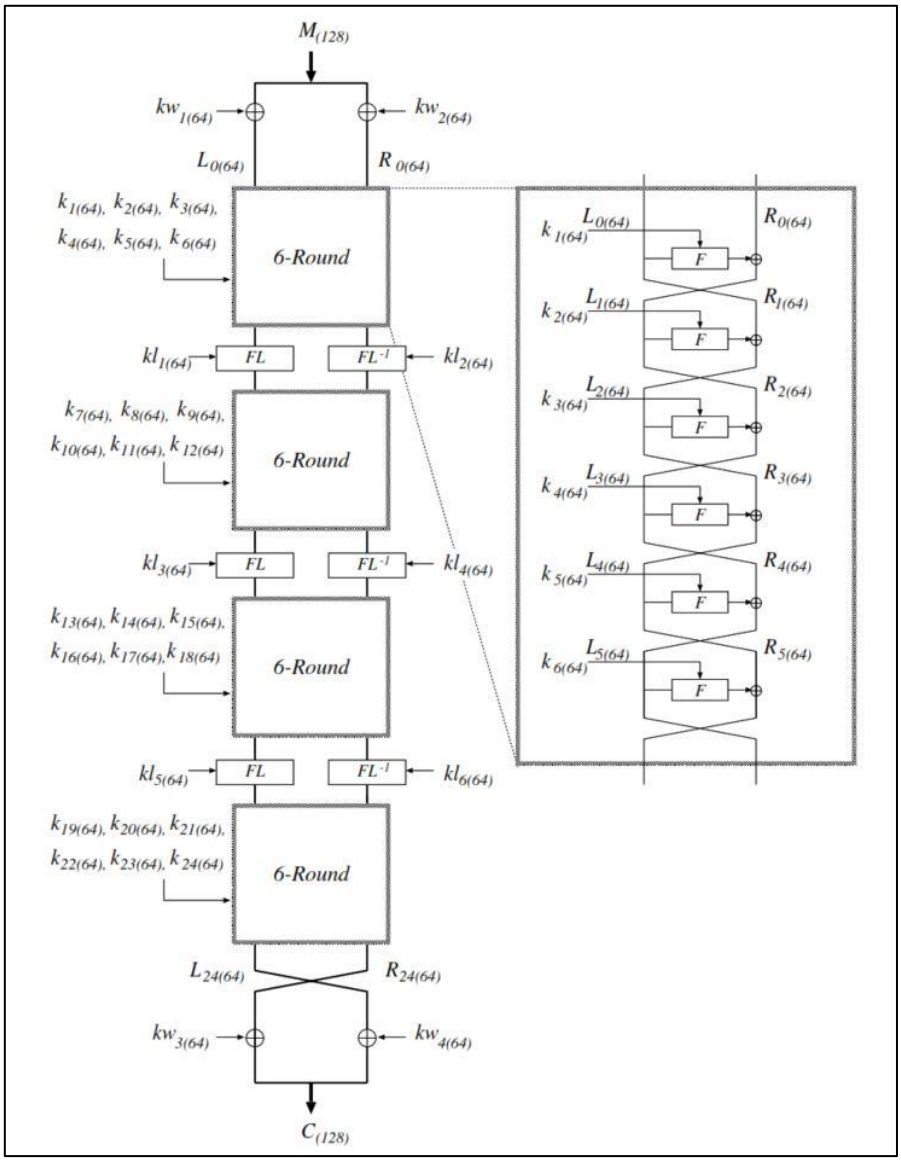


Figure 2. Camellia 192/256-bit Encryption Process [26][27]

The encryption process for the Camellia-256 algorithm can be summarized as follows:
Input: A 128-bit plaintext block (M) and a 256-bit secret key (K).
Output: A 128-bit ciphertext block (C).

1. Key Schedule. From the 256-bit key, form four 128-bit values: K_L (left 128 bits) and K_R (right 128 bits). Compute two intermediate values K_A and K_B through 2×2 rounds based on the F-function with predefined constants $\Sigma_1 ... \Sigma_6$, as illustrated in Figure 3.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

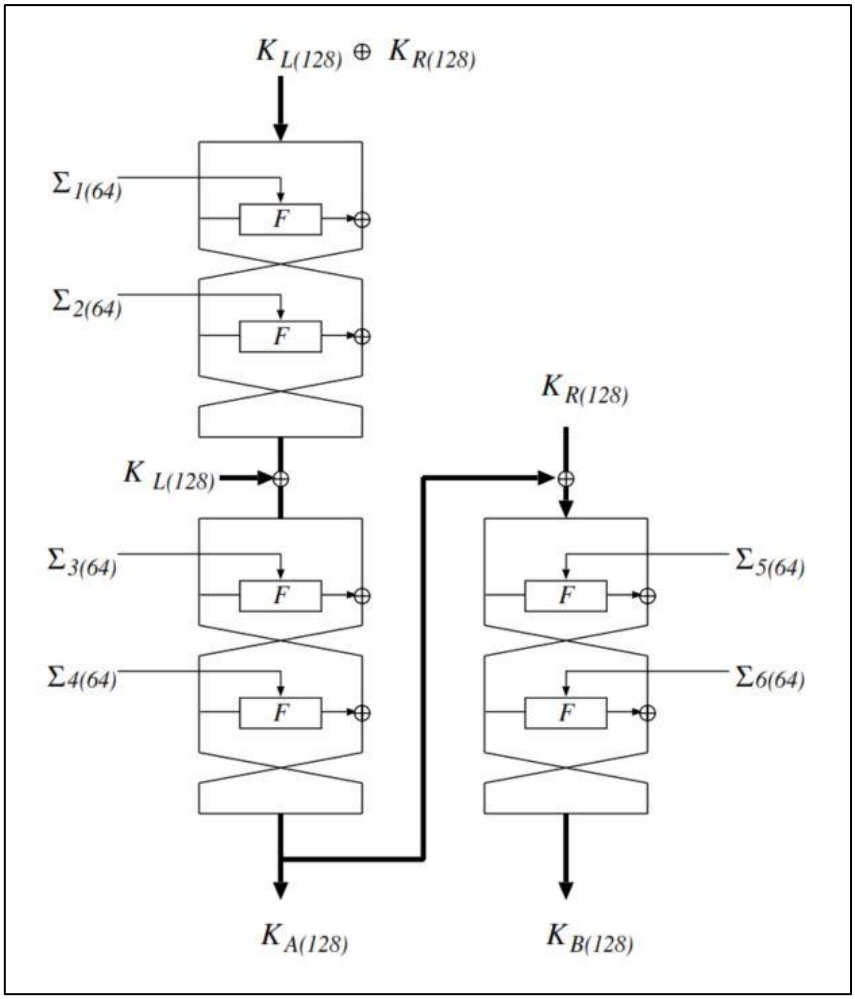


Figure 3. Camellia Algorithm Key Schedule Process [26][27]

The constants (Σ) used in the key schedule are:
Σ1 = A09E667F3BCC908B
Σ2 = B67AE8584CAA73B2
Σ3 = C6EF372FE94F82BE
Σ4 = 54FF53A5F1D36F1C
Σ5 = 10E527FADE682D1D
Σ6 = B05688C2B3E6C1FD

From fixed-bit rotations of K_L, K_R, K_A, K_B (by 0, 15, 30, 45, 60, 77, 94, 111 bits), 64-bit subkeys are extracted for pre-/post-whitening (kw_1 to kw_4), the 24 rounds (k_1 to k_{24}), and the FL/FL^{-1} layers (kl_1 to kl_6).

2. Pre-whitening & left/right side formation.

$$M \oplus (kw1 \parallel kw2) = L0 \parallel R0 \ (where\ L0\ and\ R0\ are\ each\ 64\ bits) \quad (1)$$

3. 24 Feistel Rounds. For each round (r) = 1…24, except for r = 6,12,18:
   - $L_r = R_{r-1} \oplus F(L_{r-1}, k_r)$
   - $R_r = L_{r-1}$

Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

At rounds r = 6, 12, 18, there are layers:

- Compute temporary $L'_r = R_{r-1} \oplus F(L_{r-1},k_r)$, $R'_r = L_{r-1}$
- Apply FL to L'_r with $kl_{2r/6-1}$ and FL^{-1} to $R'r$ with $kl_{2r/6}$ to obtain L_r,R_r.

4. Post-whitening & Ciphertext Finalization.

$$C = (R24 \parallel L24) \oplus (kw3 \parallel kw4) \quad (2)$$

The decryption process for Camellia-256 is identical in structure to the encryption process, with the only difference being that the round subkeys are applied in reverse order [26][27]:

2.2.1. Electronic Codebook (ECB) Mode of Operation

In this research, the ECB mode was implemented due to its simplicity and its ability to isolate the performance evaluation of the core block cipher. In ECB mode, each 128-bit plaintext block is encrypted independently using the same key. While this mode is not recommended for practical applications because patterns in the plaintext can manifest in the ciphertext, its use here allows for a pure measurement of the Avalanche Effect of the Camellia algorithm itself, without the influence of chaining mechanisms found in modes like CBC [33].

2.2.2. PKCS#7 Padding

Since block ciphers operate on blocks of a fixed size (16 bytes), any message whose length is not a multiple of the block size must be padded. This implementation uses the PKCS#7 padding scheme. If the final block of a message has a length of L bytes, N=16-L bytes are appended to it. Each of these appended bytes has the value N. If the message length is already a multiple of 16, a full block of padding (16 bytes, each with the value 16) is added [34].

2.2.3. Implementation Cross-Validation Against a Standard Library (Bouncy Castle)

To strengthen confidence in the functional correctness of the from-scratch Camellia-256 implementation, we performed cross-validation against the widely used Bouncy Castle cryptographic provider. Specifically, we (i) reproduced official Camellia-256 test vectors from the Camellia specification [26][27] and (ii) executed additional randomized tests by encrypting the same plaintext blocks under the same 256-bit keys using both implementations. Across all tested cases, the ciphertext outputs matched bit-for-bit, and decryption recovered the original plaintext consistently. This cross-check complements the Avalanche Effect analysis by validating correctness at the block-cipher level using an independent reference implementation.

2.3. Implementation of LSB Steganography

The LSB steganography technique was implemented to embed the ciphertext into the cover image. The two primary processes involved are as follows [31][32]:

1. Embedding Process: The program reads the cover image pixel by pixel. The ciphertext generated by Camellia-256 is converted into a bitstream. Each bit from this stream is then sequentially embedded into the least significant bit (LSB) of each color component (Red, Green, or Blue) of the pixels. For instance, to embed one byte (8 bits) of ciphertext, the LSBs of 8 pixel color components are required. The embedding order proceeds from the first pixel of the first row to the last pixel of the last row.
2. Extraction Process: This process is the inverse of embedding. The program reads the stego-image pixel by pixel and extracts the bit from the LSB of each color component. These bits are then concatenated to reconstruct the original ciphertext.

Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

2.4. Cryptographic Strength Analysis: The Avalanche Effect

The Avalanche Effect is a metric used to measure the degree of diffusion in a cryptographic algorithm. It is characterized by the property that a change of a single bit in the input (either plaintext or key) results in a change of approximately 50% of the bits in the output (ciphertext). This property is quantitatively measured using the Avalanche Rate, which is the percentage of output bits that change when one input bit is flipped. An Avalanche Rate approaching or exceeding 50% indicates that the algorithm possesses strong diffusion and is resistant to differential cryptanalysis [35].

The procedure for calculating the Avalanche Rate is as follows [35]:

1. A plaintext block P is encrypted using a key K to produce a ciphertext C.
2. A new plaintext block, P′, is created by flipping a single bit in P.
3. P′ is encrypted with the same key K to produce a new ciphertext C′.
4. The Hamming distance between C and C′ is calculated, which is the number of bit positions at which they differ.
5. The Avalanche Rate is calculated using the formula:

$$Avalanche\ Rate(\%) = \frac{Hamming\ Distance(\mathrm{C},\mathrm{C}\prime)}{Block\ Size(\mathrm{bits})}\ x\ 100\% \quad (3)$$

To ensure transparency and facilitate independent verification and replication of the cryptographic evaluation, the application code used for calculating the Avalanche Rate is publicly accessible at: http://bit.ly/3JcYwBX.

2.5. Visual Quality Analysis: Peak Signal-to-Noise Ratio (PSNR)

PSNR is a metric used to measure the level of distortion between an original image (the cover) and its modified version (the stego-image). A high PSNR value, measured in decibels (dB), indicates that the distortion is minimal and the stego-image has excellent, imperceptible visual quality [36]. The calculation of PSNR is based on the Mean Squared Error (MSE), which is computed first [36]:

1. Calculate the MSE between the cover image *I* and the stego-image *K*, both of size *M*x*N*:

$$MSE = \frac{1}{MxN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[I(i,j) - K(i,j)]^2 \quad (4)$$

2. Next, calculate the PSNR using the formula:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \quad (5)$$

where MAX_I is the maximum possible pixel value. For an image with 8 bits per color channel, MAX_I is 255. A PSNR value above 40 dB is generally considered to indicate that the difference between two images is undetectable by the human visual system [36][37].

In line with the commitment to methodological openness, the code for calculating MSE and PSNR is also provided and can be accessed at: http://bit.ly/45OAW7g.

3. RESULT

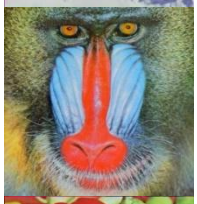
3.1. Presentation of Experimental Results

The experiments were conducted using a set of standard cover images from the USC-SIPI dataset, each with dimensions of 512×512 pixels, and secret messages with sizes ranging from 167 to 192 bytes. All procedures followed the design outlined in the Research Methodology section: Camellia-256 (ECB mode, PKCS#7 padding) was used for encryption, and LSB was used for embedding the resulting ciphertext into the images.

Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

In general, the system demonstrated two primary findings: (1) excellent cryptographic diffusion strength, with a consistent Avalanche Rate averaging 54.37%, and (2) extremely high visual imperceptibility, with an average PSNR of 75 dB. The test results on several standard USC-SIPI cover images are presented in Table 1, while a summary of the Avalanche Effect test for the scenario of flipping one bit in the plaintext while using the same key is provided in Table 2. The presentation of this data is crucial for demonstrating that the reported average results are not derived from a single idealized test case but are a reliable representation of the system's performance across various inputs, thereby enhancing the credibility and validity of the research findings.

Table 1. Performance Test Results on Various Standard Cover Images

Cover Image (512×512)	Embedded Message	MSE	PSNR (dB)
	Jaga perbatasan dan amankan jalur darat. Gunakan kata kunci ‘langit cerah’. Kirim sinyal tiga kali bila ada gerak asing; tunggu balasan ‘senja’. Hindari penandaan lokasi.	0.002	75.134
	Tutup pelabuhan sampai pemberitahuan. Alihkan kargo ke titik ‘ombak tenang’. Catat nomor lambung yang mendekat. Jika terdengar kode ‘malam’, matikan lampu pandu dan buka rute evakuasi.	0.002	74.847
	Beri veto penuh pada arahan yang melanggar protokol. Semua keputusan ditulis pada buku ‘daun hijau’. Buka komunikasi setelah kode ‘fajar’ diumumkan. Lakukan verifikasi ganda.	0.002	75.474
	Kode biru aktif hingga status normal. Periksa perangkat, simpan log, dan amankan kunci. Bila indikator merah muncul, jalankan prosedur 'angin sepoi'. Laporkan ringkas.	0.002	75.695
	Tutup akses ke sistem dan terowongan data. Kunci manual memakai frasa ‘batu karang’. Awasi percobaan masuk pukul 23.00 sampai 03.00. Jika pola berulang, tambah kunci pada menit ganjil.	0.002	75.156

Note: All messages were encrypted using the same key “PresidenRI”, and then embedded into 512×512 pixel images (262,144 pixels). This resulted in a very low embedding rate, which contributes to the high PSNR values. For a 512×512 RGB cover image, 1-LSB embedding across the three color channels provides a maximum capacity of 512×512×3 = 786,432 bits (≈98,304 bytes), i.e., 3 bits per pixel (bpp). In this study, the plaintext messages are relatively short (approximately 167-192 bytes), and after PKCS#7 padding the encrypted payload size becomes 176-208 bytes (1,408-1,664 bits). Therefore, the effective payload is only about 0.0054-0.0063 bpp, which corresponds to roughly 0.18%-0.21% of the available 3 bpp capacity. At such low payloads, only a very small fraction of pixel components is modified, which theoretically leads to a very small MSE and consequently a very high PSNR (consistent with the reported average PSNR of 75 dB). However, as the payload (bpp) increases toward higher-

Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

capacity regimes (e.g., larger messages or k-LSB variants), the expected MSE increases and PSNR decreases, and the stego-image becomes more susceptible to statistical steganalysis.

Table 2. Avalanche Rate Test Results for Camellia-256 by Flipping 1 Bit in the Plaintext

Initial Plaintext	New Plaintext	Initial Ciphertext	New Ciphertext	Hamming Distance	Avalanche Rate
Jaga perbatasan	Jaga perbatasam	8BB2424EB8EBEEB0D 790CEA6FC564311	24342A61D7C0B940B 9DC5EC8A7A808C0	71 bits	55.47%
Tutup pelabuhan	Tutup pelabuhbn	BABAAB8C2DFD8A78 7EF14ADC32C8ACB7	6B222A64A205C5094E AAD06EE9BB4008	70 bits	54.69%
Beri veto penuh	Beri ueto penuh	1750DCD82E39F85829 88A7708B4B6FF4	004D60FEABE2534E1 A71785A44557792	69 bits	53.91%
Kode biru aktif	Kode biru aktig	A9395807E337C54FA8 488A51AC019DA2	892695ACF03A2660B6 F371363003433B	69 bits	53.91%
Tutup akses	Tutuq akses	170E8BCF8176F67EC1 51650EA39D6D8D	5A8E16329E1DE19B1 33DC683C670184D	69 bits	53.91%

Note: The messages in the Initial Plaintext and New Plaintext columns were encrypted using the same key, “PresidenRI”.

3.2. Application Implementation and Interface

The application was implemented using the Java programming language, ensuring transparency and ease of code auditing without reliance on external libraries for the core cryptographic and steganographic functionalities. The application is organized into two main modules, corresponding to the system workflow shown in Figure 1: (1) the Encryption and Embedding Module, and (2) the Extraction and Decryption Module, both accessible from a main menu. The main menu provides simple navigation for selecting the desired module. Figure 4 shows the main menu interface of the application.

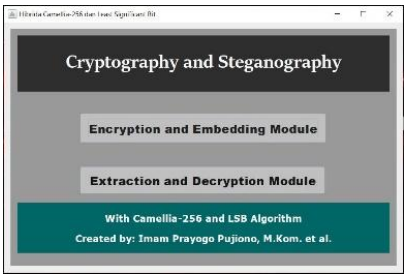


Figure 4. Application Main Menu Display

The Encryption and Embedding Module allows the user to input a plaintext message, a 256-bit key, and a cover image. It then performs Camellia-256 encryption (ECB with PKCS#7) followed by LSB embedding to generate the stego-image. Figure 5 shows the interface for this module.

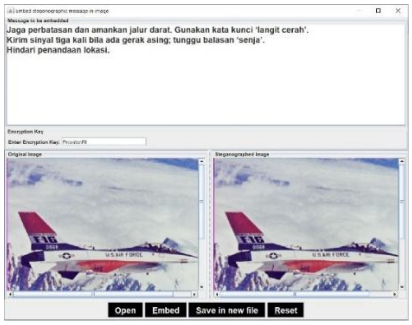


Figure 5. Encryption and Embedding Module Display

Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

The Extraction and Decryption Module accepts a stego-image and the corresponding key. It performs LSB extraction to retrieve the ciphertext, followed by Camellia-256 decryption and unpadding to recover the original plaintext. Figure 6 displays the interface for this module.

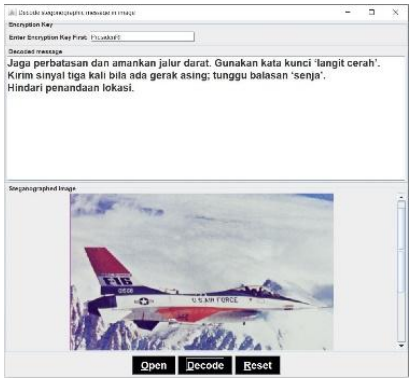


Figure 6. Extraction and Decryption Module Display

To ensure methodological transparency and facilitate independent verification and replication, the complete application code, including the manual implementation of the algorithms, can be downloaded from: http://bit.ly/45t8HK0. The code is designed in a modular fashion to facilitate replication and further development in subsequent research.

4. DISCUSSIONS

4.1. Analysis of Cryptographic Performance

The experimental results indicate that the manual implementation of Camellia-256 achieved an average Avalanche Rate of 54.37%. This value is a very strong indicator of cryptographic security. Theoretically, an ideal block cipher would flip approximately 50% of its ciphertext bits in response to a single-bit change in the input. This 50% mark represents perfect diffusion, where every input bit has an even and unpredictable influence on every output bit. The achievement of a 54.37% rate in this experiment significantly exceeds the 50% ideal threshold, demonstrating that the from-scratch implementation of Camellia-256 successfully created an extremely high level of diffusion and confusion. Any small change in the plaintext is effectively propagated throughout the entire ciphertext block, making it exceptionally difficult for a cryptanalyst to identify statistical correlations between the plaintext and the ciphertext. This performance directly contributes to the algorithm's resilience against differential cryptanalysis, an attack that exploits how differences in input can affect differences in output to deduce the secret key.

For context, other block ciphers considered strong, such as AES, exhibit an Avalanche Effect of approximately 50%-52.5% [38][39], while the classic Data Encryption Standard (DES) has an Avalanche Effect around 54.38%-54.68% [39][40][41]. The 54.37% result from this Camellia-256 implementation places its performance in a highly favorable category, comparable to these well-established and rigorously tested algorithms. This comparison serves to position the work within the broader academic discourse on block cipher validation and confirms that the manual implementation has successfully replicated the fundamental security properties of the Camellia-256 algorithm.

4.2. Analysis of Steganographic Quality

The second metric evaluated was the PSNR, which measures the visual imperceptibility of the stego-image. The experimental results show an exceptionally high average PSNR value of 75 dB. In the steganography literature, a PSNR value above 40 dB is generally considered sufficient to render any distortion invisible to the Human Visual System (HVS). The 75 dB value obtained in this study far

Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

surpasses this benchmark, which quantitatively confirms that the generated stego-image is visually identical to the original cover image. The extremely low distortion, as measured by an MSE of 0.002, further substantiates that the LSB embedding process introduced almost no perceptible noise.

To support this claim of imperceptibility, Figure 7 provides a visual comparison between the "Airplane 4.2.05" cover image and the resulting stego-image, along with their respective color histograms.

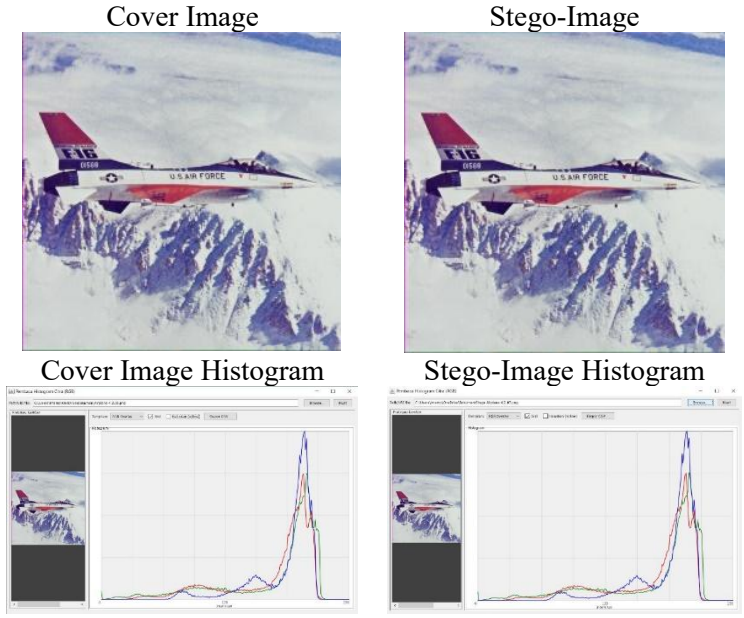


Figure 7. Comparison of Cover Image and Stego-Image for "Airplane 4.2.05"

As seen in Figure 7, there is no observable visual difference between the two images. Furthermore, the color histograms (Red, Green, and Blue) for the original and stego-images show nearly identical distributions. This provides qualitative corroboration for the quantitative PSNR results, confirming that embedding the ciphertext via LSB did not significantly alter the statistical properties of the image.

4.3. Implications and Synthesis of Findings

The synthesis of the two performance analyses demonstrates that the proposed hybrid system successfully achieved its established dual objectives. The Camellia-256 cryptographic layer provides robust and measurable content security, validated by an Avalanche Rate of 54.37%. Concurrently, the LSB steganography layer provides an extremely high degree of concealment for the communication's existence, validated by a PSNR of 75 dB. The successful validation of a manual, from-scratch implementation shows that a strong security system can be built and verified from its foundational components. This is a critical consideration for high-assurance systems where reliance on third-party, closed-source libraries may be undesirable. Overall, this research has successfully demonstrated the feasibility and effectiveness of the proposed hybrid scheme, providing a strong empirical foundation for future development.

4.4. Comparison with AES-256 (Theoretical Performance Considerations).

AES-256 and Camellia-256 are both 128-bit block ciphers designed for high security, but their internal structures differ, which has practical implications for speed and memory in software implementations. AES-256 is a substitution-permutation network with 14 rounds [23], whereas Camellia-256 is a 24-round Feistel network with additional FL/FL^{-1} layers [26][27]. In general-purpose

Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

software, throughput is strongly influenced by the number of rounds, the cost of each round operation, and (in practical deployments) the availability of platform-specific optimizations. AES-256 is widely supported and heavily optimized in mainstream libraries and can benefit from dedicated CPU instructions on many modern platforms; consequently, AES-256 is often used as a baseline for high-throughput encryption in practice. Camellia-256, while offering comparable security assurance and efficient software/hardware realizations [28][29], may show different throughput characteristics depending on the implementation strategy (e.g., table-based vs constant-time) and platform constraints. From a memory standpoint, both ciphers require only a few hundred bytes to store expanded round keys (AES-256 typically stores 15 round keys × 16 bytes = 240 bytes; Camellia-256 stores whitening, round, and FL/FL^{-1} subkeys totaling 34 × 8 bytes = 272 bytes). Thus, the expected memory difference is modest, but speed can vary substantially with implementation and hardware support.

5. CONCLUSION

This study has successfully designed, manually implemented, and empirically evaluated a multi-layered hybrid security scheme in response to the increasing information security threats in digital communications. By integrating the cryptographic strength of the Camellia-256 algorithm and the concealment capabilities of LSB steganography, the proposed system provides a dual protection solution. The quantitative evaluation results demonstrate excellent performance: the cryptographic strength is validated by achieving an average Avalanche Rate of 54.37%, indicating superior diffusion and resistance to cryptanalysis, while the superior visual quality is confirmed by an average Peak Signal-to-Noise Ratio (PSNR) value of 75 dB, ensuring the imperceptibility of the stego-image. This study explicitly answers the research question by demonstrating that the combination of Camellia-256 and LSB is an effective and validated method for securing data in digital images. These findings confirm that the research objectives have been achieved, with the main contribution being the provision of benchmark data from a transparent manual implementation. The practical implication of this system is its ability to provide highly classified communications secured by a combination of Camellia-256 and LSB.

Nevertheless, several limitations must be emphasized. First, the reported PSNR values reflect a low-payload regime (low bits-per-pixel), where only a small fraction of pixel components is modified. In large-scale deployments or higher-capacity embedding scenarios (higher bpp), classical spatial-domain LSB is expected to introduce stronger statistical artifacts, which can reduce PSNR and increase detectability under statistical steganalysis. Second, the current implementation employs ECB mode primarily as a baseline to isolate the behavior of the core block cipher during evaluation; however, ECB is not recommended for real-world secure communication due to its determinism and potential pattern leakage on repetitive plaintext.

REFERENCES

[1] H. Al Asyari, "Between Freedom And Protection: A Critical Review Of Indonesia'S Cyberspace Law," *Prophetic Law Review*, vol. 5, no. 1, pp. 79–103, 2023.

[2] Sachin *et al.*, "Advances in optical visual information security: a comprehensive review," in *Photonics*, MDPI, 2024, p. 99.

[3] A. Asmadi, H. Almutahar, S. Sukamto, Z. Zulkarnaen, E. I. Listiani, and A. Sikwan, "Digital Information Security Policy In The National Security Strategy," *International Journal Of Multidisciplinary Approach Research And Science*, vol. 1, no. 02, 2023.

[4] N. Alzahrani, "Security importance of edge-IoT ecosystem: An ECC-based authentication scheme," *PLoS One*, vol. 20, no. 6, p. e0322131, 2025.

[5] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, "A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions," *Electronics (Basel)*, vol. 12, no. 6, p. 1333, 2023.

Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

[6] G. ZHANG, "Cryptographic Techniques in Digital Media Security: Current Practices and Future Directions.," *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 8, 2024.
[7] A. M. Abed, H. Hermassi, and W. Barhoumi, "A New Encryption-Based Algorithm for Embedded Image Steganography," *International Journal of Sociotechnology and Knowledge Development (IJSKD)*, vol. 16, no. 1, pp. 1–28, 2024.
[8] E. M. Hamad, S. Alabed, A. Alsaraira, and O. A. Saraereh, "Implementing and developing multi-stage cryptography technique for low-cost long-range communication system," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 264–276, 2024.
[9] R. V Chethana, J. Vrindavanam, S. Roy, and P. C. Deshmukh, "A Review of Block Ciphers and Its Post-Quantum Considerations," *IEEE Access*, 2025.
[10] I. P. Pujiono, E. H. Rachmawanto, and D. A. Nugroho, "The Implementation of Improved Advanced Encryption Standard and Least Significant Bit for Securing Messages in Images," *Journal of Applied Intelligent System*, vol. 8, no. 1, pp. 69–80, Feb. 2023, doi: 10.33633/jais.v8i1.7324.
[11] C.-C. Chang, S. Xu, K. Gao, and C.-C. Chang, "Cryptanalysis of Dual-Stage Permutation Encryption Using Large-Kernel Convolutional Neural Network and Known Plaintext Attack.," *Cryptography (2410-387X)*, vol. 8, no. 3, 2024.
[12] P. K. Robinette, H. D. Wang, N. Shehadeh, D. Moyer, and T. T. Johnson, "SUDS: sanitizing universal and dependent steganography," *arXiv preprint arXiv:2309.13467*, 2023.
[13] U. Choudhary and P. Agarwal, "Image Steganography Combined with Cryptography for Covert Communication," in *Proceedings of the 2024 Sixteenth International Conference on Contemporary Computing*, 2024, pp. 207–212.
[14] R. Apau, M. Asante, F. Twum, J. Ben Hayfron-Acquah, and K. O. Peasah, "Image steganography techniques for resisting statistical steganalysis attacks: A systematic literature review," *PLoS One*, vol. 19, no. 9, p. e0308807, 2024.
[15] A. Goyal and B. Gupta, "A Hybrid Approach Using Cryptography and K-Least Significant Bit Steganography Algorithm in Cloud Computing," in *2025 First International Conference on Advances in Computer Science, Electrical, Electronics, and Communication Technologies (CE2CT)*, IEEE, 2025, pp. 196–202.
[16] K. Woźniak, M. R. Ogiela, and L. Ogiela, "A Two-Phase Embedding Approach for Secure Distributed Steganography," *Sensors*, vol. 25, no. 5, p. 1448, 2025.
[17] I. P. B. G. P. Raharja, J. A. Deka, R. M. Achmad, N. J. De La Croix, and T. Ahmad, "Steganography in Spatial Domain Images: Using Image Edge to Hide the Secret Data with a Quality Stego Image," *Ingenierie des Systemes d'Information*, vol. 29, no. 5, p. 1731, 2024.
[18] A. Itzhak Weinberg, "Singularity Cipher: A Topology-Driven Cryptographic Scheme Based on Visual Paradox and Klein Bottle Illusions," *arXiv e-prints*, p. arXiv-2507, 2025.
[19] M. M. Abd Zaid, A. A. T. Al-Khazaali, and A. A. Mohammed, "LSB Steganography using Dual Layer for Text Crypto-Stego," in *BIO Web of Conferences*, EDP Sciences, 2024, p. 00069.
[20] S. Rajabi-Ghaleh, B. Olyaeefar, R. Kheradmand, and S. Ahmadi-Kandjani, "Image security using steganography and cryptography with sweeping computational ghost imaging," *Front Phys*, vol. 12, p. 1336485, 2024.
[21] K. R. Malik *et al.*, "A hybrid steganography framework using DCT and GAN for secure data communication in the big data era," *Sci Rep*, vol. 15, no. 1, p. 19630, 2025.
[22] S. Dassanayaka, "Mixing Algorithm for Extending the Tiers of the Unapparent Information Send through the Audio Streams," *arXiv preprint arXiv:2502.12544*, 2025.
[23] K. D. Muthavhine and M. Sumbwanyambe, "An application of the khumbelo function on the camellia algorithm to prevent attacks in iot devices," *IEEE Access*, vol. 11, pp. 119959–119992, 2023.
[24] A. Jain and T. Singh, "Securing communication in IoT ecosystem using cryptographic algorithms," *Int J Eng Adv Technol*, vol. 9, no. 1, pp. 7258–7268, 2019.
[25] I. Holm and J. Dahl, "Anonymization of Sensitive Data through Cryptography," 2023.
[26] K. Aoki *et al.*, "Specification of Camellia-a 128-bit block cipher," *Specification Version*, vol. 2, 2000.

Vol. 7, No. 3, June 2026, Page. 2491-2505
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5323

[27] K. Aoki *et al.*, "Camellia: A 128-bit block cipher suitable for multiple platforms—design andanalysis," in *International workshop on selected areas in cryptography*, Springer, 2000, pp. 39–56.
[28] B. Rashidi, "Flexible and high-throughput structures of Camellia block cipher for security of the Internet of Things," *IET Comput Digit Tech*, vol. 15, no. 3, pp. 171–184, 2021.
[29] V. Thakor, M. A. Razzaque, and M. Khandaker, "Lightweight cryptography for IoT: a state-ofthe-art," 2020.
[30] C. W. Ci, S. Z. M. Naziri, R. C. Ismail, R. Hussin, M. N. M. Isa, and M. S. S. M. Basir, "Crypto-Core Design using Camellia Cipher," in *Journal of Physics: Conference Series*, IOP Publishing, 2021, p. 012019.
[31] M. Alanzy, R. Alomrani, B. Alqarni, and S. Almutairi, "Image steganography using LSB and hybrid encryption algorithms," *Applied Sciences*, vol. 13, no. 21, p. 11771, 2023.
[32] R. S. Hameed, S. S. Mokri, M. S. Taha, and M. M. Taher, "High capacity image steganography system based on multi-layer security and LSB exchanging method," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 8, 2022.
[33] M. Mansour, W. Elsobky, A. Hasan, and W. Anis, "Appraisal of multiple AES modes behavior using traditional and enhanced substitution boxes," *International Journal of Recent Technology and Engineering*, vol. 8, no. 5, pp. 530–539, 2020.
[34] A.-K. Wickert, L. Baumgärtner, M. Schlichtig, K. Narasimhan, and M. Mezini, "To fix or not to fix: a critical study of crypto-misuses in the wild," in *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, 2022, pp. 315–322.
[35] D. Upadhyay, N. Gaikwad, M. Zaman, and S. Sampalli, "Investigating the avalanche effect of various cryptographically secure hash functions and hash-based applications," *IEEE Access*, vol. 10, pp. 112472–112486, 2022.
[36] O. Keleş, M. A. Yılmaz, A. M. Tekalp, C. Korkmaz, and Z. Doğan, "On the Computation of PSNR for a Set of Images or Video," in *2021 Picture Coding Symposium (PCS)*, IEEE, 2021, pp. 1–5.
[37] A. Al Akbar, M. T. Sumadi, and F. Faldi, "IMPLEMENTATION OF LSB AND PLAYFAIR METHODS TO SECURE TEXT FILES INTO WAV AUDIO FILES," *Jurnal Teknik Informatika (Jutif)*, vol. 5, no. 6, pp. 1529–1537, 2024.
[38] S. Aljawarneh, M. B. Yassein, and W. A. Talafha, "A resource-efficient encryption algorithm for multimedia big data," *Multimed Tools Appl*, vol. 76, no. 21, pp. 22703–22724, 2017.
[39] K. D. Muthavhine and M. Sumbwanyambe, "An analysis and a comparative study of cryptographic algorithms used on the Internet of Things (IoT) based on avalanche effect," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, IEEE, 2018, pp. 114–119.
[40] S. Ramanujam and M. Karuppiah, "Designing an algorithm with high avalanche effect," *IJCSNS International Journal of Computer Science and Network Security*, vol. 11, no. 1, pp. 106–111, 2011.
[41] K. Mohamed, M. N. Mohammed Pauzi, F. H. Mohd Ali, and S. Ariffin, "Analyse on avalanche effect in cryptography algorithm," *European Proceedings of Multidisciplinary Sciences*, 2022.