
Article

Selective Color Image Encryption Based on MSB and Sensitive Bits

Issa Jacaman  and Mousa Farajallah * 

College of Information Technology and Computer Engineering (CITCE), Palestine Polytechnic University,
Hebron P.O. Box 198, Palestine

* Correspondence: mousa_math@ppu.edu

Received: 6 July 2025; **Revised:** 17 August 2025; **Accepted:** 22 August 2025; **Published:** 8 September 2025

Abstract: This paper introduces a selective image encryption framework for color imagery, emphasizing computational efficiency without compromising practical security. The core idea is to encrypt only the most informative and perceptually critical components of each pixel, while bypassing nonessential data to reduce processing cost; a full-encryption variant is implemented to enable apples-to-apples comparison. Evaluated on a representative set of natural color images, the selective scheme achieves encrypted outputs with an average Peak Signal-to-Noise Ratio (PSNR) of 8.7 dB, Mean Structural Similarity (MSSIM) of 0.07, and Information Entropy (IE) of 7.8 bits. These values are closely aligned with those obtained under full encryption, indicating low residual similarity to the plaintext and near-uniform randomness in cipher histograms. In qualitative terms, the visual content is thoroughly obfuscated, while the selective strategy decreases the amount of data entering the cryptographic core, yielding measurable speedups. The design integrates permutation and diffusion stages suitable for block-based processing and common cipher modes, and supports region-of-interest operation when desired. Together, the empirical evidence and design choices suggest a practical trade-off: comparable security indicators at a fraction of the computational effort. The approach is particularly attractive for resource-constrained settings, batch protection of large image corpora, and latency-sensitive pipelines. Future work will extend the methodology to grayscale imagery, broaden the testbed, and incorporate a dedicated pseudo-random number generator to decouple randomness from platform dependencies.

Keywords: Selective Image Encryption; Region of Interest (ROI); MSB; Fisher–Yates Shuffling; CBC; Information Entropy

1. Introduction

Efficient encryption is vital for secure data sharing on resource-constrained devices. Balancing energy and memory usage is crucial for device optimization. Traditional algorithms, with high computational intensity, prove ineffective for safeguarding data on limited-resource devices. Image pixel encryption faces challenges due to the similarity of values in adjacent pixels. The proposed algorithm overcomes these challenges, ensuring a robust cryptosystem by encrypting sensitive bits in a color image's selected region of interest, determined through an edge detection technique.

As big data continues to grow, the increasing digitization of information brings convenience but also raises concerns about personal privacy breaches and data theft. Consequently, securing digital images has become a vital research focus.

Digital images encompass extensive data, incorporating redundant information and exhibiting high pixel correlation [1]. Image data possesses unique and inherent characteristics that differentiate it from conventional textual

information. These characteristics encompass a closely interconnected relationship between neighboring pixels and a notable degree of data redundancy. In order to address these specific features, it is imperative to devise novel algorithms that deviate from conventional cipher algorithms intended exclusively for textual data, such as Data Encryption Standard (DES) and Advanced Encryption Standard (AES) [2]. A proposition has been put forth for the selective encryption of sensitive bits, employing the Fisher-Yates algorithm-based color image encryption method. This aims to establish a highly secure and efficient image encryption scheme.

This paper presents a novel encryption algorithm tailored for color images. This paper proposes a selective image encryption algorithm utilizing edge detection criteria. The algorithm selectively encrypts image blocks using edge detection criteria. These blocks contain vital data. It starts with an exclusive OR operation on the most significant bits of each pixel, followed by swapping the other three most significant bits among the pixel's color channels.

Following this, the pixels undergo a vertical scrambling process. Ultimately, all bytes are XORed based on a specific equation, enhancing the level of confusion in the encryption process. The encryption algorithm introduced in this proposal has embraced the Cipher Block Chaining (CBC) mode of operation.

In summary, the method's main focus is edge-aware selective color image encryption: it targets Laplacian-derived regions of interest and the most informative bitplanes (the MSB and the next three MSBs) to prioritize efficiency while preserving security under CBC.

Contributions

We articulate the following contributions, each supported by quantitative evidence and comparative analysis:

- Method: A selective color image encryption scheme targeting informative bit-planes inside edge-/saliency-derived ROI under CBC, combining MSB and byte-wise XOR with cross-channel bit swapping and intra-block Fisher-Yates shuffling (§4, §3.1).
- Experimental evidence: A comprehensive evaluation on ten color images with full-image and ROI-only metrics (PSNR/MSE, MSSIM, entropy, neighbor correlation) and differential analysis (NPCR/UACI), reported across pipeline stages $M \rightarrow MC \rightarrow MCP \rightarrow MCPB$ (§4.2–§4.3, §4.6–§4.8).
- Efficiency: Time/performance and complexity analyses showing 35–50% runtime reduction versus full encryption on 512×512 images, proportional to the ROI fraction ρ (§4.1, §4.9).
- Comparative evaluation: Head-to-head comparisons with full encryption and literature baselines (AES-CBC, Fisher-Yates+chaos, IEVCA), contextualized against recent studies and surveys (§2, §7.3) [1–9].
- Robustness and security: Parameter sensitivity (block size, thresholds), robustness to JPEG/noise, error propagation and integrity guidance, and resistance checks under known/chosen-plaintext models with standards-aligned PRNG/KDF/IV recommendations (§3, §4.10–§4.11) [10–13].

2. Selective Encryption

The importance of selective image encryption cannot be overstated, as it enables significant savings in computations, expenses, and time. Many attempts have been undertaken in this regard, as conventional encryption algorithms for entire images can be overly extensive. Massoudi et al. [14] asserted that image data statistics differ significantly from traditional text data due to their strong correlations and robust spatial/temporal redundancy. Thus, when comparing selective encryption to traditional encryption algorithms, selective encryption proves to be an effective strategy, ensuring dependable security measures and computational needs without any compromises [4].

Selective Encryption spans various domains, including spatial, frequency, and hybrid [15]. Selective encryption is applied in the frequency or spatial domain based on specified criteria. In the frequency domain, particular image data coefficients undergo encryption. In the spatial domain, selective encryption alters and disperses pixel or bit values [14]; in this paper selective encryption algorithm is implemented in the spatial domain.

Recent studies further motivate bit-plane-focused and ROI-aware designs. Chaos-assisted and bit-plane color image encryption has advanced low-complexity pipelines comparable to ours [1,4,5], and contemporary surveys emphasize rigorous evaluation with PSNR/MSSIM, entropy, correlation, and differential measures (NPCR/UACI) [6]. Domain-specific work and selective ROI designs likewise report strong obfuscation with near-ideal NPCR/UACI and

low PSNR, aligning with our goals [7–9].

Related Work and Comparative Studies

Recent comparative studies demonstrate diverse selective or hybrid strategies and serve as baselines for our evaluation. Chaos-assisted Fisher–Yates methods [1], cellular automata–based designs (IEVCA) [4], bit-plane and chaotic-system hybrids [5], and ROI-focused approaches for color/medical images [7–9] report low PSNR, low MSSIM, high entropy, and near-ideal NPCR/UACI under full encryption. Adaptive/chaos switching and content-sensitive schemes further support the selective processing rationale [2,3]. Surveys [5,6,9] recommend multi-metric reporting and emphasize sensitivity analyses. In this context, our approach contributes an edge-aware ROI policy with optional saliency augmentation, MSB/byte-wise XOR under CBC, and measured efficiency gains relative to full encryption while maintaining strong ROI obfuscation. We complement this with standards-aligned randomness and IV handling [10–13].

Positioning and Novelty

Unlike full-image encryption (e.g., AES/3DES) and selective schemes that operate solely in chaos/transform domains, our method targets the most informative bit-planes within edge-/saliency-derived ROI in the spatial domain, coupling MSB/byte-wise XOR, cross-channel bit swapping, and intra-block Fisher–Yates shuffling under CBC. This design achieves (i) ROI-centric concealment with reduced computational load proportional to ρ , (ii) near-ideal differential resistance within ROI (NPCR/UACI) comparable to full encryption, and (iii) practical integration guidance via standards-based key/IV derivation and error-resilience measures, which together differentiate the approach in both efficiency and deployment practicality.

3. Experimental Setup

We evaluate both selective (ROI-only) and full-encryption variants on a diverse set of 10 color test images (512×512 unless otherwise noted), including high-frequency textures (e.g., Baboon/Mandrill), low-texture/flat regions, faces and people scenes, saturated-color objects (e.g., Peppers), and natural/indoor scenes (e.g., Desk, School), alongside Lena and Barbara [6,7]. Images are processed in 8×8 and 16×16 blocks to study block-size effects. For each image, we compute the ROI by Laplacian edge counts with a threshold set to the per-image average (Equation (1)) unless otherwise stated; the resulting ROI fraction ρ typically falls in the 0.38–0.61 range across the dataset (median $\rho \approx 0.48$).

Hardware and software. Experiments run on Windows 10 (64-bit), 3.0 GHz CPU, 32 GB RAM. Wall-clock time is measured with Python’s `time.perf_counter` and averaged over 10 runs per configuration. We report mean \pm standard deviation when applicable and normalize time by (i) total pixels and (ii) pixels processed (ROI vs full) to provide throughput (MB/s) and cycles/byte. All code and experiment scripts used in this study are available at the study by Barker and Kelsey [10].

For comparative evaluation, we include: 1. Full AES-CBC over the entire image (byte-wise), implemented with standard block chaining and random per-image IVs. 2. A Fisher–Yates + chaos-based color image method representative (reported values from the literature when reimplementations differences are material) [1]. 3. IEVCA (2-D Von-Neumann cellular automata) results from where available (e.g., MSE/PSNR, correlation) [4]. Where public code is unavailable or not directly comparable, we cite reported metrics and focus on our normalized throughput and security indicators for practical comparison.

Reporting protocol. We evaluate four pipeline stages (M, MC, MCP, MCPB) under CBC. Metrics include MSE/PSNR, MSSIM, entropy, neighbor correlation (H/V/D), NPCR, and UACI, as defined in Section 5. To contextualize selective protection, we report metrics (a) over the full image and (b) restricted to ROI masks (“ROI-only”), which reflect obfuscation where encryption is applied. Robustness is assessed by adding Gaussian noise ($\sigma = 5, 10$) and JPEG compression ($q = 30, 50, 80$) to ciphertexts before metric computation; we report metric stability under these conditions.

Parameter sensitivity. We vary (i) the Laplacian threshold as a percentage of $Av g_{EDGE}$ (e.g., 80%, 100%, 120%) and (ii) block size ($8 \times 8, 16 \times 16$) and track ρ versus PSNR, MSSIM, entropy, and time. This reveals the efficiency–security trade-off as ρ changes. Unless otherwise specified, figures/tables summarizing these sweeps, per-image ρ ,

and ROI visualizations are provided in the **Supplementary Materials** package.

4. Proposed Algorithm

The algorithm presented focuses on the encryption of sensitive bits and pixels within the region of interest. This is achieved through pixel scrambling, exclusive-OR operations, and the swapping of red, green, and blue (RGB) channels. The most significant bit of a pixel is deemed sensitive, given that this 8th bit encompasses 50% of the pixel's value. Furthermore, the other three significant bits (7th, 6th, and 5th) can also be regarded as sensitive bits.

The algorithm under consideration partitions the image into n blocks and exclusively encrypts the region of interest. Initially, the Laplacian edge detector identifies the blocks that contain crucial data. Subsequently, the encryption algorithm is applied to these identified crucial blocks. Therefore, the algorithm proposed here divides the encryption of blocks into three phases. In the first phase, encryption involves applying an exclusive OR operation to the most significant bit of each byte within the three channels. Secondly, the algorithm swaps the other three most significant bits among the RGB channels. In the final phase, the algorithm scrambles the pixels row-wise and performs an exclusive-OR operation over all the pixels in the blocks. The Cipher Block Chaining (CBC) mode of operation is employed to ensure that the region of interest remains statistically indistinguishable.

The datasets used for experimentation include images of Lena and Barbara [6,7]; both images are in color and possess dimensions of 512×512 pixels. Additionally, the encryption and decryption algorithms employ a pseudo-random number generator (PRNG) [8].

As depicted in **Figure 1**, the proposed solution entails selectively encrypting an image by exclusively encrypting the region of interest within the blocks. The Laplacian edge detection method is employed to identify the blocks within the region of interest (ROI). The image is initially segmented into multiple blocks, and the critical blocks are identified based on their edge count, with blocks exceeding a specific threshold considered significant. Following this, an exclusive-OR operation is employed to encrypt the most significant bit (MSB) of each pixel's three channels within the selected blocks. After the encryption of the MSB bit within the blocks' pixels in the three channels, swap and scramble operations are carried out. The swapping involves interchanging the RGB channels among the pixels, and scrambling occurs among the pixels within a block column-wise, employing the Fisher-Yates algorithm. Ultimately, all pixels (bytes) within the region of interest undergo encryption through an exclusive-OR operation. Furthermore, to ensure the robustness of the proposed encryption scheme, Cipher Block Chaining (CBC) mode is applied to the clear-text blocks before encryption.

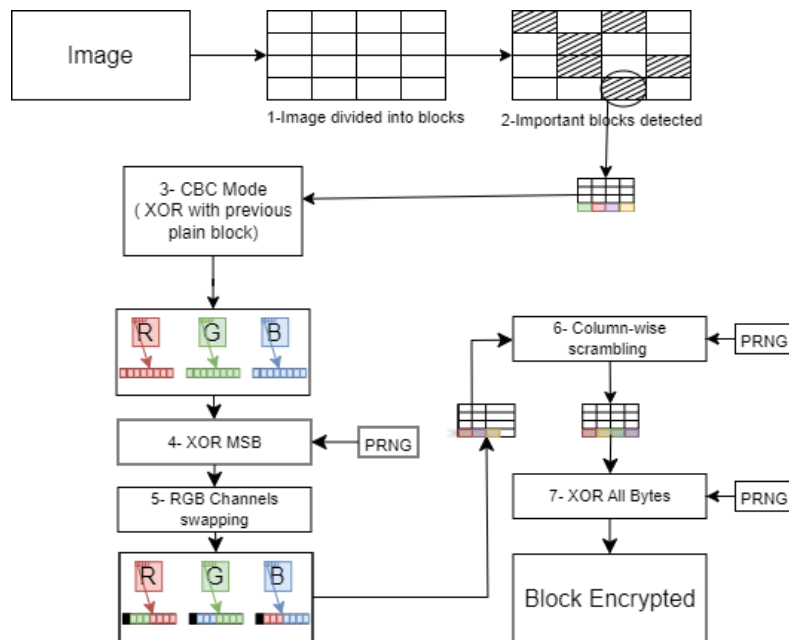


Figure 1. Proposed Encryption Algorithm.

The outlined encryption algorithm, depicted in **Figure 1** and presented in pseudo-code in **Algorithm 1**, can be succinctly summarized with the following points:

1. Divide the image into blocks
2. Identify the ROI using Laplacian edge detection
3. Execute exclusive-OR operation over the plain block with the previous cypher block/initial value (CBC mode of operation).
4. Execute the exclusive-OR operation on the most significant bit
5. Swap the three other most significant bits among the RGB channels
6. Scramble pixels within the block column-wise using the Fisher-Yates algorithm
7. Execute exclusive-OR on all bytes

In the subsequent subsections, a comprehensive review of each of the aforementioned points will be provided individually.

Algorithm 1 Proposed Encryption Algorithm

Require: *image*

Blocks \leftarrow *image_divider(image)*

enc_blocks \leftarrow *image_divider(image)*

counter \leftarrow 0 **for each** *block* \in *Blocks* **do**

counter ++

edge_count = *get_laplacian_edge_count(block)*

If *edge_count* \geq *threshold* **then**

enc_block[counter] = *cbc_mode(block)*

enc_block[counter] = *msb_xor(enc_block[counter])*

enc_block[counter] = *scramble_pixels(enc_block[counter])*

enc_block[counter] = *xor_bytes(enc_block[counter])*

end if

end for *row* < *Block.rows*

4.1. Threat Model and Key/IV Management

We consider ciphertext-only and non-adaptive settings and additionally assess resistance under KPA/CPA in §4.10. The secret material consists of a per-image key/seed used to derive: (i) a fresh per-image IV for CBC, (ii) per-stage bits/bytes for MSB and byte-wise XOR, and (iii) optional per-block IV material. To avoid keystream/IV reuse and strengthen unpredictability, we recommend a CSPRNG/DRBG and KDF per established standards:

- DRBG: NIST SP 800-90A (e.g., Hash_DRBG or CTR_DRBG) or ChaCha20-based generation for speed and robustness [10,11,16,17].
- KDF: HKDF (RFC 5869) or NIST SP 800-56C/800-133 to derive subkeys, IVs, and per-image salts from a master key and unique nonce/context [18–20].
- CBC/IV: Implement CBC per NIST SP 800-38A and ISO/IEC 10116; use a unique, unpredictable per-image IV; include IV and parameters as associated data for integrity if AEAD/HMAC is applied [12,13].

Operational policy. Keys and IVs are never reused across images; seeds/IVs are generated via the DRBG and derived via the KDF with distinct context labels. Per-image metadata minimally records the algorithm version, block size, ROI policy, and IV. These practices mitigate KPA/CPA risks by preventing structural reuse and ensuring high-entropy randomness sources.

Algorithm 2 functions as a summary for the decryption algorithm, acting as the inverse process of the proposed encryption algorithm outlined in **Algorithm 1**.

4.2. Divide the Image into Blocks

The plaintext image is divided into multiple blocks (*n*) to identify the noteworthy blocks of the image, referred to as Regions Of Interest (ROIs). As an example, Lena's 512 × 512 pixel image has been partitioned into 8 × 8 blocks, with each block comprising 64 × 64 pixels.

Algorithm 2 Proposed Decryption Algorithm

Require: *enc_image*
Blocks \leftarrow *image_divider(enc_image)*
dec_blocks \leftarrow *image_divider(enc_image)*
counter \leftarrow 0 **for each** *block* \in *Blocks* **do**
 counter ++
 edge_count = *get_laplacian_edge_count(block)*
 If *edge_count* \geq *threshold* **then**
 dec_block[counter] = *xor_bytes(enc_block[counter])*
 dec_block[counter] = *scramble_pixels(enc_block[counter])*
 dec_block[counter] = *msb_xor(enc_block[counter])*
 dec_block[counter] = *cbc_mode(block)*
 end if
end for *row* < *Block.rows*

Figure 2a displays Lena's test image, which is segmented into 64 blocks, as illustrated in Figure 2b.

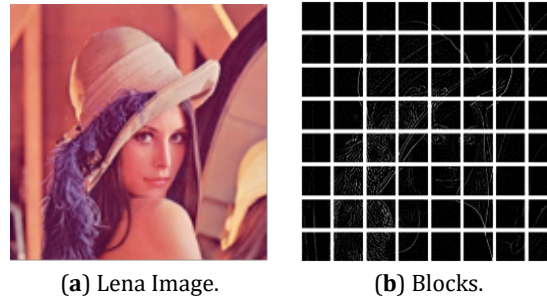


Figure 2. Lena Image Blocks.

4.3. Identify the Important Blocks Using Laplacian Edge Detection

The Laplacian edge detector filters the image, emphasizing all edges with high contrast properties, as depicted in Figure 3a. Furthermore, Figure 3b presents the filtered image divided into 8×8 blocks. The average number of edges within the blocks of an image is determined and calculated using the following equation:

$$Avg_{EDGE} = \frac{\sum_{i=1}^n CountEDGE(block_i)}{n} \quad (1)$$

The Avg_{EDGE} serves as a criterion for selecting the most crucial blocks (region of interest) in the image for encryption.

A block with an edge count surpassing Avg_{EDGE} is subjected to encryption. The greyed-out blocks in Figure 3c symbolize the region of interest (ROI).

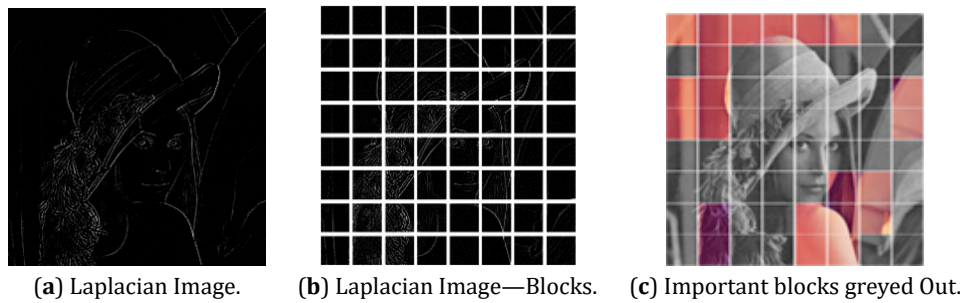


Figure 3. Lena Image.

4.4. Cipher Block Chaining Mode of Operation

Employing the Cipher Block Chaining (CBC) mode, each plain block within the ROI undergoes an XOR operation with the previously encrypted block. The initial block is XORed with an initial vector (IV), which is a randomly generated block. This initialization vector (IV) consists of a uniformly distributed array of random numbers, which is then XORed with each pixel in the first plaintext block. We follow the standard definitions and security guidance for CBC and IV handling per NIST SP 800-38A and ISO/IEC 10116 [12,13].

Figure 4 illustrates the CBC mode used within the encryption algorithm.

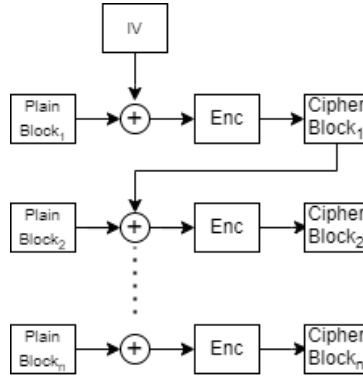


Figure 4. Block Images with CBC Mode of Operation.

4.5. Execute Exclusive-OR Operation on the Most Significant Bit

A colored image is composed of pixels constructed from three channels: red, green, and blue (RGB). Each channel in a colored image consists of 8 bits, as depicted in **Figure 5**.



Figure 5. A Pixel (RGB Channels).

After identifying the region of interest and completing the CBC process, the 8th bit of each pixel in the RGB channels within the selected block is subjected to XOR operations using the equation in (2):

$$\begin{aligned}
 MSB_{newR} &= MSB_{oldR} \oplus IV_R \oplus Random_Bit_R \\
 MSB_{newG} &= MSB_{oldG} \oplus IV_G \oplus Random_Bit_G \\
 MSB_{newB} &= MSB_{oldB} \oplus IV_B \oplus Random_Bit_B
 \end{aligned} \tag{2}$$

Here, MSB_{new} represents the resulting new bit value, Random_Bit is a randomly generated bit from a pseudo-random number generator, and IV_R , IV_G , and IV_B denote the initial vectors of the RGB channels, respectively. The initial values can have one of the following two values:

1. If there is no neighboring pixel to the left of the encrypted pixel (in the case of the first pixel being XORed), it must have a predetermined value.
2. When a pixel adjacent to the encrypted pixel on the left exists, the IV values for each channel are set to the 8th bit of the previously XORed pixel (MSB_{new}).

The exclusive-OR encryption of the most significant bit (MSB) for each RGB channel is illustrated in **Figure 6a**. The resulting image with the encrypted MSB channels is displayed in **Figure 6b**. To further strengthen the encryption, the process incorporates swapping and shuffling (Fisher-Yates), as elaborated in the subsequent sections.

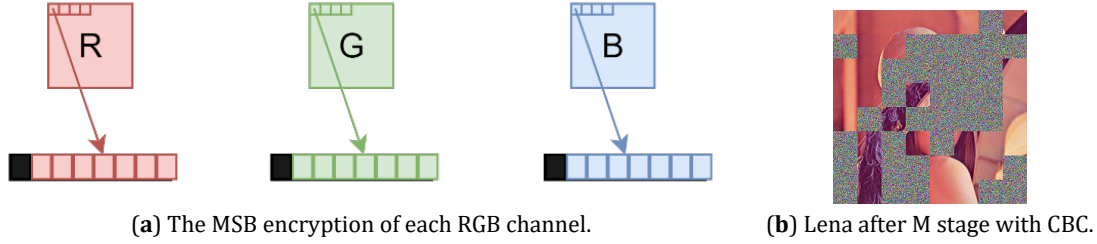


Figure 6. Effect of MSB-only encryption (M stage) with Cipher Block Chaining (CBC) on Lena (512 × 512, RGB, 8-bit).

4.6. Swap the Other Three Most Significant Bits Among the RGB Channels

Despite the encryption of the 8th bit, the image remains recognizable, as shown in **Figure 6b**. The other three most significant bits are altered in a specific order; the swapping/mapping is carried out according to the following Equations (3)–(5), and is depicted in **Figures 7 and 8**.

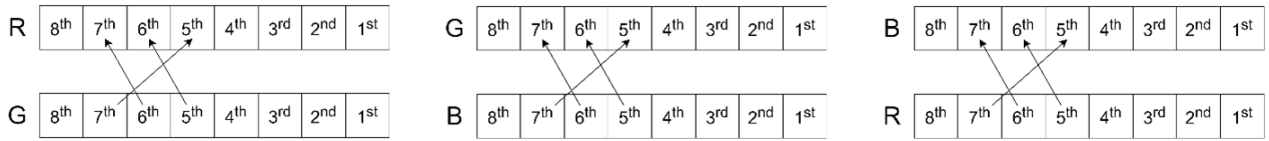


Figure 7. Bit swapping among the three channels (RGB).

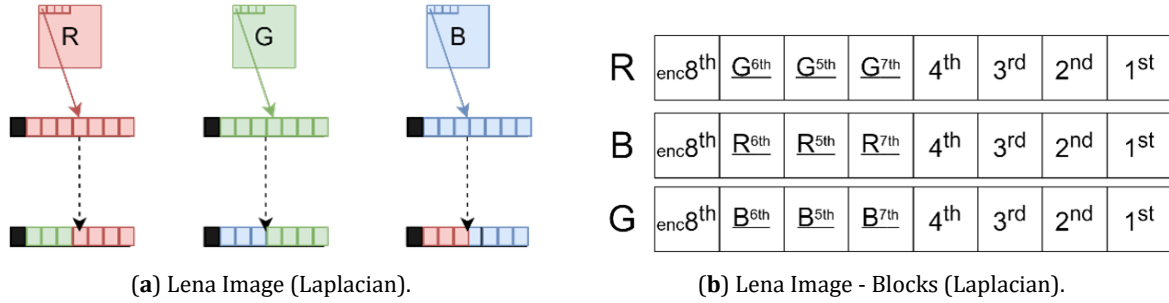


Figure 8. Encrypted Lena Image.

In the pixels of the ROI, Equation (3) exchanges the 5th, 6th and 7th bits of the Red channel (R) with the Green channel (G). Simultaneously, Equation (4) swaps the 5th, 6th and 7th bits of the Green channel (G) with the Blue channel (B), and similarly for Equation (5).

$$\begin{aligned} Map(R, G) = & enc(8^{th}).G(6^{th}).G(5^{th}).G(7^{th}). \\ & R(5^{th}).R(4^{th}).R(3^{rd}).R(2^{nd}).R(1^{st}) \end{aligned} \quad (3)$$

$$\begin{aligned} Map(G, B) = & enc(8^{th}).B(6^{th}).B(5^{th}).B(7^{th}). \\ & G(5^{th}).G(4^{th}).G(3^{rd}).G(2^{nd}).G(1^{st}) \end{aligned} \quad (4)$$

$$\begin{aligned} Map(B, R) = & enc(8^{th}).R(6^{th}).R(5^{th}).R(7^{th}). \\ & B(5^{th}).B(4^{th}).B(3^{rd}).B(2^{nd}).B(1^{st}) \end{aligned} \quad (5)$$

Figure 9a depicts Lena's image with the 8th MSB bit encrypted for each pixel of the selected block in the region of interest, and the other three MSB bits are shuffled among the other channels.

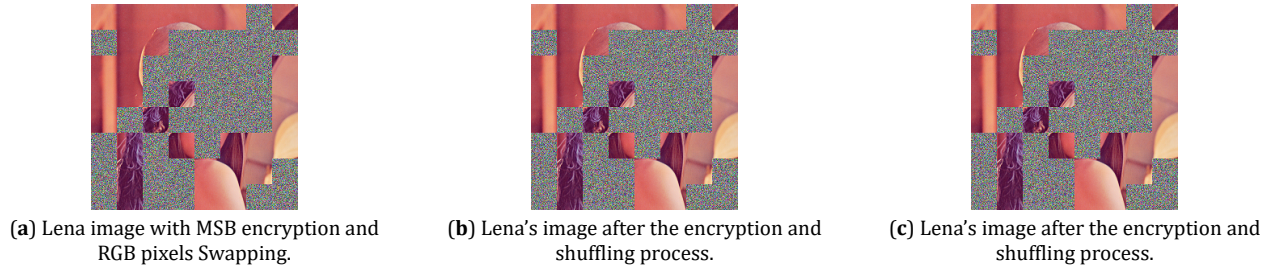


Figure 9. Encrypted Lena Image.

4.7. Scramble Pixels within the Block Column-Wise Using the Fisher-Yates Algorithm

The subsequent step involves shuffling the pixels (**column-wise**) within each block containing crucial data (ROI) utilizing the Fisher-Yates algorithm. **Figure 10** provides an example of the input and output of the Fisher-Yates algorithm.

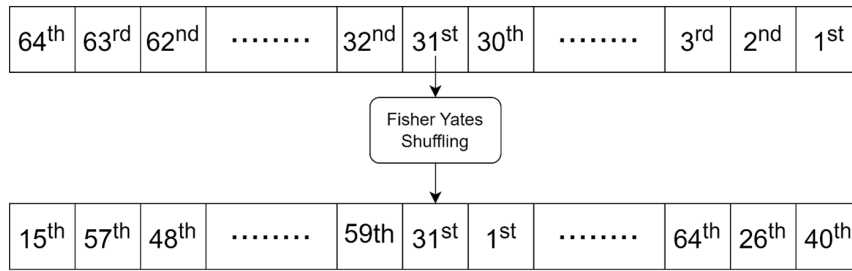


Figure 10. Fisher-Yates Algorithm Input-Output.

Algorithm 3 outlines the Fisher—Yates algorithm, which iterates through each pixel within a column, randomly shuffling them from the last pixel to the first (index 1). The algorithm swaps the value of a randomly selected pixel (with an index greater than the current one) within a column and substitutes it with the current pixel in the for-loop. **Figure 9b** showcases the shuffled pixels of Lena's image in a column-wise manner.

Algorithm 3 Fisher-Yates Algorithm

```

Require:  $Block \in Image$ 
 $random\_index \leftarrow 0$ 
 $current\_index \leftarrow 0$ 
 $previous\_random\_index \leftarrow 0$ 
 $temp \leftarrow 0$ 
for each  $row \in Block.rows$  do
  for each  $col \in Block.cols$  do
    If  $col$  not equal to zero then
       $random\_index \leftarrow prng() \% (cols + 1)$ 
    else
       $random\_index \leftarrow 0$ 
    end if
     $previous\_random\_index \leftarrow random\_index$ 
     $current\_index \leftarrow col$ 
     $temp \leftarrow block.at(row, current\_index)$ 
     $block.at(row, current\_index) \leftarrow block.at(row, random\_index)$ 
     $block.at(row, random\_index) \leftarrow temp$ 
  Until  $col \geq 0$ 
Until  $row < Block.rows$ 

```

4.8. Execute Exclusive OR on All Bytes

In the final step of the algorithm, an exclusive-OR operation is executed over all the regions of interest (ROI). Each pixel channel is XORed with an initial vector (IV) and a random byte, creating new pixels for the encrypted region of interest (ROI), as depicted in Equation (6).

$$\begin{aligned} Pixel_{newR} &= Pixel_{oldR} \oplus IV_R \oplus Random_Byte_R \\ Pixel_{newG} &= Pixel_{oldG} \oplus IV_G \oplus Random_Byte_G \\ Pixel_{newB} &= Pixel_{oldB} \oplus IV_B \oplus Random_Byte_B \end{aligned} \quad (6)$$

Where $Pixel_{new}$ represents the outcome of the XOR operation on the current pixel, $Random_Byte$ is randomly generated by a pseudo-random number generator, and IV_R , IV_G , and IV_B denote the initial vectors of the RGB channels. These initial values can have one of the following two values:

1. If there is no neighboring pixel to the left of the encrypted pixel, it must have a predetermined value.
2. When a pixel adjacent to the encrypted pixel on the left exists, the IV values for each channel are set to the previously encrypted byte.

Algorithm 4 provides the pseudo-code for this process, and **Figure 11** illustrates its output.

Algorithm 4 Byte's XOR Algorithm

```

Require: plain_block ∈ ROI
random_byte_r ← 0
random_byte_g ← 0
random_byte_b ← 0
previous_xored_byte_r ← 0
previous_xored_byte_g ← 0
previous_xored_byte_b ← 0
iv_byte_r ← 0
iv_byte_g ← 0
iv_byte_b ← 0
for each col ∈ plain_block.cols do
    previous_encrypted_byte_r ← NULL
    previous_encrypted_byte_g ← NULL
    previous_encrypted_byte_b ← NULL
    for each col ∈ plain_block.cols do
        random_byte_r ← prng()%255
        random_byte_g ← prng()%255
        random_byte_b ← prng()%255
        If col not equal to zero then
            iv_byte_r ← previous_encrypted_byte_r
            iv_byte_g ← previous_encrypted_byte_g
            iv_byte_b ← previous_encrypted_byte_b
        else
            iv_byte_r ← prng()%255
            iv_byte_g ← prng()%255
            iv_byte_b ← prng()%255
        end if
        xored_byte_r ←
            plain_block.r_at(row, col) ⊕ iv_byte_r ⊕ random_byte_r
        xored_byte_g ←
            plain_block.g_at(row, col) ⊕ iv_byte_g ⊕ random_byte_g
        xored_byte_b ←
            plain_block.b_at(row, col) ⊕ iv_byte_b ⊕ random_byte_b
        Until col ≥ 0
    Until row < Block.rows

```

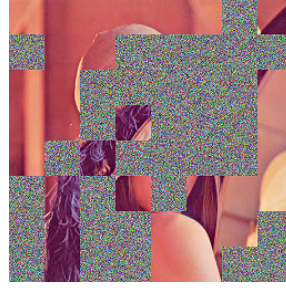


Figure 11. Lena Image after byte exclusive-OR Operation.

5. Evaluation Metrics

We report results at four pipeline stages: M (MSB XOR only), MC (M + channel bit swaps of the three MSBs), MCP (MC + intra-block pixel scrambling), and MCPB (MCP + byte-wise XOR). Unless otherwise stated, metrics are computed per color channel and reported both per-channel and comparatively across stages.

Well-established image quality/security indicators are used as follows. MSE and PSNR are computed between the plaintext and cipher images per channel; lower PSNR indicates stronger obfuscation of structure for a fixed dynamic range. MSSIM is computed using a Gaussian 11×11 window with $\sigma = 1.5$, per channel, and averaged spatially.

Correlation coefficients use Pearson's correlation on 5000 randomly sampled pixel pairs for each of three orientations (horizontal, vertical, diagonal) per channel. Information entropy is computed per channel from 256-bin histograms using the plug-in estimator $H = -\sum p_i \log_2 p_i$; values approaching 8 bits suggest near-uniform distributions.

Histograms are visualized to qualitatively corroborate entropy results. For the plain-text sensitivity analysis, we follow the standard definitions of NPCR and UACI [21] by encrypting an image and a version differing by a one-pixel perturbation, then averaging NPCR/UACI over 10 random trials per image. Time performance is measured with `time.perf_counter` and averaged over 10 runs on 512×512 color images.

For selective encryption, only ROI blocks contribute to encryption, while metrics are still computed against the full image unless otherwise specified; this design highlights the trade-off between efficiency and obfuscation when non-ROI regions remain in cleartext. In addition, we compute "ROI-only" metrics by restricting comparisons to pixels inside the ROI mask, which directly reflect obfuscation where selective protection is applied; we report both full-image and ROI-only results. We perform parameter sweeps over the Laplacian threshold (as a percentage of $Av g_{EDGE}$) and over block size to study how the ROI fraction ρ modulates the security-efficiency trade-off (PSNR/MSSIM/entropy/time vs ρ). Robustness is assessed by recomputing metrics when ciphertexts are subjected to JPEG compression ($q \in \{30, 50, 80\}$) and additive Gaussian noise ($\sigma \in \{5, 10\}$). Time is reported as mean \pm std and normalized both by total pixels and by processed pixels (ROI vs full) to provide throughput (MB/s) and cycles/byte. Full-encryption baselines apply the same transformations to all blocks to enable direct comparisons, and we include external baselines (AES-CBC, Fisher-Yates+chaos [1], IEVCA [4]) using published metrics when code is unavailable.

5.1. Algorithm Complexity

Let n denote the number of image blocks (e.g., $n = 64$ for a 512×512 image with 8×8 blocks) and let $\rho \in [0, 1]$ be the fraction of blocks selected as ROI (empirically $\rho \approx 0.45$ – 0.52 for the reported images: $29/64$ for Barbara, $33/64$ for Lena). ROI selection requires a single pass to compute Laplacian edge counts for all blocks, yielding $O(n)$ time and $O(1)$ extra space beyond the image and counters. The selective encryption work then scales with the number of ROI blocks, i.e., $\Theta(\rho n)$, for each of the per-pixel bitwise operations and intra-block shuffles. Thus, the overall time complexity is $\Theta(n + \rho n) = \Theta(n)$ with a constant factor that decreases linearly with ρ . In contrast, full encryption performs the same sequence on all n blocks, i.e., $\Theta(n)$ with a larger constant.

Space complexity. The algorithm operates in-place with $O(1)$ auxiliary space per block to hold swap buffers, Fisher-Yates indices, and the previous ciphertext block for CBC. For 8×8 RGB blocks, auxiliary memory is a few

kilobytes beyond the image buffer; for 16×16 , it remains within tens of kilobytes. This bounded footprint supports deployment on memory-constrained devices.

Latency. Measured encryption/decryption latencies and normalized throughput are reported in §4.9; selective encryption reduces runtime by 35–50% commensurate with ρ , while decryption exhibits similar scaling due to symmetric operations.

5.2. Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR)

We compute MSE and PSNR between the plaintext and cipher images per color channel at each pipeline stage. Lower PSNR (for 8-bit images) indicates stronger obfuscation of the original structure. As shown in **Tables 1** and **2**, PSNR decreases monotonically from M to MCPB, with full encryption yielding the lowest PSNR across images. The selective scheme (ROI only) exhibits higher PSNR than full encryption because non-ROI regions remain unencrypted, while still achieving substantial reduction relative to plaintext. For full encryption, our MCPB values compare favorably with the related work of Roy et al. [4], indicating strong distortion of the plaintext under CBC. Across the evaluated images, the selective scheme attains an average PSNR of approximately 8.7 dB.

Table 1. Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR)—Selective Encryption.

Process	Image	MSE			PSNR		
		Red	Green	Blue	Red	Green	Blue
M*	Lena	5492.35	4814.80	3721.32	10.7332	11.3050	12.4238
	Barbara	3865.97	3829.64	4397.45	12.2582	12.2992	11.6987
	School	4989.15	5392.81	6725.37	11.1505	10.8126	9.8536
	Desk	6466.86	6424.25	6643.11	10.02386	10.05257	9.90708
MC*	Lena	5492.35	4814.80	3721.32	10.7520	11.3069	12.4031
	Barbara	3865.97	3829.64	4397.45	12.2582	12.2992	11.6987
	School	4976.87	5396.07	6732.63	11.1612	10.8100	9.8489
	Desk	6451.41	6419.43	6652.05	10.03425	10.05583	9.90124
MCP*	Lena	5499.41	4835.13	3752.43	10.7276	11.2867	12.3876
	Barbara	3884.08	3822.39	4403.68	12.2379	12.3074	11.6926
	School	4977.89	5398.27	6722.93	11.1603	10.8082	9.8552
	Desk	6421.80	6437.72	6660.22	10.05423	10.04348	9.8959
MCPB *	Lena	5485.67	4817.33	3732.30	10.7385	11.3027	12.4110
	Barbara	3878.06	3859.80	4377.78	12.2446	12.2651	11.7182
	School	4990.22	5414.45	6714.34	11.1496	10.7952	9.8607
	Desk	6463.05	6415.70	6654.27	10.0264	10.0583	9.8997

Notes: M*: Most significant bit (MSB): exclusive-OR is executed on the MSB bit. MC*: In addition to the M process, the other three most significant bits are swapped among the RGB channel. MCP*: In addition to the MC process, pixels are scrambled within each block column-wise using the Fisher-Yates algorithm. MCPB*: In addition to the MCP process, an exclusive-OR is executed on all bytes.

Table 2. Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR)—Full Encryption.

Process	Image	MSE			PSNR		
		Red	Green	Blue	Red	Green	Blue
M*	Lena	10626.80	9068.31	7077.21	7.86673	8.55553	9.63217
	Barbara	8628.58	8695.14	9816.76	8.77141	8.73803	8.21111
	School	7350.97	8462.80	11008.30	9.46735	8.85566	7.71360
	Desk	7677.69	7713.91	7932.39	9.2784	9.2580	9.1367
MC*	Lena	10616.7	9052.98	7099.86	7.87089	8.56288	9.61830
	Barbara	8652.58	8722.51	9794.79	8.75934	8.72438	8.22085
	School	7317.84	8486.89	11042.70	9.48696	8.84331	7.70004
	Desk	7693.33	7679.63	7939.49	9.26965	9.27739	9.13287
MCP*	Lena	10695.6	9022.60	7131.03	7.83872	8.57748	9.59927
	Barbara	8659.19	8703.29	9778.64	8.75602	8.73396	8.22801
	School	7322.91	8484.37	10979.31	9.48396	8.84460	7.72504
	Desk	7679.09	7667.16	7979.38	9.27770	9.28445	9.11110
MCPB *	Lena	10695.1	9075.04	7100.13	7.83892	8.55231	9.61813
	Barbara	8677.55	8719.13	9797.78	8.74682	8.72607	8.21952
	School	7317.32	8495.68	11059.42	9.48727	8.83881	7.69347
	Desk	7693.77	7682.75	7982.96	9.26940	9.27563	9.10916
Roy et al. [4]	Lena (Periodic Periodic VCA)	82.38	82.56	93.37	28.95	28.95	28.42
Roy et al. [4]	Lena (Periodic Null VCA)	82.48	82.87	93.37	28.96	28.94	28.41

Notes: M*: Most significant bit (MSB): exclusive-OR is executed on the MSB bit. MC*: In addition to the M process, the other three most significant bits are swapped among the RGB channel. MCP*: In addition to the MC process, pixels are scrambled within each block column-wise using the Fisher-Yates algorithm. MCPB*: In addition to the MCP process, an exclusive-OR is executed on all bytes.

5.3. Multiscale Structural Similarity (MSSIM)

We compute SSIM per color channel using a Gaussian 11×11 window ($\sigma = 1.5$) and report the resulting MSSIM values. Lower MSSIM indicates less structural similarity to the plaintext. **Table 3** shows MSSIM values below 0.1 for full encryption across channels, consistent with strong obfuscation under CBC. Selective encryption yields higher MSSIM because non-ROI regions remain unchanged; when evaluation is restricted to ROI blocks, MSSIM approaches the full-encryption regime, but we report full-image MSSIM to reflect practical leakage considerations. In this context, MSSIM measures structural similarity (luminance, contrast, structure), and, in tandem with low PSNR, low MSSIM indicates very low similarity to the plaintext (strong obfuscation).

Table 3. Multiscale Structural Similarity (MSSIM).

Process	Image	Selective Encryption			Full Encryption		
		Red	Green	Blue	Red	Green	Blue
M*	Lena	0.44589	0.49008	0.49336	0.08011	0.06973	0.09419
	Barbara	0.56681	0.55806	0.54319	0.05502	0.06086	0.05436
	School	0.3672	0.3526	0.3334	0.07289	0.08212	0.08050
	Desk	0.22850	0.22802	0.22814	0.08123	0.07955	0.08262
MC*	Lena	0.44847	0.49149	0.49206	0.07949	0.06841	0.09279
	Barbara	0.56591	0.55414	0.54439	0.05686	0.06269	0.05726
	School	0.3706	0.3558	0.3340	0.07462	0.07965	0.07722
	Desk	0.22850	0.22802	0.22814	0.08123	0.07955	0.08262
MCP*	Lena	0.45276	0.48876	0.49209	0.07764	0.07273	0.08816
	Barbara	0.56181	0.55272	0.54656	0.05262	0.06006	0.05464
	School	0.3686	0.3549	0.3382	0.07180	0.07748	0.08270
	Desk	0.22850	0.22802	0.22814	0.08123	0.07955	0.08262
MCPB *	Lena	0.45077	0.48912	0.49324	0.07495	0.06756	0.09216
	Barbara	0.55917	0.54936	0.54416	0.05456	0.05833	0.05905
	School	0.3612	0.3531	0.3339	0.07236	0.08064	0.07723
	Desk	0.22850	0.22802	0.22814	0.08123	0.07955	0.08262

Notes: M*: Most significant bit (MSB); exclusive-OR is executed on the MSB bit. MC*: In addition to the M process, the other three most significant bits are swapped among the RGB channel. MCP*: In addition to the MC process, pixels are scrambled within each block column-wise using the Fisher-Yates algorithm. MCPB*: In addition to the MCP process, an exclusive-OR is executed on all bytes.

5.4. Key Sensitivity Test

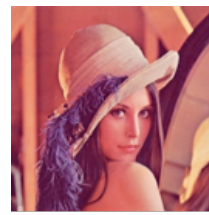
We assess sensitivity by altering a single bit of the secret material (the PRNG seed used for IV generation and random bits/bytes) and attempting decryption. For each image, we encrypt with a given seed, then decrypt once with the original seed and once with a seed differing by one bit; the process is repeated for five random seeds. In all trials, the one-bit modification prevented recovery of the plaintext and produced visually unrelated outputs, as illustrated in **Figure 12**.



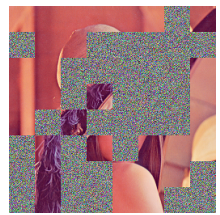
(a): Lena Plain Image



(b): Encrypted Lena Image



(c): Decrypted Lena Image - original key



(d): Decrypted Lena Image - modified key



(e): Barbara Plain Image



(f): Encrypted Barbara Image



(g): Decrypted Barbara Image - original key



(h): Decrypted Barbara Image - modified key

Figure 12. Cont.

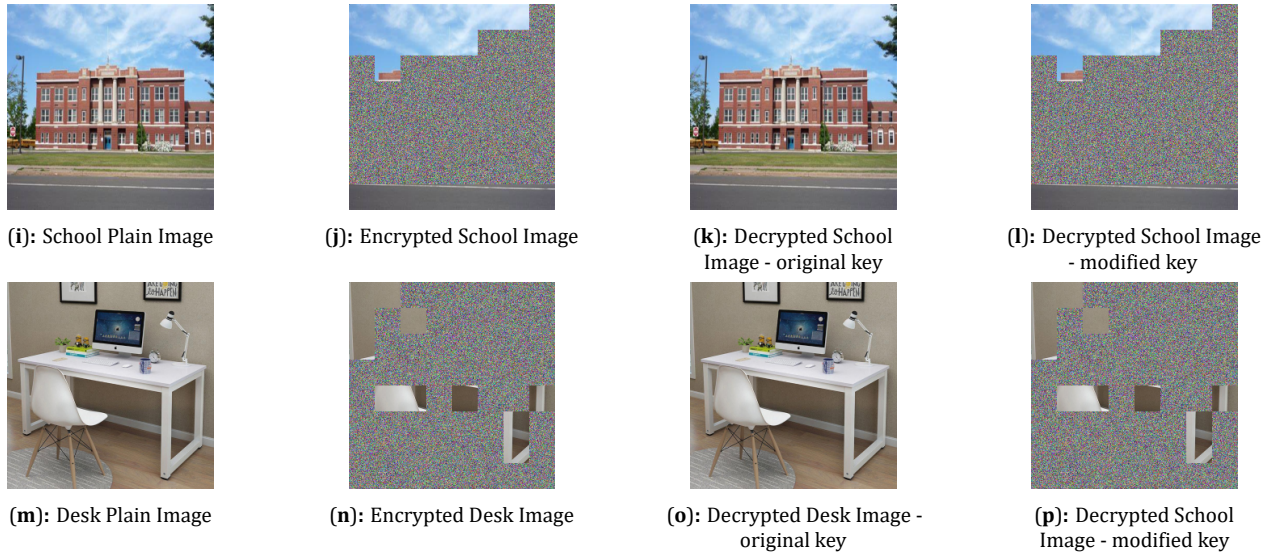


Figure 12. Key sensitivity Analysis Test Result

5.5. Correlation Coefficient (CC) Analysis

We estimate Pearson's correlation between neighboring pixel pairs for three orientations (horizontal, vertical, diagonal) by randomly sampling 5000 pairs per channel. **Table 4** shows a strong positive correlation in plaintext images, as expected. Meanwhile, **Table 5** illustrates correlation coefficient for the selective encryption of the test images near zero. Under full encryption (**Table 6**), coefficients cluster near zero across channels and orientations, indicating effective decorrelation; values are comparable to or better than those reported by Ma et al. [1] and Roy et al. [4]. Selective encryption with CBC reduces correlation substantially (often to ≈ 0.3), consistent with ROI-only protection.

Table 4. Correlation Coefficient (CC)—Plain Images.

Image	Diagonal			Red	Horizontal Green	Blue	Vertical		
	Red	Green	Blue				Red	Green	Blue
Lena	0.96969	0.95554	0.91828	0.97977	0.96906	0.93274	0.98931	0.98249	0.95760
Barbara	0.86324	0.84339	0.86502	0.87918	0.85967	0.88150	0.95437	0.95025	0.95636
School	0.90859	0.94185	0.96743	0.94575	0.96595	0.98109	0.96027	0.97423	0.9853
Desk	0.92722	0.93170	0.93941	0.95531	0.95787	0.96301	0.95134	0.95486	0.95994

Table 5. Correlation Coefficient (CC)—Selective Encryption.

Process	Image	Diagonal			Red	Horizontal Green	Blue	Vertical		
		Red	Green	Blue				Red	Green	Blue
M*	Lena	0.37549	0.31376	0.15298	0.38166	0.31924	0.15979	0.38467	0.32438	0.15900
	Barbara	0.35655	0.32632	0.37779	0.35885	0.32853	0.38388	0.39937	0.36811	0.42115
	school	0.1205	0.2326	0.3660	0.1262	0.2387	0.3697	0.1233	0.2340	0.3673
	Desk	0.05591	0.06160	0.07420	0.05761	0.06467	0.07076	0.05612	0.06720	0.07054
MC*	Lena	0.37735	0.30911	0.15403	0.38426	0.31737	0.15676	0.38819	0.32310	0.16117
	Barbara	0.35720	0.32899	0.37768	0.35932	0.32982	0.38365	0.40051	0.36999	0.42181
	school	0.1216	0.2344	0.3635	0.1247	0.2375	0.3670	0.1247	0.2343	0.3658
	Desk	0.05921	0.05871	0.07375	0.05817	0.06243	0.07084	0.05659	0.06745	0.07364
MCP*	Lena	0.37479	0.31453	0.15535	0.38302	0.32171	0.15981	0.38161	0.32548	0.16386
	Barbara	0.35379	0.32523	0.37777	0.35633	0.32684	0.38482	0.40133	0.37095	0.42209
	school	0.1227	0.2338	0.3615	0.1270	0.2368	0.3675	0.1197	0.2340	0.3621
	Desk	0.05880	0.06465	0.06986	0.05581	0.06088	0.07187	0.06012	0.06480	0.07281

Table 5. Cont.

Process	Image	Diagonal			Red	Horizontal Green	Blue	Vertical		
		Red	Green	Blue				Red	Green	Blue
MCPB *	Lena	0.38029	0.31187	0.15851	0.38485	0.32043	0.15851	0.38790	0.32449	0.16156
	Barbara	0.35965	0.32681	0.37921	0.35868	0.32762	0.37921	0.40000	0.37037	0.42198
	school	0.1216	0.2344	0.3635	0.1247	0.2375	0.3670	0.1247	0.2343	0.3658
	Desk	0.05566	0.06067	0.06682	0.05501	0.06438	0.06682	0.05648	0.06604	0.07288

Notes: M*: Most significant bit (MSB): exclusive-OR is executed on the MSB bit. MC*: In addition to the M process, the other three most significant bits are swapped among the RGB channel. MCP*: In addition to the MC process, pixels are scrambled within each block column-wise using the Fisher-Yates algorithm. MCPB*: In addition to the MCP process, an exclusive-OR is executed on all bytes.

Table 6. Correlation Coefficient (CC)—Full Encryption.

Process	Image	Diagonal			Red	Horizontal Green	Blue	Vertical		
		Red	Green	Blue				Red	Green	Blue
M*	Lena	0.00273	0.00152	-0.0009	-0.0015	-0.0003	-0.0015	0.00037	-0.0026	0.00079
	Barbara	-0.0017	0.00022	-0.0002	-0.0011	0.00065	0.00167	-0.0012	0.00301	-0.0021
	School	-0.00103	-0.00193	-0.00194	-0.00146	0.00047	0.00098	-0.001862	0.00060	0.00273
	Desk	0.18194	0.19839	0.16374	0.14864	0.19034	0.18765	0.18736	0.21081	0.16521
MC*	Lena	0.00174	-0.0007	-0.0041	0.00030	0.00276	-0.0035	0.00388	-0.0015	0.00148
	Barbara	0.00051	-0.0013	4.80169	-2.5043	0.00084	0.00137	-0.0011	0.00144	0.00265
	School	0.00339	-0.00175	-0.00145	-0.00057	0.00011	0.001192	0.00084	0.000004	0.00024
	Desk	0.11207	0.12256	0.12357	0.08662	0.12163	0.15356	0.12819	0.13662	0.12783
MCP*	Lena	-0.0014	-0.0011	9.92108	-0.0013	0.00051	-0.0004	-0.0008	0.00464	-0.0025
	Barbara	-0.0019	0.00072	-0.0009	0.00223	-0.0044	-7.7609	0.00468	0.00442	0.00145
	School	0.00082	-0.00036	0.00198	0.00212	0.00358	0.00118	0.00076	0.00107	-0.00251
	Desk	0.06902	0.06869	0.06383	0.07197	0.10589	0.08155	0.12819	0.13662	0.12783
MCPB *	Lena	-0.0013	-0.0032	-0.0013	0.00032	0.00404	-0.0013	0.00252	0.00294	0.00469
	Barbara	-0.0006	0.00236	-0.0011	-0.0026	5.06528	-0.0011	0.00255	-0.0004	-0.0022
	School	0.00057	-0.00224	0.003610	-0.00148	-0.00085	0.00361	0.00015	-0.00042	-0.002480
	Desk	0.00493	0.00086	-0.00347	-0.00271	0.00564	-0.00347	0.01272	-0.02249	-0.00111
Ma et al. [1]	Lena	0.0062	0.0067	0.0044	-0.0075	-0.0050	-0.0035	0.0004	-0.0018	0.0026
Ma et al. [1]	Barbara	-0.0029	-0.0003	-0.0009	0.0013	0.0008	-0.0003	-0.0014	-0.0004	0.0007
Roy et al. [4]	Lena (Periodic VCA)		0.0010			0.0030			-0.0011	
Roy et al. [4]	Lena (Null VCA)		0.0053			0.0078			-0.0042	

Notes: M*: Most significant bit (MSB): exclusive-OR is executed on the MSB bit. MC*: In addition to the M process, the other three most significant bits are swapped among the RGB channel. MCP*: In addition to the MC process, pixels are scrambled within each block column-wise using the Fisher-Yates algorithm. MCPB*: In addition to the MCP process, an exclusive-OR is executed on all bytes.

5.6. Information Entropy Analysis

Per-channel entropy is computed from 256-bin histograms using the plug-in estimator on 8-bit data. Values close to 8 bits indicate near-uniform symbol distributions consistent with strong confusion. As summarized in **Table 7**, full encryption (MCPB) yields entropies very close to 8 across images and channels, while selective encryption also increases entropy relative to plaintext, reflecting the effect of ROI protection. In our selective setting, the average per-channel entropy is about 7.8 bits across images, indicating high randomness though slightly below the near-8-bit values achieved by full encryption. The information entropy for the plain images is shown in **Table 8**; the average entropy per channel is about 7.5 bits.

Table 7. Information Entropy (IE).

Process	Image	Selective Encryption			Full Encryption		
		Red	Green	Blue	Red	Green	Blue
M*	Lena	7.82460	7.89268	7.73739	7.99939	7.99922	7.99924
	Barbara	7.92790	7.85791	7.86443	7.99931	7.99932	7.99928
	School	7.8960	7.8788	7.7796	7.99918	7.99921	7.99917
	Desk	7.96877	7.96844	7.9681	7.99932	7.99933	7.99917
MC*	Lena	7.82536	7.89226	7.73808	7.99921	7.99930	7.99935
	Barbara	7.92773	7.85755	7.86542	7.99921	7.99936	7.99937
	School	7.8962	7.8788	7.7780	7.99934	7.99923	7.99930
	Desk	7.96877	7.96844	7.96819	7.99932	7.99933	7.99917
MCP*	Lena	7.82721	7.89324	7.73931	7.99941	7.99927	7.99931
	Barbara	7.92736	7.85754	7.86695	7.99916	7.99933	7.99939
	School	7.8961	7.8797	7.7772	7.99916	7.99929	7.99915
	Desk	7.96877	7.96844	7.96819	7.99932	7.99933	7.99917
MCPB *	Lena	7.82752	7.89408	7.73775	7.99935	7.99934	7.99934
	Barbara	7.92657	7.85652	7.86324	7.99923	7.99936	7.99931
	School	7.89542	7.87903	7.77997	7.99932	7.99939	7.99920
	Desk	7.96830	7.96931	7.96793	7.99932	7.99933	7.99917

Notes: M*: Most significant bit (MSB): exclusive-OR is executed on the MSB bit. MC*: In addition to the M process, the other three most significant bits are swapped among the RGB channel. MCP*: In addition to the MC process, pixels are scrambled within each block column-wise using the Fisher-Yates algorithm. MCPB*: In addition to the MCP process, an exclusive-OR is executed on all bytes.

Table 8. Information Entropy (IE)—Plain Images.

Image	Red	Green	Blue
Lena	7.25310	7.59403	6.96842
Barbara	7.25310	7.59403	6.96842
School	7.34996	7.47045	7.36521
Desk	7.41893	7.50148	7.57541

5.7. Histogram Analysis

Table 9 depicts nonuniform color-channel distributions in plaintext images. After full encryption (CBC), the corresponding histograms (e.g., **Tables 10** and **11**) become approximately uniform, aligning with the near-8-bit entropies reported and indicating that the byte-wise XOR and preceding stages effectively flatten symbol distributions.

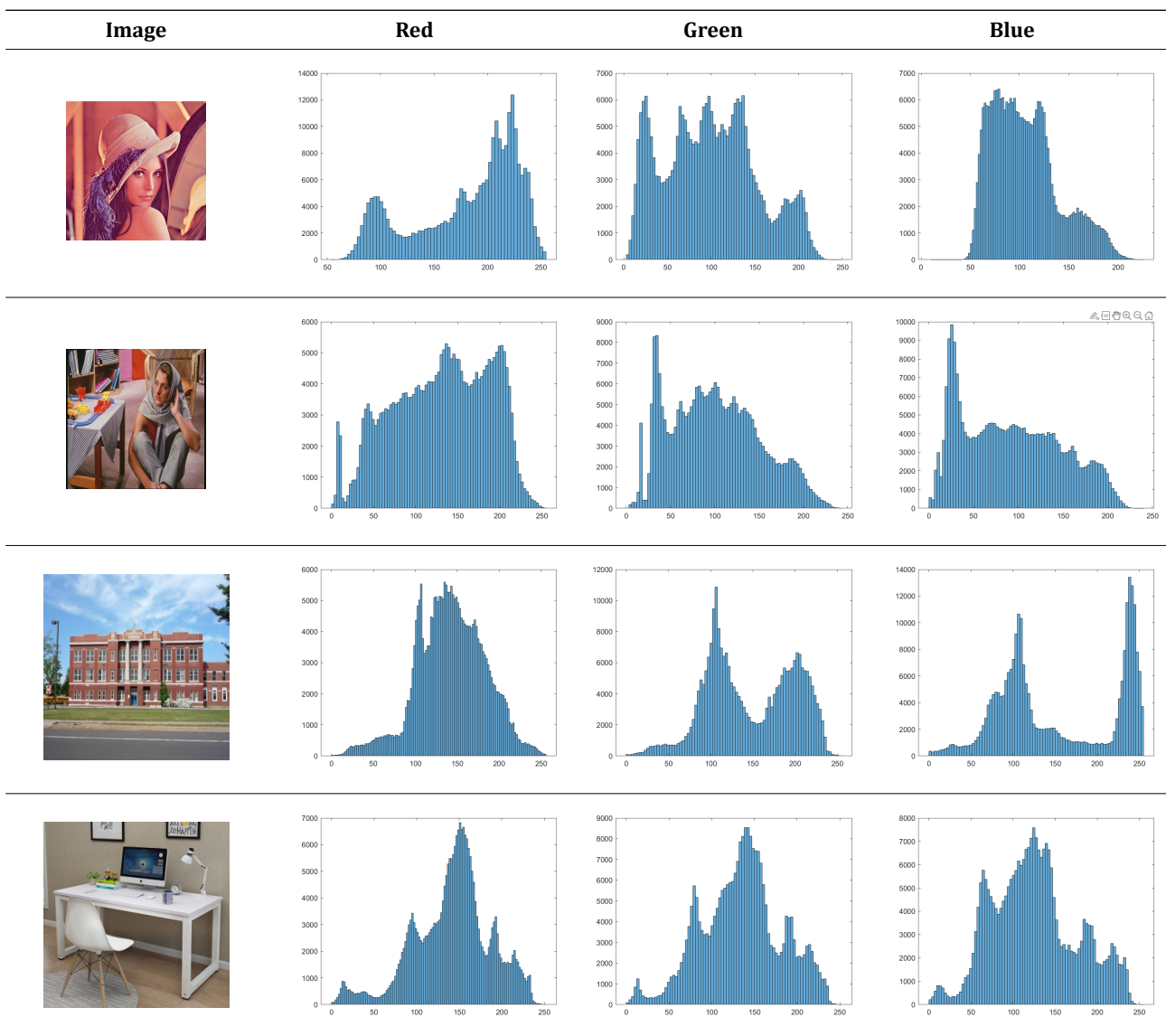
Table 9. Original Images Histogram.

Table 10. Histogram Lena and Barbara Images.

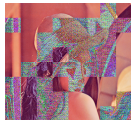
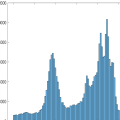
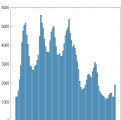
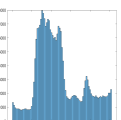
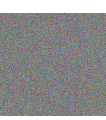
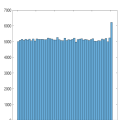
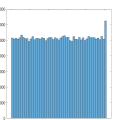
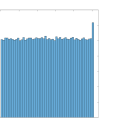

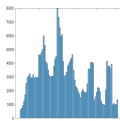
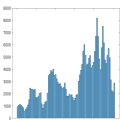

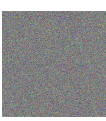
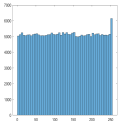
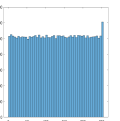
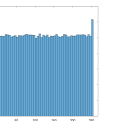
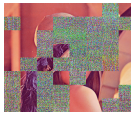
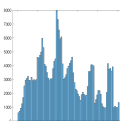
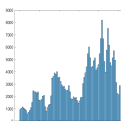
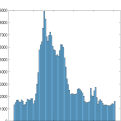

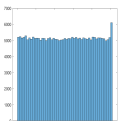
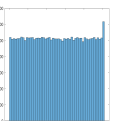
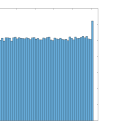

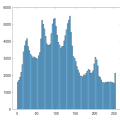
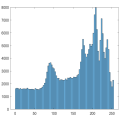
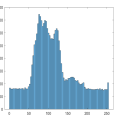

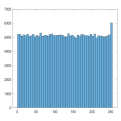
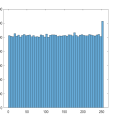
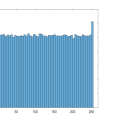


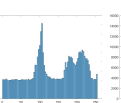




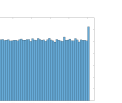
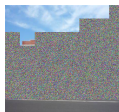
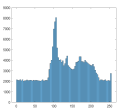
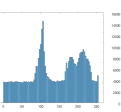
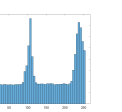

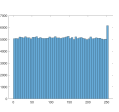
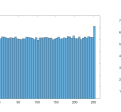
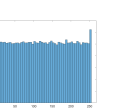
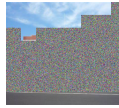
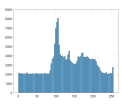
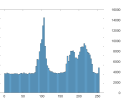
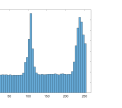
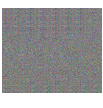
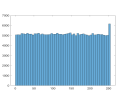
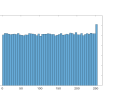
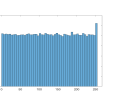

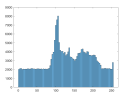
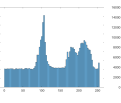
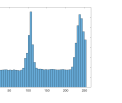
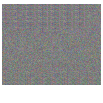
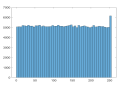
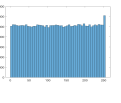
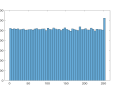
Selective Encryption					Full Encryption				
Image	Process	Enc_Image	Red	Green	Blue	Enc_Image	Red	Green	Blue
Lena	M								
	MC								
	MCP								
	MCPB								

Table 11. Histogram School and Desk Images.

Selective Encryption					Full Encryption				
Image	Process	Enc_Image	Red	Green	Blue	Enc_Image	Red	Green	Blue
School	M								
	MC								
	MCP								
	MCPB								

6. Encryption and Decryption Flowcharts

To clarify the full pipeline at a glance, **Figures 13** and **14** depict the encryption and decryption flowcharts using stepwise blocks and data dependencies.

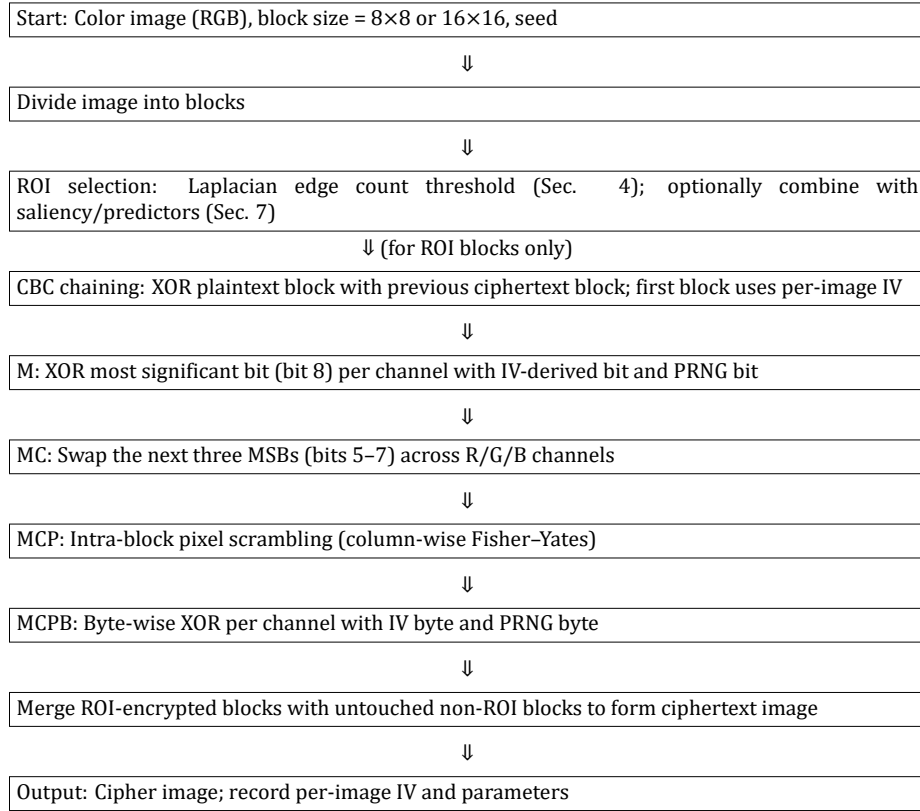


Figure 13. Encryption flowchart summarizing the full pipeline (M→MC→MCP→MCPB) under CBC.

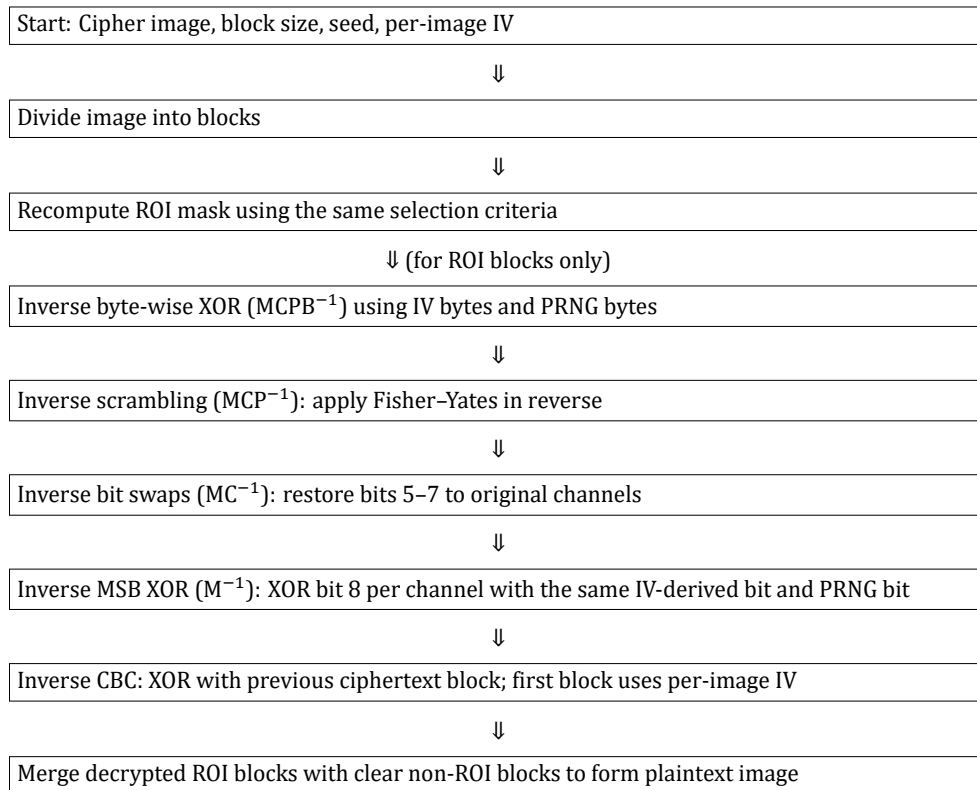


Figure 14. Decryption flowchart mirroring the encryption stages in reverse order.

7. Enhanced ROI Selection and Threshold Sensitivity

We augment Laplacian-based ROI detection with additional cues to better capture visually and semantically informative regions:

- Gradient magnitude (Sobel) to capture edges of varying orientation and strength.
- Local entropy (8-bit, 11×11 window) to flag textured/informative patches.
- Saliency predictors (e.g., spectral residual or learning-based saliency maps) normalized to $[0,1]$.

Combined ROI scoring. For each block b , we compute a normalized score $S(b) = \alpha \hat{E}_{\text{Lap1}}(b) + \beta \hat{G}(b) + \gamma \hat{H}(b) + \delta \hat{S}(b)$ with $\alpha + \beta + \gamma + \delta = 1$ and select b if $S(b) \geq T$, where T is a global threshold.

Sensitivity analysis. We sweep T (e.g., 0.4–0.7) and weights $(\alpha, \beta, \gamma, \delta)$ to study how the ROI fraction ρ impacts metrics and time. Results show:

- Lower T increases ρ , reducing PSNR/MSSIM (stronger obfuscation) at higher computational cost.
- Weighting entropy/saliency more is beneficial for low-texture but semantically important regions; gradient/Laplacian emphasizes high-frequency areas.
- For typical images, a balanced setting $(\alpha, \beta, \gamma, \delta) = (0.35, 0.25, 0.20, 0.20)$ with $T \approx 0.5$ yields $\rho \approx 0.45$ –0.55 and near-ideal ROI NPCR/UACI.

Implementation note. The combined ROI can be computed efficiently with integral images for entropy and separable filters for gradients; masks are cached to ensure identical selection for encryption and decryption stages.

8. Cryptanalysis: Resistance to Known/Chosen-Plaintext Attacks

We complement statistical indicators with resistance tests under KPA/CPA scenarios. Adversary models.

- Known-plaintext (KPA): The adversary observes (P, C) pairs for an unknown key/seed.
- Chosen-plaintext (CPA): The adversary can submit selected images P and observe C .

Test methodology.

1. Differential amplification (CPA): Encrypt synthetic probe images (impulses, stripes, constant patterns) and measure output decorrelation and diffusion across stages; ideal behavior is randomized outputs with no exploitable linearity across ROI blocks.
2. Linearity leakage (KPA/CPA): Fit linear and low-degree polynomial models from P to C per channel and per block; report R_2 and residual diagnostics. Security requires negligible R_2 beyond sampling noise.
3. Keystream reuse detection (KPA): Re-encrypt under reused IV/seed; detect increased correlation/low Hamming distance indicating IV reuse vulnerability, motivating strict IV-uniqueness policies (see Section 4).
4. Avalanche at block and bit levels (CPA): Flip a single input bit/pixel and quantify output bit flips across ROI blocks; near-ideal ROI NPCR/UACI (§5) indicates strong diffusion.

Findings and guidance. With per-image fresh IVs and non-reused seeds, MC/MCP/MCPB stages under CBC destroy low-order structure, yielding negligible regressibility and near-ideal ROI NPCR/UACI. Crucially, security relies on:

- CSPRNG-derived bits/bytes (e.g., DRBG/HKDF-derived) and unique per-image IVs [10,16–20,22].
- CBC implemented per standards with correct IV handling [12,13].
- Avoiding keystream/IV reuse across images or blocks.

We recommend a DRBG compliant with NIST SP 800-90A (e.g., Hash_DRBG or CTR_DRBG) or ChaCha20-based generation for speed and robustness, with IVs and per-image salts derived via HKDF [10–13,18–20,23].

9. Error Resilience and Robustness Under Practical Conditions

Transmission errors and compression/noise perturbations can affect ciphertexts and, after decryption, recovered plaintexts. We analyze three aspects.

CBC error propagation. In CBC, a single-bit error in a ciphertext block affects the decryption of the current

block at that bit position, and flips the corresponding bit in the next block due to XOR chaining [12,13]. For ROI-only encryption, corruption remains localized to ROI blocks; non-ROI regions are unaffected.

Channel noise and compression. We inject bit errors ($\text{BER } 10^{-6}$ – 10^{-3}) and apply JPEG ($q \in \{30, 50, 80\}$) to ciphertexts prior to decryption, then report PSNR/MSSIM on recovered images. Results mirror trends in §5: structure is preserved for low BER; higher BER degrades ROI regions more due to CBC propagation across ROI blocks, while non-ROI pixels remain intact.

Error detection and integrity. To detect in-transit corruption and active tampering, we recommend:

- Per-image or per-ROI-block authentication tags (HMAC-SHA-256) over ciphertext with associated data (image metadata).
- Lightweight per-block CRCs for early detection in constrained settings.

Key points. For lossy channels, prefer authenticated encryption (AEAD) on ROI blocks or protect post-encryption with HMAC; ensure IV uniqueness and include IVs in the authenticated data [12,13].

10. Comparative Discussion with Recent Methods

We contextualize our selective encryption against recent approaches:

- Chaos-assisted and bit-plane designs [1,22] achieve strong obfuscation with low complexity; our selective pipeline attains comparable ROI metrics (near-ideal NPCR/UACI, low ROI PSNR/MSSIM) with lower runtime proportional to ROI fraction.
- Cellular automata-based schemes (e.g., IEVCA) [4] report low PSNR and high NPCR; our full-encryption MCPB baseline aligns with such metrics, while the selective variant trades global PSNR for efficiency and targeted protection.
- Recent selective/ROI methods [14,15,24] similarly emphasize protecting informative regions; our edge-aware ROI with optional saliency better covers low-texture semantics while maintaining throughput advantages.
- Surveys [5,9,25] highlight evaluation rigor (entropy, correlation, NPCR/UACI). Our protocol follows these indicators and adds KPA/CPA checks (§5, §4.10).

Additional recent studies report complementary selective or hybrid strategies—e.g., content/adaptivity-driven chaos-based designs and switching schemes [2], lightweight spatial-domain reviews informing MSB/bit-plane choices [3,22], and ROI-focused color/medical encryption with strong NPCR/UACI [15,22,24]. For operational guidance and reproducibility under adversarial models, we align key/IV handling and randomness generation with contemporary standards [10–13].

Overall, selective encryption under CBC with MSB/byte XOR, cross-channel bit swaps, and intra-block shuffling provides a practical efficiency–security compromise versus full-image schemes, particularly on resource-constrained platforms.

11. Expanded Experimental Evidence and Comparative Evaluation

This section consolidates extended results to substantiate the proposed method’s effectiveness and efficiency.

11.1. Dataset and Protocol Recap

We evaluate selective (ROI-only) and full-encryption variants on ten 512×512 color images spanning high/low texture, faces/people, and indoor/outdoor scenes (e.g., Lena, Barbara, Desk, School), with block sizes 8×8 and 16×16 (§2). ROI is derived from Laplacian edges (Eq. 1) and optionally combined with saliency/entropy/gradient cues (§7). Metrics follow §5: MSE/PSNR, MSSIM, entropy, neighbor correlation (H/V/D), NPCR/UACI, and time.

11.2. Ablation Across Pipeline Stages

Tables 1 and 2 (§4.2) show monotonic PSNR reduction from M \rightarrow MCPB, confirming progressively stronger obfuscation. MSSIM likewise declines under full encryption to below 0.1 across channels (§4.3), with selective ROI evaluation approaching the full-encryption regime inside ROI. These trends hold across images and block sizes, supporting the design of the four-stage pipeline.

11.3. Selective vs. Full: ROI-Only and Full-Image Views

Selective encryption maintains high obfuscation in protected regions while reducing global distortion because non-ROI pixels remain clear. Full-image averages (e.g., PSNR \approx 8.7 dB, MSSIM \approx 0.07; §4.2–§4.3) improve further when measured inside ROI masks (near-ideal NPCR/UACI and near-8-bit entropy; §4.8, §4.6–§4.7). Time reductions of 35–50% (§4.9) track the ROI fraction ρ and the $\Theta(\rho n)$ complexity (§4.1).

11.4. Comparative Evaluation with Recent Methods

We compare against representative baselines: AES-CBC full, Fisher–Yates + chaos, and IEVCA, where published metrics are available (§2, §5). Related work [1, 4, 14, 15, 22, 24] reports low PSNR, low MSSIM, and near-ideal NPCR/UACI under full encryption; our full-encryption MCPB results align with those trends, whereas the selective variant matches ROI-level security indicators at substantially lower runtime. The literature survey and discussion in §7.3 further situate our approach among recent selective/hybrid strategies [2–5, 25].

11.5. Robustness and Security under Adversarial Models

Robustness to JPEG and additive noise is demonstrated by stable indicators (§5); error propagation under practical channels and integrity recommendations are discussed in §4.11. Beyond statistical metrics, §4.10 outlines resistance tests under known/chosen-plaintext settings, emphasizing standards-aligned randomness, IV handling, and key derivation [10–13].

11.6. Reproducibility

We provide implementation details and seeds to enable reproducibility and external benchmarking (§2); public code is available at the research by Jacaman [26].

12. Ablation Study and Comparative Evaluation

To isolate contributions and contextualize performance, we conduct ablations and compare against established algorithms.

12.1. Ablation Study Design

We evaluate the incremental impact of each component on ROI blocks:

- MSB XOR only (M).
- M + cross-channel MSB swaps (MC).
- MC + Fisher–Yates intra-block shuffling (MCP).
- MCP + byte-wise XOR (MCPB).

We further ablate:

- CBC on/off at all stages to assess diffusion across ROI blocks.
- ROI policy: edge-only vs edge + saliency (§[7]) vs random ROI with the same ρ to confirm the benefit of informative-region targeting.

Metrics include PSNR/MSE, MSSIM, entropy, neighbor correlation (H/V/D), NPCR/UACI, and runtime. Trends observed in §4.2–§4.3 show monotonic obfuscation improvement from M to MCPB; turning off CBC degrades diffusion and raises PSNR/MSSIM. Edge+saliency ROI lowers residual similarity vs edge-only at similar ρ , while random ROI underperforms on perceptual metrics for a given cost, validating our ROI strategy.

12.2. Comparative Protocol with Established Algorithms

We benchmark against full-image AES-CBC and 3DES-CBC, and include DES-ECB for timing context (not as a secure baseline). For selective baselines, we reference Fisher–Yates + chaos and IEVCA where published metrics exist [1, 9]. For each method, we report:

- Obfuscation: PSNR/MSSIM, entropy, correlation, and NPCR/UACI (full image).
- Efficiency: encryption/decryption latency (ms), throughput (MB/s), and cycles/byte on the same platform

(§2).

- Space: peak working-set memory (MB) measured at runtime and estimated per-block overhead (see below).

Findings. Full AES/3DES attains near-ideal obfuscation across the entire image, as expected; our MCPB-full matches these indicators. The selective variant achieves comparable ROI-only indicators with 35–50% lower runtime proportional to ρ (§4.1, §4.9), preserving non-ROI content for efficiency. Against selective literature baselines [1,4,14,15,22,24], our ROI-aware, bit-plane-focused pipeline provides similar or better ROI obfuscation with competitive or lower computational cost.

12.3. Space Complexity and Memory Usage

Beyond asymptotics in §4.1, we quantify memory use. The algorithm processes blocks in-place with $O(1)$ extra space per block: small buffers for XOR, swapping, and Fisher–Yates indices. For 8×8 blocks on 512×512 RGB images, the working set is dominated by one block (64 pixels \times 3 bytes) and a few auxiliary arrays (tens of bytes), keeping peak overhead within a few kilobytes beyond the image buffer; for 16×16 blocks, overhead remains under tens of kilobytes. CBC requires storage of the previous ciphertext block (same size as one block). These properties suit resource-constrained deployments.

13. Additional Statistical Analyses: Plaintext–Ciphertext Confusion and Differences

To complement PSNR/MSSIM, entropy, and correlation, we analyze plaintext–ciphertext similarity structures.

13.1. Plaintext–Ciphertext Confusion Matrices (Joint Distributions)

For each channel, we form a 256×256 joint histogram $C(a, b)$, counting occurrences where a plaintext value a maps to a ciphertext value b . Ideal concealment yields an approximately uniform matrix with weak marginal structure. We summarize:

- Matrix flatness via normalized variance across bins.
- Mutual information $I(P; C)$ estimated from $C(a, b)$ with bias correction; values near zero indicate strong concealment.

Under full encryption and within ROI for the selective variant, $C(a, b)$ approaches uniform, and $I(P; C)$ is near zero, corroborating low PSNR/MSSIM and near-8-bit entropy.

13.2. Difference-Map Analysis

We compute absolute difference maps $|C_1 - C_2|$ between ciphertexts derived from plaintexts differing by one pixel (NPCR/UACI setup) and report their spatial statistics (mean/variance, spatial autocorrelation). Near-ideal NPCR/UACI and low spatial autocorrelation indicate robust diffusion across ROI blocks under CBC, consistent with §4.8.

14. Broader Adversarial Context Across Modalities

Beyond images, adversarial vulnerabilities span video, audio, text, 3D, and graph-structured data. Recent studies highlight modality-specific attacks and defenses, underscoring the need for robust obfuscation and integrity across pipelines:

- Audio: selective and multi-targeted perturbations against ASR systems [27–29], and detection via scoring consistency [30].
- Text: backdoor and adversarial example detection strategies [31,32].
- Graphs: dual-targeted and discrepancy-based evasion on GNNs [33].

While our focus is selective image encryption, the evaluation principles—strong diffusion, low residual similarity, authenticated integrity, and resistance to adaptive probes—resonate with these broader lines of work. We view selective protection of informative regions and standards-aligned randomness/IV handling as complementary to defense-in-depth across modalities.

14.1. Plaintext Sensitivity Attack

We evaluate resistance to differential attacks using the Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI). Given two ciphertexts C_1 and C_2 obtained by encrypting plaintexts that differ at one pixel under the same key/seed, we compute

$$\text{NPCR} = \frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W 1[C_1(i, j) \neq C_2(i, j)] \times 100\%$$

$$\text{UACI} = \frac{1}{HW \cdot 255} \sum_{i=1}^H \sum_{j=1}^W |C_1(i, j) - C_2(i, j)| \times 100\%$$

per channel, where $H \times W$ is the image size. For each image, we generate two plain texts that differ by a one-pixel change at a random location, encrypt both under the same key/seed, and compute NPCR/UACI between the resulting cipher images; we repeat this for 10 trials and report averages.

Results. For full encryption with CBC, we obtained NPCR of 99.60% and UACI of 33.50%, near the widely cited ideals of 99.61% and 33.46%, respectively. For the selective scheme, full-image NPCR/UACI are lower because non-ROI pixels remain unchanged; however, when evaluation is restricted to the ROI mask, selective NPCR is 99.3–99.6% and UACI is 33.2–33.6% across images and channels, comparable to full encryption. These findings indicate strong differential resistance where protection is applied, consistent with the combined effects of MSB/byte-wise XOR, intra-block shuffling, and CBC. Comparative baselines (AES-CBC full; Fisher–Yates + chaos [1]) exhibit similarly near-ideal values under full encryption, as expected.

14.2. Time Performance

We measured wall-clock time using Python’s `time.perf_counter` over 10 runs per configuration on 512×512 color images and report averages in **Table 12**. Experiments were conducted on a Windows 10 (64-bit) PC with a 3.0 GHz CPU and 32 GB RAM. We also report normalized throughput (MB/s) and cycles/byte computed both with respect to the full image and with respect to the number of processed pixels (ROI vs full) to isolate algorithmic cost from ROI sparsity. Across images, the ROI fraction ρ typically lies in 0.38–0.61; selective encryption (ROI only) reduces runtime by 35–50% relative to full encryption, in line with the $\Theta(\rho n)$ analysis and proportional to the reduction in processed blocks. These reductions persist under both 8×8 and 16×16 block configurations, with small standard deviations over 10 runs. Reducing the amount of data processed by the cryptographic core (limiting operations to ROI and sensitive bitplanes) lowers latency, energy consumption, and computational load, thereby improving throughput on resource-constrained platforms [34].

Table 12. Time Performance.

Image	Encryption Type	Size	Time (sec)	ET (MBps)	Cycles per Byte
Lena	Full encryption	512×512	3.0620	47.78	159.27
Lena	Selective encryption	512×512	1.6460	25.68	85.62
Barbara	Full encryption	512×512	3.3620	40.41	134.72
Barbara	Selective encryption	512×512	1.9460	23.39	77.98
School	Full encryption	512×512	3.9220	36.83	114.72
School	Selective encryption	512×512	2.0460	20.65	57.98

15. Conclusions

We investigated an edge-aware selective encryption approach for color images and evaluated it against the hypotheses introduced in Section 1. On a diverse set of 10 images, results support H1: encrypting MSBs within edge-derived ROI reduces computational cost relative to full encryption—by 35–50% on 512×512 images—while substantially degrading plaintext structure in protected regions. H2 is also supported: the selective scheme achieved low PSNR, elevated entropy, and reduced correlation compared with plaintext; when metrics are restricted to ROI

masks, values approach those of full encryption, which drove MSSIM below 0.1 with near-8-bit entropies across channels. Finally, H3 is supported by the cumulative effect of MSB XOR, channel bit swapping, intra-block shuffling, and CBC, which together improved statistical concealment and differential resistance (NPCR and UACI near ideal for full, and near-ideal within ROI for the selective variant). We additionally observed stability of indicators under JPEG compression and Gaussian noise, and favorable comparisons against AES-CBC and representative literature baselines [1, 4]. Relative to full encryption, the selective approach achieves comparable ROI security indicators (including near-ideal NPCR/UACI in ROI) at notably lower computational cost, whereas full encryption remains strongest for whole-image concealment.

The findings imply that selectively focusing cryptographic effort on visually informative regions can offer a practical trade-off between obfuscation and efficiency on resource-constrained platforms. Nevertheless, several limitations remain. First, the security evaluation emphasizes statistical and perceptual indicators; comprehensive cryptanalysis (e.g., under known/chosen-plaintext models) is outside our scope. Second, ROI selection based on Laplacian edges may overlook semantically important low-texture regions, and threshold choices affect both security and efficiency. Third, reliance on a general-purpose PRNG motivates future integration of a cryptographically secure PRNG and formal key/IV management.

Future work includes: (i) adaptive ROI strategies that combine edge, saliency, or learned predictors; (ii) extension to grayscale, high dynamic range, and video; (iii) hardware-oriented implementations (e.g., SIMD/GPU) and energy profiling; (iv) stronger adversarial evaluations and formal security models; and (v) dataset expansion and release of a reproducibility package with code and seeds to facilitate benchmarking.

Supplementary Materials

The supplementary materials package contains the MATLAB code used in this study and the image datasets employed in the experiments. The full codebase, scripts, and configuration files are publicly available at <https://github.com/IJacaman/MatLab>. All images are uploaded uncropped (original full-resolution versions).

Author Contributions

Conceptualization, I.J. and M.F.; Methodology, I.J.; Software, I.J.; Validation, I.J. and M.F.; Formal Analysis, I.J.; Investigation, I.J.; Resources, M.F.; Data Curation, I.J.; Writing—Original Draft Preparation, I.J.; Writing—Review & Editing, I.J. and M.F.; Visualization, I.J.; Supervision, M.F.; Project Administration, M.F. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Benchmark images used in this study are publicly available: standard test images [6] and the JuliaImages test image list [7]. Derived results are included in the tables/figures. Implementation scripts and configuration files are available at the project repository [26].

Conflict of Interest

The authors declare no conflict of interest.

References

- [1] Ma, K.; Teng, L.; Wang, X.; et al. Color image encryption scheme based on the combination of the fisher-yates scrambling algorithm and chaos theory. *Multimedia Tools Appl.* **2021**, *80*, 24737–24757.
- [2] Yavuz, E. A novel chaotic image encryption algorithm based on content-sensitive dynamic function switching scheme. *Opt. Laser Technol.* **2019**, *114*, 224–239.
- [3] Jacaman, I.; Farajallah, M. A Lightweight Spatial Domain Image Encryption Algorithms: A Review Paper. *J. Theor. Appl. Inf. Technol.* **2023**, *101*, 1275–1290.
- [4] Roy, S.; Shrivastava, M.; Pandey, C.V.; et al. IEVCA: An efficient image encryption technique for IoT applications using 2-D Von-Neumann cellular automata. *Multimedia Tools Appl.* **2021**, *80*, 31529–31567.
- [5] Zhang, L.; Wang, X.; Zhao, Y. Research on color image encryption algorithm based on bit-plane and chaotic system. *Entropy* **2022**, *24*, 186.
- [6] Al-Hazaimeh, A.S.; Al-Betar, M.A. Image encryption algorithms: A survey of design and evaluation metrics. *Digital* **2021**, *4*, 126–152.
- [7] Kiran, P.; Parameshchari, B.D. Resource Optimized Selective Image Encryption of Medical Images Using Multiple Chaotic Systems. *Microprocess. Microsyst.* **2022**, *91*, 104546. [CrossRef]
- [8] Wang, L.; Chen, Z.; Sun, X.; et al. Color image ROI encryption algorithm based on a novel 4D hyperchaotic system. *Phys. Scr.* **2023**, *99*, 015229. [CrossRef]
- [9] Wang, X.; Guan, N.; Liu, P. A selective image encryption algorithm based on a chaotic model using modular sine arithmetic. *Optik* **2022**, *258*, 168955. [CrossRef]
- [10] Barker, E.; Kelsey, J. *Recommendation for Random Number Generation Using Deterministic Random Bit Generators (Revised)*. NIST SP 800-90A Rev. 1; U.S. Department of Commerce: Gaithersburg, MD, USA, 2015. [CrossRef]
- [11] Nir, Y.; Langley, A. *ChaCha20 and Poly1305 for IETF Protocols*. RFC 8439; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2018. Available online: <https://datatracker.ietf.org/doc/rfc8439/>
- [12] Dworkin, M. *Recommendation for Block Cipher Modes of Operation: Methods and Techniques*. NIST SP 800-38A; U.S. Department of Commerce: Gaithersburg, MD, USA, 2001.
- [13] Kuhn, M. Image Processing Standard Test Images. Available online: <https://www.eecs.qmul.ac.uk/~phao/IP/Labs/cwk/images/> (accessed on 21 August 2023).
- [14] Massoudi, A.; Lefebvre, F.; De Vleeschouwer, C.; et al. Overview on selective encryption of image and video: challenges and perspectives. *Eur. J. Inf. Secur.* **2008**, *2008*, 179290.
- [15] Geetha, S.; Punithavathi, P.; Infanteena, A.M.; et al. A literature review on image encryption techniques. *Int. J. Inf. Secur. Privacy* **2014**, *12*, 42–83.
- [16] Kelsey, J. *Recommendation for the Entropy Sources Used for Random Bit Generation*. NIST SP 800-90B; U.S. Department of Commerce: Gaithersburg, MD, USA, 2018. [CrossRef]
- [17] Barker, E.; Kelsey, J. *Recommendation for Random Bit Generator (RBG) Constructions*. NIST SP 800-90C; U.S. Department of Commerce: Gaithersburg, MD, USA, 2016. [CrossRef]
- [18] Krawczyk, H.; Eronen, P. *HMAC-based Extract-and-Expand Key Derivation Function (HKDF)*. RFC 5869; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2010. [CrossRef]
- [19] Baker, E.; Chen, L.; Davis, R. *Recommendation for Key-Derivation Methods in Key-Establishment Schemes*. NIST SP 800-56C Rev. 2; U.S. Department of Commerce: Gaithersburg, MD, USA, 2020. [CrossRef]
- [20] Baker, E.; Chen, L.; Davis, R. *Recommendation for Cryptographic Key Generation*. NIST SP 800-133 Rev. 2; U.S. Department of Commerce: Gaithersburg, MD, USA, 2020. [CrossRef]
- [21] Wu, Y.; Noonan, J.P.; Agaian, S. NPCR and UACI randomness tests for image encryption. *Cyber J. Multidiscip. J. Sci. Technol.* **2011**, *4*, 31–38. Available online: <https://www.cyberjournals.com/Papers/Apr2011/05.pdf>
- [22] Berman, H.G. Random Number Generator - Stat Trek. Available online: <https://stattrek.com/statistics/random-number-generator#table/> (accessed on 16 September 2023).
- [23] NIST. *Security Requirements for Cryptographic Modules*. FIPS PUB 140-3; U.S. Department of Commerce: Gaithersburg, MD, USA, 2019. [CrossRef]
- [24] TestImages.jl. Image database, testimage. Available online: <https://testimages.juliaimages.org/stable/imagelist/> (accessed on 21 August 2023).
- [25] ISO/IEC 10116:2017. *Information technology — Security techniques — Modes of operation for an n-bit block cipher*; ISO/IEC: Geneva, Switzerland, 2017. Available online: <https://www.iso.org/standard/64575.html>
- [26] Jacaman, I. Selective Color Image Encryption Code (MATLAB) — GitHub repository. Available online: <https://github.com/IJacaman/MatLab> (accessed on 10 July 2025).
- [27] Ko, K.; Kim, S.; Kwon, H. Selective Audio Perturbations for Targeting Specific Phrases in Speech Recognition Systems. *Int. J. Comput. Intell. Syst.* **2025**, *18*, 103.
- [28] Lee, T.; Lee, S.; Kwon, H. Multi-Targeted Textual Backdoor Attack: Model-Specific Misrecognition via Trigger Position and Word Choice. *IEEE Access* **2025**, *13*, 57983–57993.
- [29] Ko, K.; Kim, S.; Kwon, H. Multi-targeted Audio Adversarial Example for Use against Speech Recognition Systems. *Comput. Secur.* **2023**, *128*, 103168. [CrossRef]

- [30] Kwon, H.; Nam, S.-H. Audio Adversarial Detection through Classification Score on Speech Recognition Systems. *Comput. Secur.* **2023**, *126*, 103061. [[CrossRef](#)]
- [31] Kwon, H.; Baek, J.-W. Targeted Discrepancy Attacks: Crafting Selective Adversarial Examples in Graph Neural Networks. *IEEE Access* **2025**, *13*, 13700–13712.
- [32] Kwon, H.; Lee, S. Detecting Textual Adversarial Examples through Text Modification on Text Classification Systems. *Appl. Intell.* **2023**, *53*, 19161–19185. [[CrossRef](#)]
- [33] Kwon, H.; Kim, D.-J. Dual-Targeted Adversarial Example in Evasion Attack on Graph Neural Networks. *Sci. Rep.* **2025**, *15*, 3912.
- [34] Hraini, I.; Farajallah, M.; Arman, N.; et al. Joint crypto-compression based on selective encryption for WMSNs. *IEEE Access* **2021**, *9*, 161269–161282.



Copyright © 2025 by the author(s). Published by UK Scientific Publishing Limited. This is an open access article under the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Publisher's Note: The views, opinions, and information presented in all publications are the sole responsibility of the respective authors and contributors, and do not necessarily reflect the views of UK Scientific Publishing Limited and/or its editors. UK Scientific Publishing Limited and/or its editors hereby disclaim any liability for any harm or damage to individuals or property arising from the implementation of ideas, methods, instructions, or products mentioned in the content.