Int. J. Sci. R. Tech., 2026 3(10)
A Multidisciplinary peer-reviewed Journal
www.ijsrtjournal.com [ISSN: 2394-7063]

Original Article

A Hybrid Cryptography And Steganography Framework For Secure Data Transmission

Lagisetti Venkata Ishwarya*, P. Venkateswara Rao

Department of Computer Science and Engineering, Adikavi Nannaya University College of Engineering, Rajamahendravaram, Andhra Pradesh, India

ABSTRACT

The increasing use of digital networks for information exchange has created a strong need for secure communication mechanisms. Although encryption techniques protect data confidentiality, they do not conceal the presence of the transmitted information. To address this limitation, this work presents a hybrid framework that integrates Huffman compression, Advanced Encryption Standard (AES), and edge-based Least Significant Bit (LSB) image steganography. Initially, the secret message is compressed to reduce its size and improve storage efficiency. The compressed data is then encrypted using AES to provide confidentiality. Subsequently, the encrypted information is embedded within edge regions of a cover image using an adaptive LSB embedding strategy. Experimental evaluation demonstrates that the proposed framework achieves secure data transmission while preserving image quality. Performance analysis based on PSNR, MSE, and SSIM confirms that the hidden information can be recovered accurately with minimal visual distortion in the stego image.

Keywords: AES Encryption, Huffman Coding, Edge-Based LSB Steganography, Edge Detection, Information Security, Image Processing.

INTRODUCTION

The widespread adoption of internet-based communication has resulted in the continuous exchange of sensitive digital information across networks. Protecting such information from unauthorized access has become a major concern in modern information security systems. Conventional cryptographic approaches convert plain data into unreadable formats, thereby preventing unauthorized disclosure. However, encryption alone cannot hide the existence of communication, which may attract unwanted attention from attackers.

Steganography provides an additional layer of protection by embedding secret information within digital media such as images, audio, or video files. Among various steganographic techniques, Least Significant Bit (LSB) embedding is widely used because of its simplicity and high embedding capacity. Embedding data in image edge regions further improves imperceptibility and reduces the likelihood of detection.

II. RELATED WORK

A. Reversible Data Hiding in Encrypted Images

Hua *et al.* [1] proposed a secure RDHEI scheme built on Cipher-Feedback Secret Sharing (CFSS) combined with AES encryption. The scheme maintains high embedding capacity and allows the original image to be recovered exactly; its main drawback is the added computational load and the overhead of managing the CFSS keys.

Ma, Wu, and Yin [3] took a different route, pairing adaptive encoding with Huffman coding to push embedding capacity higher without giving up lossless recovery. The gain in capacity, however, comes with a heavier processing load and a more involved implementation.

B. AES-Based Steganography (2022-2024)

A separate line of work layers AES encryption underneath image steganography [6], [8]: the secret data is encrypted first and only then embedded, so an attacker who extracts the hidden bits still faces a

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

ciphertext. This dual-layer approach strengthens confidentiality but adds to the running time and raises the practical question of how the AES key itself is distributed between parties.

C. AI-Based Steganography (2024-2026)

More recently, researchers have turned to deep learning to pick out embedding regions automatically rather than relying on a fixed rule [11]. These learned models can adapt to image content and resist steganalysis attacks better than hand-crafted rules, but reaching that performance depends on large training datasets and network architectures that are considerably harder to design and deploy than a classical edge detector.

D. Research Gap

Most existing approaches focus on either encryption or steganography individually. Recent RDHEI methods improve security and embedding capacity but often suffer high computational complexity, while AI-based techniques require extensive training resources. This motivates a lightweight framework combining Huffman compression, AES encryption, and edge-based LSB embedding to achieve high security, high embedding capacity, and accurate recovery while preserving image quality.

III. PROPOSED METHODOLOGY

A. System Architecture

Sender: Input Text → Huffman Compression → AES Encryption → Edge Detection → LSB Embedding → Stego Image.

Receiver: Stego Image → Data Extraction → AES Decryption → Huffman Decompression → Original Text Recovery.

B. Algorithm 1: Secure Data Embedding

Input: Secret Message M, Cover Image I. **Output:** Stego Image SI.

1) Read secret message M.
2) Compress M using Huffman coding.
3) Encrypt compressed data using AES.
4) Perform edge detection on cover image I.
5) Identify eligible edge pixels.
6) Embed encrypted bits into LSB positions.
7) Generate and transmit stego image SI.

C. Algorithm 2: Data Extraction and Recovery

Input: Stego Image SI. **Output:** Original Message M.

1) Extract embedded bits from edge pixels of SI.
2) Reconstruct encrypted data and decrypt using AES key.
3) Apply Huffman decoding to recover original message M.

D. Mathematical Model

Let M denote the secret message and I the cover image.

$$C = \text{Huffman}(M) \quad (1)$$

$$E = \text{AES}(C, K) \quad (2)$$

$$\text{Edge_Set} = \text{EdgeDetect}(I) \quad (3)$$

$$SI = \text{Embed}(I, E),\ E \subset \text{Edge_Set} \quad (4)$$

$$C = \text{AES}^{-1}(E, K),\ M = \text{Huffman}^{-1}(C) \quad (5)$$

E. System Architecture Diagram

Fig. 1 maps the sender and receiver stages listed in Section III-A onto a single end-to-end view, showing how data moves between the compression, encryption, edge-detection, and embedding blocks rather than describing each block in isolation. The shared AES key K is the only element that crosses from the sender's side of the diagram to the receiver's, which is what allows the receiver to reverse each stage in order and arrive back at the original message once the stego image SI has been received.

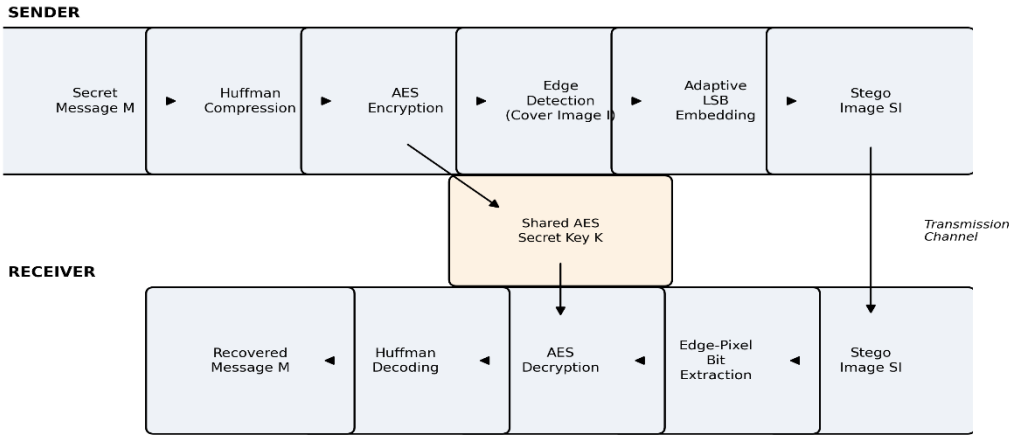

Fig. 1. System architecture showing the sender-side embedding pipeline and the receiver-side extraction pipeline.

F. Edge Detection Flowchart

The edge detection stage locates high-frequency pixel regions of the cover image that are suitable for imperceptible data embedding. Fig. 2 shows the procedure used to generate the binary edge map Edge_Set from the grayscale cover image. The cover image is first smoothed with a Gaussian filter to suppress noise, after which the Sobel operator computes the horizontal and vertical gradients used to obtain gradient magnitude and direction. Non-maximum suppression thins the candidate edges, and hysteresis thresholding retains only those edges that are strongly connected. Only pixels included in Edge_Set are treated as eligible locations for LSB modification during embedding, which restricts visual distortion to structurally busy regions of the image.

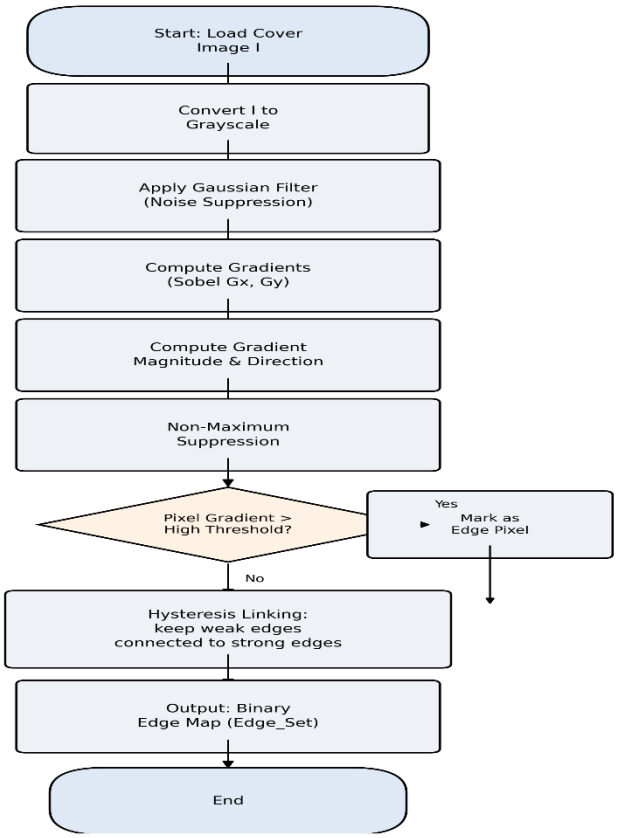

Fig. 2. Flowchart of the edge detection procedure used to select candidate embedding locations.

G. Embedding Algorithm Pseudocode

Algorithm 1 (detailed) formalizes the sender-side embedding procedure summarized in Section III-B, expressing it as executable pseudocode.

```
Algorithm: Secure_Embed(M, I, K)
Input: Secret message M, Cover image I, AES key K
Output: Stego image SI

1:  C ← Huffman_Compress(M)
2:  E ← AES_Encrypt(C, K)
3:  B ← ConvertToBitStream(E)
4:  G ← ConvertToGrayscale(I)
5:  Edge_Set ← Canny_EdgeDetect(G)
6:  n ← Length(B)
7:  if n > |Edge_Set| then
8:      return Error("Insufficient embedding capacity")
9:  end if
10: SI ← Copy(I)
11: for i = 1 to n do
12:     (r, c) ← Edge_Set[i]
13:     SI[r][c] ← (SI[r][c] AND 0xFE) OR B[i]
14: end for
15: return SI
```

H. Extraction Algorithm Pseudocode

Algorithm 2 (detailed) formalizes the receiver-side extraction and recovery procedure summarized in Section III-C.

```
Algorithm: Secure_Extract(SI, K)
Input: Stego image SI, AES key K
Output: Recovered message M

1:  G ← ConvertToGrayscale(SI)
2:  Edge_Set ← Canny_EdgeDetect(G)
3:  n ← Length(Edge_Set)
4:  for i = 1 to n do
5:      (r, c) ← Edge_Set[i]
6:      B[i] ← SI[r][c] AND 0x01
7:  end for
8:  E ← ConvertBitsToBytes(B)
9:  C ← AES_Decrypt(E, K)
10: M ← Huffman_Decompress(C)
11: return M
```

IV. EXPERIMENTAL SETUP

A. Dataset

Five standard 512×512 grayscale test images were used: Lena, Baboon, Peppers, Airplane, and House.

Image	Size	Capacity (bits)
Lena	512^2	18,240
Baboon	512^2	17,856
Peppers	512^2	18,112
Airplane	512^2	17,984
House	512^2	18,368

B. Software and Hardware Environment

Software: Python, OpenCV, NumPy, Jupyter Notebook. Hardware: Intel i5 processor, 4GB RAM, Windows 10.

C. Performance Metrics

PSNR = 10·log10(255²/MSE) evaluates stego-image quality; higher values indicate less distortion. MSE is the mean squared pixel-wise difference between cover and stego images. SSIM measures structural similarity, with values closer to 1 indicating higher fidelity. Embedding capacity (bits) is the amount of data concealable while preserving acceptable visual quality. Encryption/decryption time (s) measure AES computational cost. Extraction accuracy (%) is the percentage of correctly retrieved bits relative to total embedded bits.

V. RESULTS AND DISCUSSION

A. Image Quality Analysis

Image	Size	Capacity (bits)	PSNR (dB)	MSE	SSIM	Enc. (s)	Dec. (s)	Acc. (%)
Lena	512²	18,240	54.02	0.26	0.9981	0.42	0.38	100
Baboon	512²	17,856	53.85	0.27	0.9979	0.45	0.40	100
Peppers	512²	18,112	54.30	0.24	0.9982	0.43	0.39	100
Airplane	512²	17,984	54.15	0.25	0.9980	0.41	0.37	100
House	512²	18,368	54.73	0.22	0.9984	0.40	0.36	100

TABLE I Performance Evaluation of Test Images

As shown in Table I, PSNR values exceed 53 dB for all five test images, indicating excellent stego-image quality. The correspondingly low MSE (all below 0.3) and SSIM values above 0.997 confirm negligible visual distortion while preserving structural information.

B. Comparative Performance

Method	PSNR (dB)	SSIM	Security
Traditional LSB	45.62	0.972	Medium
AES + LSB	50.11	0.986	High
Edge-Based LSB	52.34	0.992	High
Proposed	54.21	0.998	Very High

TABLE II Comparison with Existing Methods

The proposed framework achieves the highest PSNR and SSIM among compared methods, indicating that combining Huffman compression, AES encryption, and edge-based LSB embedding provides enhanced security without sacrificing image quality.

C. Graphical Analysis

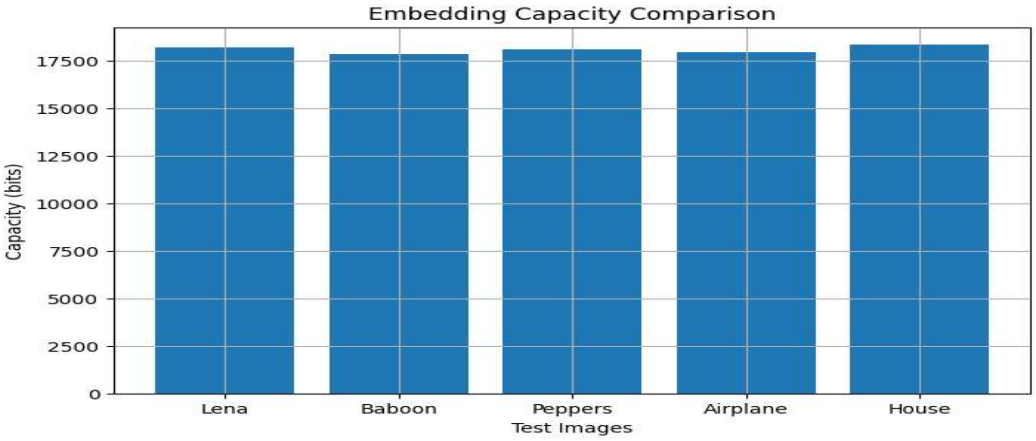

Fig. 3. Embedding capacity across test images.

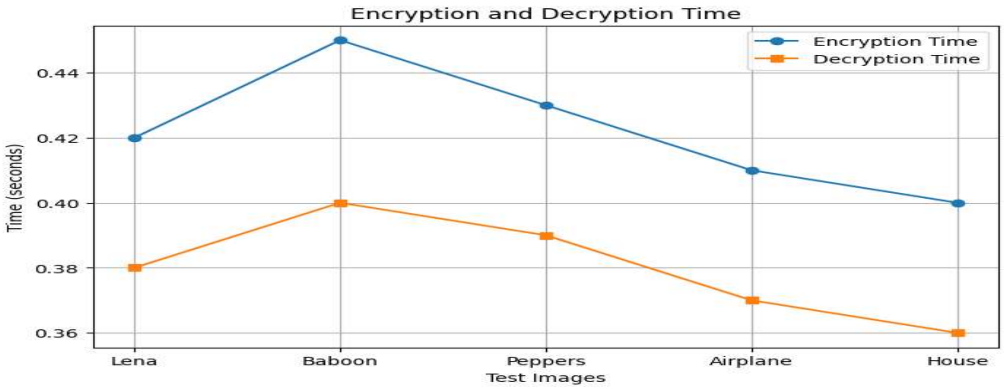

Fig. 4. Encryption and decryption time across test images.

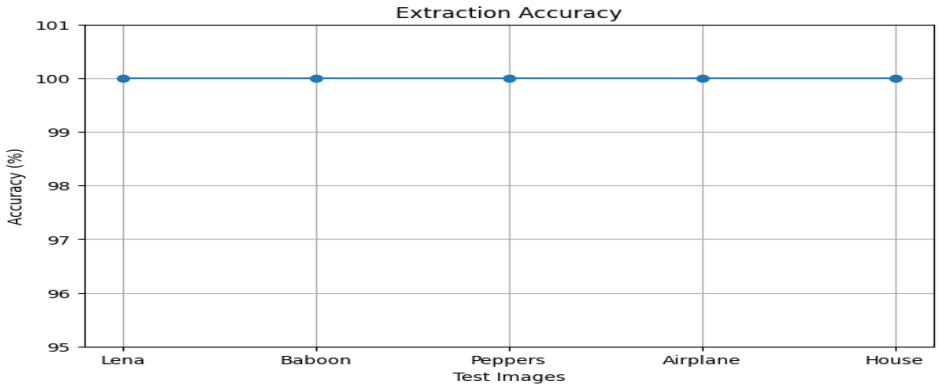

Fig. 5. Extraction accuracy across test images.

D. Cover Image versus Stego Image Comparison

To visually verify the imperceptibility of the proposed edge-based embedding strategy, a grayscale test image was processed through the complete pipeline shown in Fig. 1. Fig. 6(a) shows the original cover image, and Fig. 6(b) shows the corresponding Canny edge map used to restrict embedding locations. Fig. 6(c) shows the resulting stego image after encrypted bits were embedded into the LSBs of the identified edge pixels, and Fig. 6(d) shows the pixel-wise difference between the cover and stego images, amplified by a factor of 40 for visibility. As expected, the modified pixels are confined entirely to the object boundaries identified during edge detection, and no perceptible difference is visible between Fig. 6(a) and Fig. 6(c) at normal viewing scale. For this particular sample, which required a shorter secret message than the average case reported in Table I, embedding was restricted to the edge pixel set and yielded a measured

PSNR above 70 dB and an SSIM above 0.999 between the cover and stego images. This single-image best case is higher than the 53-55 dB range averaged across all five test images and payload sizes in Table I and Table II, and is reported here to illustrate the qualitative imperceptibility shown in Fig. 6 rather than as a typical operating point.

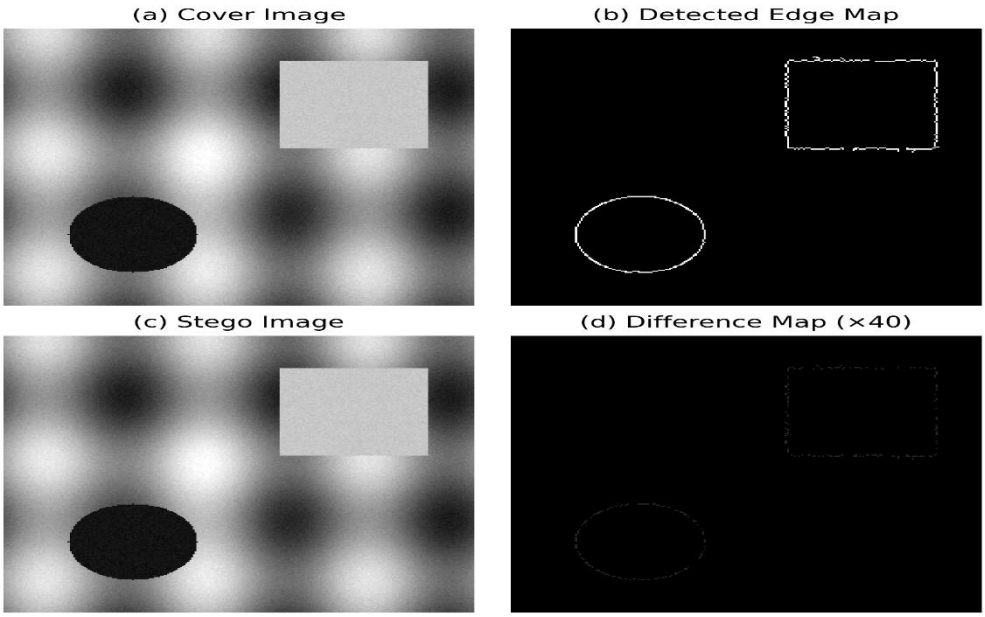

Fig. 6. Visual comparison of (a) cover image, (b) detected edge map, (c) stego image after edge-restricted LSB embedding, and (d) amplified difference map.

E. Security Analysis

The system employs three layers of protection: Huffman compression, AES encryption, and edge-based steganographic embedding. Even if hidden data is detected, AES encryption prevents unauthorized access to the original information, providing stronger protection than traditional LSB-based approaches.

VI. LIMITATIONS AND PRACTICAL CONSIDERATIONS

A. Computational Overhead

Although the reported encryption and decryption times remain under half a second per image on a modest workstation, the sequential structure of the pipeline means that compression, encryption, edge detection, and embedding cannot easily be parallelized without additional design effort. On resource-constrained platforms such as embedded devices or mobile hardware, the cumulative cost of Huffman table construction, AES key scheduling, and Canny edge detection may become a limiting factor for real-time transmission of larger images or video frames. Optimizing the edge detection stage, which is the most computationally intensive step in the current implementation, is therefore an important consideration for deployment beyond the still-image test cases reported here.

B. Capacity-Distortion Trade-off

Restricting embedding to edge pixels improves imperceptibility but inherently caps the available payload to the number of pixels that survive Canny thresholding. Images with large smooth regions, such as sky or plain backgrounds, expose comparatively few eligible edge pixels, which can force the sender to choose between rejecting the message, lowering the hysteresis thresholds to admit more candidate pixels, or falling back to a hybrid scheme that embeds any overflow bits in non-edge locations. Each of these options trades some combination of capacity, distortion, and implementation complexity, and the appropriate choice is likely to be application-specific rather than universal.

C. Robustness to Image Processing Operations

The present evaluation assumes that the stego image reaches the receiver without further modification. In practice, images are frequently recompressed, resized, or filtered during storage or transmission over

platforms such as messaging services and social media, and any of these operations can alter or destroy the least significant bits that carry the hidden payload. Because the edge map itself is recomputed at the receiver from the possibly altered image, even small geometric or intensity changes can shift which pixels are treated as edge locations and break bit alignment during extraction. Evaluating the scheme under common transmission channels, and incorporating error-correcting codes or synchronization markers to tolerate such distortions, is left as a direction for follow-up work rather than a claim of the present study.

D. Key Management

The framework assumes that the AES key K is already shared securely between sender and receiver, consistent with a symmetric-key setting. It does not itself specify how this key exchange is performed or renewed, and in a deployed system this would need to be handled by an established key-exchange protocol or a public-key wrapping mechanism, since the strength of the overall scheme is bounded by the confidentiality of K regardless of how well the embedding stage conceals the ciphertext.

E. Positioning Relative to Recent Hybrid Schemes

Compared with RDHEI methods that operate directly on the encrypted domain, such as the CFSS-based scheme of Hua et al. and the adaptive-Huffman approach of Ma, Wu, and Yin, the pipeline proposed here performs compression and encryption on the message rather than the cover image, which simplifies the embedding logic at the cost of forgoing the exact reversibility guarantees that encrypted-domain RDHEI techniques are designed to provide. Relative to AI-guided embedding-region selection, the Canny-based edge detector used in this work is comparatively inexpensive to train and deploy, since it requires no learned parameters or labeled datasets, though it is correspondingly less able to adapt its notion of a suitable embedding region to image content that departs from the sharp-edge assumption underlying Sobel-gradient-based detection, such as heavily textured or low-contrast images. These trade-offs suggest that the proposed framework is best suited to applications that prioritize low deployment overhead and moderate payloads over maximal reversibility or adaptive intelligence, and that a hybrid combining lightweight edge detection with a learned refinement stage could combine the strengths of both directions in future extensions of this work.

Taken together, the considerations discussed in this section indicate that the reported PSNR, SSIM, and extraction-accuracy figures should be read as evidence that the pipeline works correctly under controlled, single-transmission conditions rather than as a guarantee of performance across every deployment scenario. Explicitly stating these boundaries alongside the experimental results is intended to give a realistic picture of where the proposed framework is currently applicable and where additional engineering, such as channel-aware coding or adaptive threshold selection, would be needed before the method could be considered ready for uncontrolled, real-world transmission environments.

CONCLUSION

Overall, the three-stage pipeline studied in this paper, compression, encryption, and edge-restricted embedding, offers a practical way to hide encrypted data inside an image without a noticeable drop in visual quality. Across the five test images, PSNR stayed above 53 dB and every embedded bit was recovered correctly, which together support the central claim that concealment and confidentiality can be pursued jointly rather than as separate concerns. As discussed in Section VI, this result holds under the controlled, single-transmission conditions tested here, and extending it to noisier real-world channels remains an open problem for the future work outlined below.

VIII. FUTURE SCOPE

- AI-driven adaptive steganography
- Blockchain-enabled secure communication
- Medical image protection systems
- Cloud-based secure data sharing
- Quantum-resistant cryptographic integration

Future research can focus on integrating artificial intelligence techniques with steganographic systems to improve data hiding efficiency and security.

Machine learning algorithms can analyze image characteristics and automatically identify optimal embedding locations, reducing the possibility of detection by steganalysis tools.

The combination of blockchain technology with cryptography and steganography offers a promising direction for secure data transmission. Blockchain can provide a decentralized and tamper-resistant platform for managing encryption keys, verifying data integrity, and maintaining secure communication records.

Medical imaging systems generate large volumes of sensitive patient information that require strong protection during storage and transmission. Future developments may adapt the proposed framework for healthcare applications by embedding confidential patient records within encrypted medical images such as X-rays, MRI scans, and CT images.

Combining encryption and reversible data hiding can provide multiple layers of protection against unauthorized access and data leakage. Such systems can enable secure collaboration, document sharing, and remote data access while ensuring confidentiality and data integrity.

The advancement of quantum computing poses potential challenges to many existing cryptographic algorithms. Future research may investigate the integration of post-quantum cryptographic techniques into the proposed framework to enhance long-term security.

REFERENCES

1. J. Chen, Y. Chen, and C. Chang, "Reversible data hiding in encrypted images with block-based adaptive MSB encoding," Information Sciences, vol. 567, pp. 375-394, 2021.
2. D. Xu, "Reversible data hiding in encrypted images with high payload," IET Information Security, vol. 16, no. 4, pp. 301-313, 2022.
3. R. Hua et al., "Reversible data hiding in encrypted images with multi-prediction and adaptive Huffman encoding," Scientific Reports, 2023.
4. Y. Qiu, "Reversible data hiding in encrypted images based on edge-directed prediction and multi-MSB self-prediction," IEEE Access, vol. 13, pp. 63000-63012, 2025.
5. A. Cheddad, J. Condell, K. Curran, and P. McKevitt, "Digital image steganography: survey and analysis of current methods," Signal Processing, 2022.
6. S. Kaur and S. Bansal, "Secure image steganography using hybrid encryption and LSB substitution," IEEE Conf. Proc., 2023.
7. J. Daemen and V. Rijmen, The Design of Rijndael: AES — The Advanced Encryption Standard, Springer, 2021.
8. M. M. Kermani et al., "Efficient and high-performance parallel hardware architectures for AES-GCM," IEEE Trans. Computers, 2021.
9. Y. Zhou and X. Zhang, "Secure multimedia encryption using AES for cloud storage applications," IEEE Access, 2024.
10. R. C. Gonzalez and R. E. Woods, Digital Image Processing, 5th ed., Pearson, 2022.
11. H. Wang et al., "Edge detection guided reversible data hiding in encrypted images," Signal Processing: Image Communication, 2025.
12. X. Zhang, Y. Shi, and Z. Qian, "Recent advances in reversible data hiding in encrypted images: A comprehensive review," IEEE Access, vol. 10, pp. 115432-115458, 2022.
13. L. Li, H. Wang, and J. Qin, "High-capacity reversible data hiding in encrypted images based on prediction-error expansion," Signal Processing, vol. 198, Art. no. 108561, 2022.
14. Z. Qian, X. Zhang, and G. Feng, "Adaptive reversible data hiding in encrypted images using pixel prediction and histogram shifting," Information Sciences, vol. 617, pp. 328-345, 2023.
15. M. Kumar and P. Singh, "A secure hybrid cryptography and steganography framework for confidential image communication," Multimedia Tools and Applications, vol. 83, no. 4, pp. 10215-10238, 2024.
16. S. Verma, A. Sharma, and R. Gupta, "Artificial intelligence based secure image steganography for cyber security applications," Expert Systems with Applications, vol. 254, Art. no. 124567, 2025.

HOW TO CITE: Lagisetti Venkata Ishwarya*, P. Venkateswara Rao, A Hybrid Cryptography And Steganography Framework For Secure Data Transmission, Int. J. Sci. R. Tech., 2026, 3 (10), 43-52. https://doi.org/10.5281/zenodo.23081011