Steganography Using Randomized LSB and AI Encryption

Mrs.V. Anupama[1], Polavarapu Bhavya Keerthi[2], Yadla Bhavani[3],
Seelam Bhaskar Sriharsha[4], Nagiredla Suresh Kumar[5]
[1]*Associate Professor, Department of Computer Science and Information technology, Lendi Institute of Engineering and Technology (A), Vizianagaram, India*
[2,3,4,5]*Department of Computer Science and Information technology, Lendi Institute of Engineering and Technology (A), Vizianagaram, India*

***Abstract*— Secure data communication has become a critical requirement in modern digital environments due to the increasing risks of data interception and information leakage over open networks. The proposed project focuses on developing a randomized LSB-based image steganography system with AI-assisted hybrid encryption to provide a secure and covert communication platform.**

The system employs AES-256 encryption followed by the ChaCha20 stream cipher to ensure strong data confidentiality and enhanced security. An AI-based image analysis mechanism using a Random Forest Regressor supports intelligent pixel selection for data embedding by analyzing image features such as pixel intensity, edge distribution, and texture patterns, improving adaptability across different images.

The encrypted data is embedded into digital images using an AI-guided randomized Least Significant Bit (LSB) steganography technique, reducing detectability and improving resistance to steganalysis attacks while maintaining image quality. The system is implemented as a full-stack web application using React and Express, incorporating user authentication and activity monitoring features.

The proposed system enables secure and reliable information sharing without significantly affecting system performance.

***Index Terms*— Image Steganography, Randomized LSB, AI-Assisted Pixel Selection, Random Forest Regressor, Hybrid Encryption, AES-256, ChaCha20, Secure Communication, Web-Based Steganography.**

I. INTRODUCTION

In the modern digital era, the exchange of confidential information over open networks has increased significantly. With the rapid growth of internet-based communication, sensitive data such as personal information, financial records, and organizational documents are frequently transmitted across unsecured channels. Although cryptographic techniques protect data by converting it into unreadable formats, they do not conceal the existence of communication. The visibility of encrypted data can attract attackers and increase the risk of targeted cyber threats.

Steganography addresses this limitation by hiding secret information within digital media such as images, audio, or video files. Among various steganographic techniques, image-based steganography using the Least Significant Bit (LSB) method is widely adopted due to its simplicity and high embedding capacity. However, traditional LSB techniques follow predictable embedding patterns, making them vulnerable to statistical steganalysis attacks.

To overcome these limitations, modern steganography systems incorporate additional security mechanisms such as randomization and multi-layer encryption. Randomized embedding distributes hidden data across different pixel locations, reducing predictable patterns and improving resistance to detection. Hybrid encryption techniques further enhance security by protecting the message before embedding, ensuring that even if hidden data is extracted, it remains unreadable without proper decryption keys.

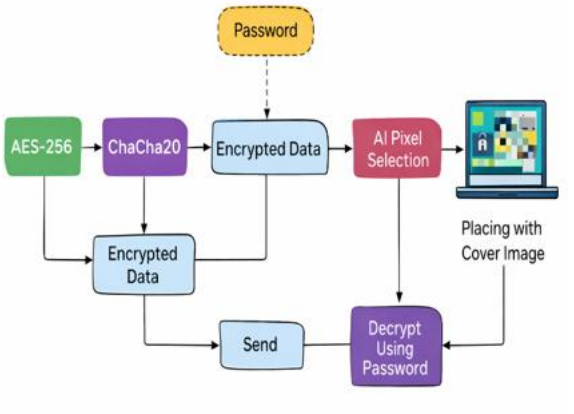
Furthermore, the integration of artificial intelligence within the embedding process enhances adaptability across diverse image types and resolutions. In the proposed system, a Random Forest Regressor is used as part of the AI-based image analysis to evaluate

characteristics such as texture intensity, edge distribution, and pixel variation. Based on these features, the model identifies optimal pixel regions for embedding where modifications are less noticeable to the human visual system.

By combining intelligent pixel selection with randomized embedding, the system minimizes distortion and reduces predictable patterns, thereby improving resistance against statistical detection techniques and modern steganalysis tools.

Thus, the proposed system integrates hybrid encryption, AI-assisted randomized LSB embedding, and secure web implementation to provide enhanced confidentiality and reduced detectability. The following sections present the system design, methodology, and performance evaluation.

The overall working of the proposed system is illustrated in the diagram, where the secret message is first encrypted using AES-256 and ChaCha20 algorithms with the help of a user-defined password. The encrypted data is then processed using an AI-based pixel selection mechanism, where a Random Forest Regressor identifies suitable pixel regions for embedding. The data is embedded into the cover image using randomized LSB technique to generate a stego image. At the receiver side, the hidden data is extracted and decrypted using the same password to recover the original message.



II. RELATED WORK, MOTIVATION AND PROBLEM IDENTIFICATION

2.1 Related Work

and F. A. P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, 2016.

Focus: Explains advanced information hiding approaches and emphasizes resistance against steganalysis.

[6] A. Cheddad et al., "Digital Image Steganography: Survey and Analysis," Signal Processing, 2010.

Focus: Analyzes strengths and weaknesses of traditional LSB techniques and identifies statistical vulnerabilities.

[7] W. Stallings, Cryptography and Network Security, Pearson, 2017.

Focus: Describes cryptographic standards such as AES-256 for secure data encryption.

[8] D. J. Bernstein, "ChaCha20 and Poly1305 for IETF Protocols," RFC 8439, 2018.

Focus: Introduces the ChaCha20 stream cipher, known for high-speed encryption and strong resistance to cryptographic attacks.

[9] A. Singh, "Hybrid Encryption Techniques for Secure Communication," International Journal of Computer Applications, 2021.

Focus: Highlights benefit of combining multiple encryption algorithms to improve layered security.

[10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.

Focus: Explains machine learning principles that support AI-based analysis and intelligent data processing.

[11] Adaptive Randomized LSB Steganography – Various Research Studies

Focus: Introduces randomized embedding strategies to reduce predictability and resist statistical detection.

[12] Transform Domain Steganography using DCT/DWT – Research Surveys.

Focus: Embeds data within frequency components to improve robustness against compression and manipulation.

[13] Statistical Steganalysis Techniques – Research Articles.

Focus: Examines detection methods used to identify hidden data in digital images.

[14] Secure Web-Based Communication Systems – Modern Web Security Research.

Focus: Discusses authentication mechanisms, secure session management, and activity monitoring in web applications.

[15] AI-Assisted Pixel Selection for Image Steganography – Recent Research Trends.

Focus: Utilizes machine learning techniques to select optimal embedding regions for improved imperceptibility and security.

2.2 Motivation

In today's interconnected digital world, secure communication has become increasingly important due to the rising threats of cyberattacks, data breaches, and unauthorized access. Sensitive information such as personal credentials, financial transactions, confidential documents, and organizational data is frequently transmitted over open networks. Although encryption techniques provide strong confidentiality, the presence of encrypted data itself may raise suspicion and attract malicious attention.

Traditional steganography methods, particularly simple LSB substitution techniques, offer basic data hiding capabilities but suffer from several limitations. Sequential embedding patterns make hidden data vulnerable to statistical steganalysis attacks. Additionally, many conventional systems lack layered encryption mechanisms and intelligent embedding strategies, reducing overall robustness. The absence of structured authentication and activity monitoring further limits their practical deployment in real-world secure communication platforms.

With advancements in artificial intelligence and modern web technologies, it is now possible to develop more secure, intelligent, and user-controlled steganographic systems. AI-assisted pixel selection can enhance imperceptibility, while hybrid encryption combining AES-256 and ChaCha20 can provide multi-layered protection. Furthermore, integrating steganography into a full-stack web application with authentication and logging features can improve usability, access control, and traceability.

These factors motivate the development of the proposed system, which aims to create a secure, scalable, and practical covert communication framework by combining encryption, randomized embedding, artificial intelligence, and structured web implementation.

2.3 Problem Identification

Despite significant research in steganography and cryptography, several challenges remain in building a secure and practical covert communication system.

First, traditional LSB-based steganography techniques follow predictable embedding patterns, making them vulnerable to statistical steganalysis and detection methods. This reduces their reliability in high-security environments.

Second, standalone steganography tools often rely on single-layer encryption or no encryption at all. If hidden data is extracted, it may be easily decrypted or exposed without additional protection. There is a need for multi-layer encryption mechanisms that provide enhanced confidentiality even if extraction occurs.

Third, many existing approaches lack intelligent pixel selection mechanisms. Random or sequential embedding without image analysis may introduce detectable visual artifacts, especially when embedding larger payloads.

Fourth, most conventional systems do not provide structured authentication, access control, or activity monitoring. This limits their usability in controlled communication environments where traceability and user accountability are important.

Finally, there is limited integration between advanced encryption techniques, AI-based embedding strategies, and full-stack web-based implementation within a unified platform.

These challenges highlight the necessity for a secure, intelligent, and web-integrated steganographic framework that ensures confidentiality, low detectability, user authentication, and operational transparency objectives addressed by the proposed system.

III. THEORETICAL FRAMEWORK

3.1 Overview

The theoretical framework of the proposed system is grounded in the principles of cryptography, digital image steganography, artificial intelligence–assisted image analysis, and secure web-based communication. Instead of relying solely on traditional encryption methods that expose the presence of protected data, the framework treats secure communication as a dual-layer process in which information is first encrypted and then concealed within digital images. This approach aligns with modern security research that emphasizes combining cryptographic strength with covert data transmission to enhance confidentiality and reduce detectability.

At the core of the framework is the separation of responsibilities between encryption, intelligent embedding, and secure transmission. The encryption

stage transforms the secret message into unreadable ciphertext using hybrid encryption techniques. The embedding stage hides the encrypted data within carefully selected image pixels using AI-guided randomized LSB substitution. Finally, the transmission and extraction stage ensures secure delivery and recovery of the hidden message through password-based decryption. This modular separation improves system reliability, enhances security, and enables independent optimization of each stage.

The framework adopts a hybrid encryption model that combines AES-256 block cipher and ChaCha20 stream cipher to provide layered confidentiality. AES-256 ensures strong resistance against brute-force attacks, while ChaCha20 introduces additional randomness and high-speed encryption. This dual-layer approach significantly reduces the risk of data compromise even if extraction occurs.

Artificial intelligence plays a crucial role in improving imperceptibility. Instead of embedding data sequentially, the system analyses image characteristics such as texture distribution, edge intensity, and noise variation to identify optimal embedding regions. By limiting data insertion to perceptually stable areas, the system preserves image quality and minimizes statistical anomalies.

The framework further integrates a secure web-based implementation that incorporates authentication, controlled access, and activity logging. Unlike standalone steganography tools, this structured design ensures traceability, accountability, and practical deploy ability in real-world digital communication environments.

Overall, the theoretical framework establishes a scalable and deployable secure communication model that prioritizes confidentiality, imperceptibility, and controlled access over simplistic data hiding techniques.

3.2 System Architecture

Figure 1 illustrates the high-level architecture of the proposed steganographic framework, which consists of four primary components:

• Hybrid Encryption Layer
• AI-Assisted Embedding Layer
• Stego Image Transmission Layer
• Extraction and Decryption Layer

The communication process begins with the hybrid encryption layer, where the user-provided secret message is encrypted using AES-256 and further secured using ChaCha20 encryption. The encrypted output is converted into binary form for embedding.

The encrypted binary data is then passed to the AI-assisted embedding layer. In this stage, the cover image is analyzed to identify optimal pixel locations for embedding. A randomized Least Significant Bit (LSB) substitution method is applied to insert encrypted bits into selected pixels without introducing visible distortion.

Once embedding is completed, the modified image becomes the stego image. This image visually appears unchanged but contains concealed encrypted data. The stego image is transmitted securely through the web-based platform.

At the receiver side, the extraction and decryption layer retrieves hidden bits using the same randomized embedding logic. The extracted ciphertext is decrypted using ChaCha20 followed by AES-256 to recover the original message.

This sequential pipeline ensures secure encryption, intelligent embedding, controlled transmission, and accurate recovery of hidden information.

3.3 Computer Vision and Deep Learning Foundations

Cryptography and steganography serve complementary roles in secure communication systems. Cryptography protects the content of a message by converting it into ciphertext using mathematical algorithms and secret keys. Steganography, on the other hand, conceals the existence of the message by embedding it within digital media.

AES-256 is a symmetric block cipher that operates on fixed-size data blocks using substitution-permutation networks. Its 256-bit key length provides strong resistance against brute-force attacks and ensures high-level confidentiality. However, encrypted data alone may attract suspicion if intercepted.

ChaCha20 is a modern stream cipher that generates a pseudo-random keystream combined with plaintext to produce ciphertext. Its computational efficiency and resistance to timing attacks make it suitable for high-speed encryption applications. When combined with AES-256, it forms a hybrid encryption model that enhances layered security.

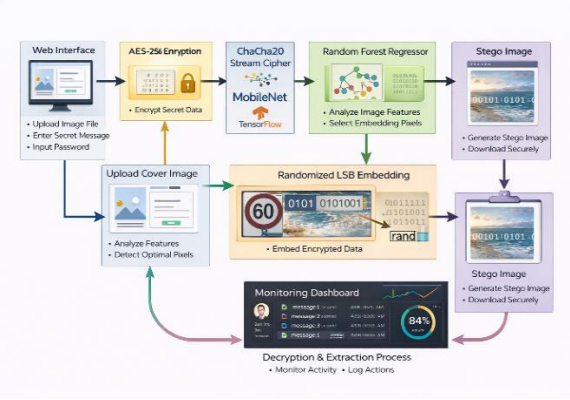
Least Significant Bit (LSB) steganography modifies the lowest bit of pixel values to embed secret information. Since LSB changes produce minimal

visual difference, image quality is preserved. However, traditional sequential LSB methods are vulnerable to detection through statistical analysis.

To overcome this limitation, the proposed system employs AI-assisted randomized LSB embedding. By analyzing image features and selecting embedding locations intelligently, the system reduces embedding predictability and improves resistance to steganalysis techniques.

From a theoretical perspective, this framework shifts the focus from simple data hiding to intelligent, layered security integration. By combining cryptographic strength, randomized embedding, and AI-based image analysis, the system achieves a practical balance between confidentiality, imperceptibility, and computational efficiency.

This theoretical foundation justifies the adoption of hybrid encryption, AI-assisted embedding, and web-based secure communication as an effective and scalable solution for modern covert digital information exchange.



IV. SYSTEM DESIGN AND ARCHITECTURE

4.1 Architectural Overview

The proposed steganography system adopts a layered full-stack web architecture designed to balance security, usability, and computational efficiency. Instead of functioning as a standalone offline tool, the system operates as a structured web-based communication platform where users can securely embed and extract hidden messages. This architecture ensures controlled access, layered encryption, and traceable communication.

The system is implemented using a React-based frontend for user interaction and an Express.js backend for processing encryption, embedding, and extraction operations. Each architectural layer is assigned a distinct responsibility: user authentication, encryption processing, intelligent embedding, and message recovery. This separation enhances modularity, scalability, and maintainability.

At the top layer, users interact with the system through a secure login interface. The processing layer handles hybrid encryption (AES-256 and ChaCha20), AI-assisted pixel selection, and randomized LSB embedding. The output layer generates the stego image and manages secure extraction and decryption at the receiver side. This layered separation improves system organization and enables independent optimization of security and performance components.

4.2 Software Component Design

The system combines both frontend and backend software modules to achieve secure data embedding and extraction through image steganography.

Software Components

- React.js for frontend user interface
- Node.js / Express.js for backend server
- Python for steganography processing
- OpenCV for image processing
- NumPy for pixel-level operations
- Encryption is done by AI randomised LSB

The React.js frontend provides an interactive interface for users to upload images, enter secret messages, and perform embedding or extraction operations. The Node.js/Express backend manages user requests, authentication, and communication between the frontend and processing modules.

Python is used to implement the core steganography logic, including encryption, AI-assisted pixel selection using Random Forest Regressor, and randomized LSB embedding. OpenCV handles image preprocessing and manipulation, while NumPy supports efficient matrix and pixel operations. Cryptographic libraries are used to perform secure encryption and decryption of data before embedding and after extraction.

4.3 Secure Communication Processing Pipeline

The system follows a structured sequential pipeline for secure message transmission. Initially, the user provides a secret message and password. The message is encrypted using AES-256 and further secured with ChaCha20 to ensure dual-layer protection.

The encrypted data is then embedded into a cover image using AI-assisted randomized LSB substitution. A Random Forest Regressor is used to analyze image

features such as pixel intensity, edge distribution, and texture patterns to identify suitable pixel locations for embedding. This intelligent pixel selection ensures that data is embedded in regions where changes are less noticeable.

This process generates a stego image that visually appears unchanged but contains hidden encrypted information.

At the receiver side, the embedded data is extracted from the stego image by identifying the selected pixel locations using the same Random Forest Regressor approach, and then decrypted using the same password. This sequential pipeline ensures confidentiality, imperceptibility, and accurate recovery of the original message.

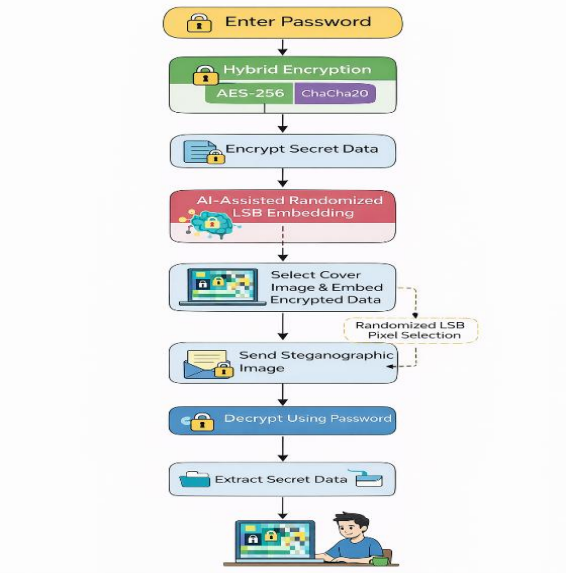
4.4 Event-Driven Processing Flow

The system follows an event-driven model to ensure efficient and responsive operation.

The process begins when a user uploads a cover image and submits a secret message. The encryption module is triggered automatically, followed by the embedding module. Once embedding is completed, a stego image generation event is triggered, allowing the user to download or transmit the image.

Similarly, during the extraction process, the decryption module is activated only when a stego image and correct password are provided. If the password is incorrect, the system prevents message recovery.

This event-driven design reduces unnecessary processing, enhances security validation, and ensures that operations occur only when valid user actions are performed.



4.5 User Interface Design

The web interface serves as the human–system interaction layer of the application, providing secure, efficient, and user-friendly access to steganography operations without requiring command-line interaction. It enables users to perform embedding and extraction processes seamlessly through an interactive environment.

The interface includes:

• Registration and Login Page for secure user authentication

• Dashboard with Send, Receive, and Activity History options for easy navigation

• Image Upload and Message Input Forms for embedding secret data

• Password Input Fields for encryption and decryption to ensure data security

• Activity Log Display for tracking user operations and system events

The interface is designed to ensure smooth interaction between the user and backend processing modules. It allows users to upload images, enter secret messages, and trigger AI-assisted embedding using the Random Forest Regressor along with hybrid encryption. The responsive design and event-driven operations improve usability and system efficiency.

4.6 Design Rationale and Trade-Offs

The system design prioritizes layered security and controlled access over minimalistic implementation. While simpler LSB tools may offer faster embedding, they lack multi-layer encryption and authentication features. The hybrid encryption approach increases computational overhead slightly but significantly enhances confidentiality.

The use of AI-assisted pixel selection improves imperceptibility but requires additional image analysis processing. This trade-off is justified as it strengthens resistance against steganalysis attacks.

The web-based architecture improves accessibility and traceability but introduces server-side processing requirements. However, running the system locally ensures privacy and eliminates dependency on external networks.

The modular layered architecture allows independent upgrades of encryption algorithms, embedding techniques, or UI components without affecting the

overall system structure. This scalability ensures adaptability to future security enhancements.

V. IMPLEMENTATION AND METHODOLOGY

5.1 Implementation Environment

The proposed steganography system is implemented as a full-stack web-based secure communication platform. The system is developed using React.js for the frontend interface and Express.js with Node.js for backend processing. The application runs locally on a personal computer using a localhost server environment.

The frontend handles user authentication, message input, image upload, and result visualization. The backend performs hybrid encryption using AES-256 and ChaCha20, AI-assisted pixel selection, randomized LSB embedding, extraction, and decryption.

All cryptographic and embedding computations occur locally without relying on external cloud services. This ensures data privacy, low latency, and secure offline operation.

5.2 Methodology Overview

The system follows a structured stepwise methodology that separates encryption, embedding, transmission, extraction, and decryption phases. Unlike traditional steganography tools, the proposed system integrates hybrid encryption and AI-based intelligent pixel selection within a web-based platform.

The methodology consists of five primary phases:

- User Authentication
- Hybrid Encryption (AES-256 + ChaCha20)
- AI-Assisted Randomized LSB Embedding
- Stego Image Generation and Transmission
- Extraction and Password-Based Decryption

This modular design enhances security, imperceptibility, and controlled communication.

5.3 Encryption and Embedding Process

In the first phase, the user enters a secret message and password. The password is used to generate cryptographic keys.

The system first encrypts the message using AES-256 block cipher. The encrypted output is then processed using ChaCha20 stream cipher to introduce additional randomness and strengthen confidentiality.

The encrypted binary data is passed to the AI-based image analysis module. The system analyzes image texture and edge density to select optimal embedding regions. Randomized Least Significant Bit substitution is applied to embed encrypted bits into selected pixels.

The output of this phase is a stego image that visually appears identical to the original cover image but contains hidden encrypted information.

5.4 Extraction and Decryption Process

At the receiver side, the stego image is uploaded to the system. The encrypted bits are extracted using the same randomized LSB logic.

The extracted ciphertext is decrypted using ChaCha20 followed by AES-256 using the user-provided password. If the password is correct, the original message is restored. If incorrect, decryption fails, ensuring access control.

This dual-layer decryption mechanism ensures secure recovery of hidden data.

5.5 Event-Driven Operation

The system operates using an event-driven mechanism. Encryption and embedding are triggered only when the user submits valid input. Similarly, decryption occurs only when a stego image and correct password are provided.

Activity history logs record operations such as login, encryption, and decryption, improving traceability and system transparency.

5.6 Algorithmic Workflow

Algorithm: Traffic Sign Capture, Recognition, and Display

Input: Secret message, Cover image, Password

Output: Stego image / Extracted original message

1. Initialize system and authenticate user
2. Accept secret message and password
3. Generate encryption keys from password
4. Encrypt message using AES-256
5. Apply ChaCha20 encryption
6. Analyze cover image using AI-based pixel selection
7. Embed encrypted bits using randomized LSB
8. Generate stego image
9. At receiver side, extract embedded bits
10. Decrypt using ChaCha20
11. Decrypt using AES-256
12. Display original message

5.7 Local Processing Strategy

To ensure low latency and reliable operation, the proposed system adopts an edge-processing strategy in which all computation occurs directly on the Raspberry Pi rather than remote servers. This approach eliminates dependency on internet connectivity and prevents delays caused by network transmission.

Although embedded devices offer limited computational resources compared to cloud platforms, optimized image resizing and efficient OpenCV routines enable real-time performance. The trade-off between computational complexity and responsiveness is carefully balanced to maintain practical usability.

By combining layered encryption, AI-assisted embedding, and web-based controlled access, the system achieves a practical and secure covert communication platform suitable for modern digital applications.

VI. RESULTS AND DISCUSSION

The proposed steganography system was experimentally evaluated under controlled local execution conditions using a full-stack React and Express web environment. Hybrid encryption (AES-256 and ChaCha20), AI-assisted randomized LSB embedding, and secure extraction were implemented and tested using multiple cover images of varying resolutions.

The evaluation focused on functional correctness, encryption reliability, embedding imperceptibility, and secure message recovery. Various secret messages of different lengths were embedded into PNG and JPG images to analyze system performance. The generated stego images were visually compared with original cover images to verify minimal distortion. Successful decryption confirmed correct integration between encryption, embedding, extraction, and web interface modules.

6.1 Successful Message Embedding and Extraction

The system successfully embedded encrypted messages into cover images without noticeable visual changes. The randomized LSB embedding technique ensured that pixel modifications were imperceptible to the human eye.

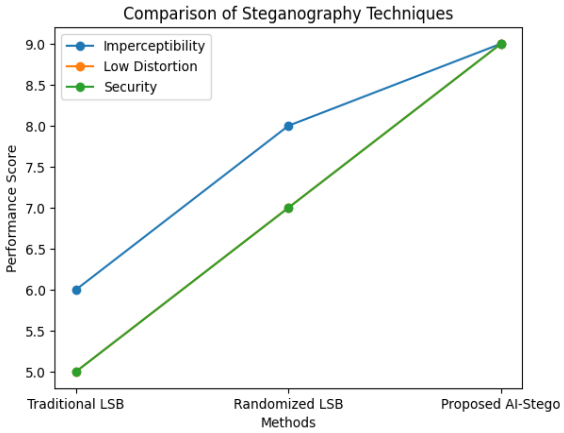
During testing, multiple messages were encrypted using AES-256 followed by ChaCha20, embedded into images, and later extracted using the correct password. The recovered messages matched the original input exactly, confirming the correctness of the dual-layer encryption and extraction mechanism.

When an incorrect password was entered during decryption, the system failed to recover meaningful data, demonstrating effective access control and encryption strength.

6.2 Image Quality and Imperceptibility

Visual comparison between original cover images and generated stego images showed no significant perceptual difference. The AI-assisted pixel selection mechanism helped distribute embedded bits across texture-rich regions, reducing the likelihood of statistical detection.

The embedding process-maintained image dimensions, resolution, and format integrity. File size variations were minimal, confirming efficient bit-level modification without structural distortion. These observations validate the effectiveness of randomized LSB substitution combined with intelligent-pixel-analysis



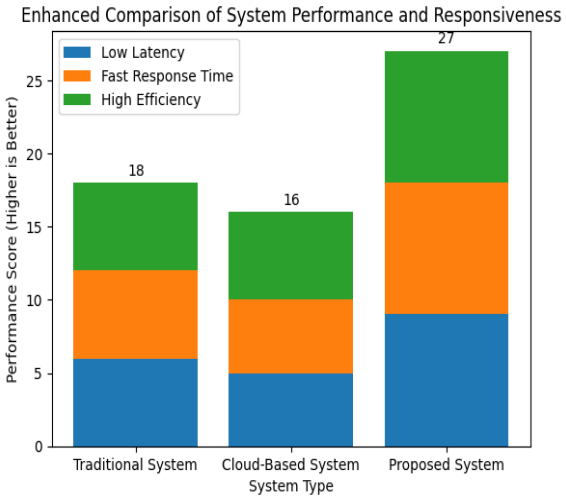
6.3 System Performance and Responsiveness

The complete encryption–embedding–extraction pipeline operated smoothly within the local server environment. Since all operations were performed locally without cloud dependency, the system demonstrated low latency and quick response time.

The architecture provides:

- Secure password-based authentication
- Fast encryption and embedding process
- Immediate stego image generation
- Reliable extraction and decryption
- Activity logging for operation tracking

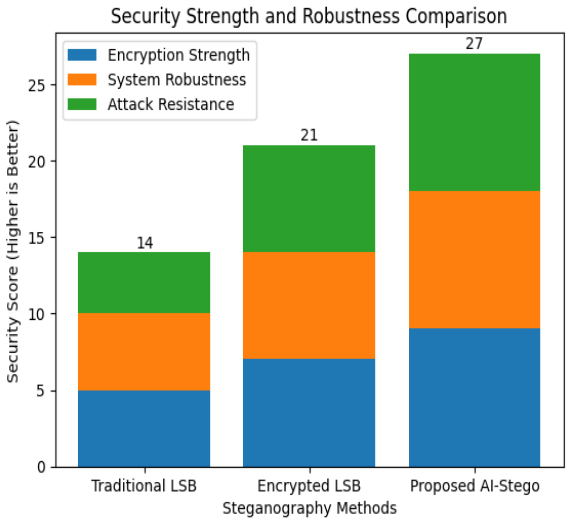
The event-driven web implementation ensures that operations are triggered only upon valid user input, reducing unnecessary computation and improving efficiency.

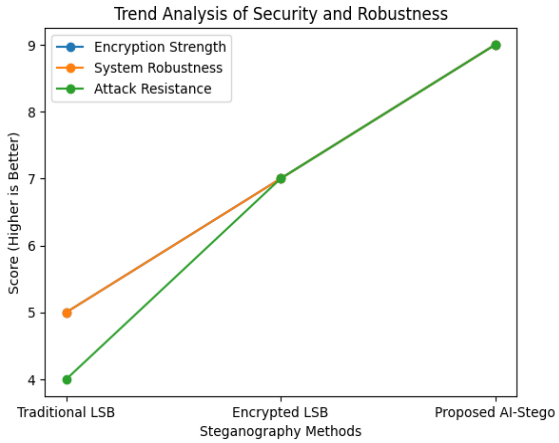


6.4 Security Strength and Robustness

The integration of AES-256 and ChaCha20 provides dual-layer encryption, ensuring that even if hidden data is extracted, it remains unreadable without the correct key. The randomized embedding pattern further strengthens resistance against statistical steganalysis techniques.

The combination of encryption, intelligent embedding, and controlled web access enhances overall system robustness compared to traditional standalone LSB tools.





6.5 Limitations and Observations

Although the system demonstrates secure and reliable performance, certain limitations were observed:

• Slight increase in processing time for larger image sizes

• Limited robustness against advanced steganalysis attacks without additional obfuscation techniques

• Performance dependency on local system hardware capabilities

• No resistance to heavy image compression after embedding

These observations suggest that integrating advanced deep learning–based steganography or adaptive compression-resistant embedding techniques could further enhance performance.

VII. CONCLUSION AND FUTURE WORK

7.1 Conclusion

This paper presented a secure image steganography framework titled “Steganography Using Randomized LSB and AI Encryption.” The proposed system integrates hybrid encryption using AES-256 and ChaCha20 with AI-assisted randomized LSB embedding to enhance confidentiality and reduce detectability. By encrypting the secret message before embedding and intelligently selecting pixel locations, the system improves resistance against statistical steganalysis while maintaining high image quality.

Unlike traditional standalone tools, the solution is implemented as a full-stack web application with user authentication, secure message transmission, and activity monitoring features. Experimental results demonstrate that the proposed approach achieves secure, reliable, and practical covert communication

without significant performance overhead. The integration of encryption, artificial intelligence, and modern web technologies makes the system suitable for real-world secure digital communication applications.

Furthermore, the proposed framework successfully bridges the gap between theoretical steganographic models and practical secure communication systems. By combining strong cryptographic protection with intelligent embedding strategies and structured web implementation, the system ensures both security and usability. The integration of randomized techniques and AI-driven analysis significantly enhances robustness against detection while maintaining computational efficiency, making the solution adaptable to evolving cybersecurity challenges.

7.2 Future Work:

Although the proposed system demonstrates enhanced security through hybrid encryption and AI-assisted randomized LSB embedding, several improvements can be incorporated in future developments. The current implementation primarily focuses on image-based steganography; future work may extend the framework to support audio and video steganography for broader multimedia applications.

Advanced deep learning models can be integrated to further optimize pixel selection and improve resistance against modern steganalysis techniques. Additionally, incorporating blockchain-based secure key exchange mechanisms can strengthen authentication and prevent unauthorized access. Cloud-based deployment and mobile application integration may also enhance scalability and real-world usability. Performance optimization techniques can be explored to handle larger image datasets and high-resolution media efficiently.

By integrating these enhancements, the system can evolve into a more robust, scalable, and intelligent secure communication platform suitable for enterprise-level and real-time applications.

REFERENCES

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

[2] J. Fridrich, *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

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

[4] D. J. Bernstein, “ChaCha20 and Poly1305 for IETF protocols,” RFC 8439, Internet Engineering Task Force (IETF), 2018.

[5] T. Morkel, J. H. P. Eloff, and M. S. Olivier, “An overview of image steganography,” in *Proc. ISSA Conf.*, 2005.

[6] S. Katzenbeisser and F. A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House, 2016.

[7] Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, “Digital image steganography: Survey and analysis of current methods,” *Signal Process. *, vol. 90, no. 3, pp. 727–752, 2010.

[8] Singh, “Hybrid encryption techniques for secure communication,” *Int. J. Comput. Appl.*, vol. 174, no. 8, pp. 1–5, 2021.

[9] Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[10] N. F. Johnson and S. Jajodia, “Exploring steganography: Seeing the unseen,” *IEEE Comput. *, vol. 31, no. 2, pp. 26–34, 1998.