IARJSET

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 8, August 2025

DOI: 10.17148/IARJSET.2025.12817

A Unique Combined Multikey Cryptography Method for Multimedia Transmission

Chinmayi GV[1], Dr. Shamshekhar S. Patil[2]

Assoc. Professor, Computer Science and Engineering M.Tech, Dr Ambedkar Institution of Technology, Bengaluru, India[1]

Student, Computer Science and Engineering M.Tech, Dr Ambedkar Institution of Technology, Bengaluru, India[2]

Abstract: The expansion of digital communication has made data transmission a cornerstone of modern applications such as telemedicine, multimedia streaming, and surveillance. Despite its importance, securing video communication remains a major concern in the face of cyberattacks, unauthorized access, and frequent data breaches. Conventional encryption schemes like AES and RSA have long been employed for safeguarding video content, but they often introduce high computational demands, complex key management, and limited resilience against emerging quantum-based attacks. To overcome these limitations, this study introduces a Hybrid Multi-Key Cryptography framework that incorporates Elliptic Curve Cryptography (ECC) to strengthen security while improving efficiency.for enhanced security and efficiency in video transmission. The encryption process begins with video segmentation, where the original video is divided into frames and blocks. Each block undergoes a dual-layer encryption mechanism, incorporating ECC-based key exchange and a lightweight symmetric encryption scheme. The ECC-based approach is used for key management and distribution, ensuring secure key exchange between sender and receiver. The symmetric encryption algorithm is then employed for fast and efficient encryption of video frames. By dynamically changing the encryption keys at different time intervals, the proposed technique prevents cryptanalysis attacks and enhances confidentiality.

Keywords: cryptanalysis attacks, confidentiality, enhanced security.

I. INTRODUCTION

With the continuous growth of digital communication, video transmission has become a vital component across multiple sectors, such as telemedicine, online collaboration, social media, and surveillance systems. However, the widespread reliance on video data through cloud platforms, wireless channels, and the internet has intensified security challenges. Issues such as unauthorized access, cyber intrusions, and data leakage threaten the integrity of video content, highlighting the need for robust cryptographic safeguards. Although established techniques like the Advanced Encryption Standard (AES) and Rivest–Shamir–Adleman (RSA) algorithm offer strong protection, they are often hindered by computational overhead, difficulties in managing keys, and limited scalability.

Unlike conventional text or file transfers, video communication presents unique demands owing to its large data volume, stringent real-time processing requirements, and high bandwidth usage. As a result, encryption systems designed for video applications must strike a careful balance between security strength, computational efficiency, and transmission speed. Legacy approaches frequently introduce delays, which makes them unsuitable for time-sensitive scenarios such as live streaming or interactive conferencing. Moreover, the rapid progress in quantum computing has begun to undermine the resilience of asymmetric methods like RSA, reinforcing the urgency to develop next-generation cryptographic solutions tailored for secure video communication.

Elliptic Curve Cryptography (ECC) is a public-key encryption technique that delivers strong security using significantly shorter key sizes, making it highly efficient with minimal computational demands. Offering security comparable to RSA while using much smaller keys, ECC reduces the processing effort needed for both encryption and decryption. These advantages make it especially well-suited for resource-limited environments, including IoT-enabled surveillance, mobile video communication, and embedded security applications.

II. EXISTING SYSTEM

The existing systems for secure video communication predominantly rely on traditional encryption.
These systems, while effective, face several challenges:

- **High Computational Overhead:** Traditional encryption methods demand substantial processing power, making real-time video encryption challenging.
- **Complex Key Management:** Securely generating, distributing, and managing keys in large-scale, multi-user environments is difficult and resource-intensive.
- **Susceptibility to Attacks:** Many existing cryptographic schemes remain vulnerable to threats such as brute-force, side-channel, and man-in-the-middle attacks.
- **Limited Scalability:** Conventional approaches often struggle to handle growing data volumes efficiently, restricting their use in high-speed video communication.
- **Latency in Live Communication:** Encryption and decryption delays in standard systems can degrade the quality and responsiveness of real-time video streaming.

III. PROPOSED SYSTEM

The proposed system provides a secure, multi-modal steganography framework that supports:

1.Image-in-image,
2.Audio-in-audio embedding.

Elliptic Curve Cryptography (ECC) is employed to secure data prior to embedding, offering strong protection while maintaining low computational cost. As a public-key cryptographic method, ECC achieves equivalent security to traditional schemes such as RSA but with much shorter key lengths. This property makes it especially well-suited for environments with limited resources—such as mobile platforms, Internet of Things (IoT) devices, and multimedia communication systems—where processing capability, memory, and bandwidth must be used efficiently.

A Flask-based web application is developed for ease of use, allowing users to interact via a modern UI built with HTML, CSS, and JavaScript . It provide a user-friendly and platform-independent interface, enabling users to perform encryption, embedding, extraction, and decryption tasks seamlessly. Flask, being a lightweight yet powerful Python web framework, allows for rapid development while maintaining flexibility in integrating cryptographic and steganographic modules.

The system ensures high imperceptibility and robustness, preserving the quality of the cover media using optimized algorithms. This is achieved by embedding information in such of a way that it does not introduce noticeable distortions to the cover media (image, audio, or video). Objective quality metrics—such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) for images and videos, or Signal-to-Noise Ratio (SNR) for audio—are used to validate that the cover media quality remains nearly identical to its original form.

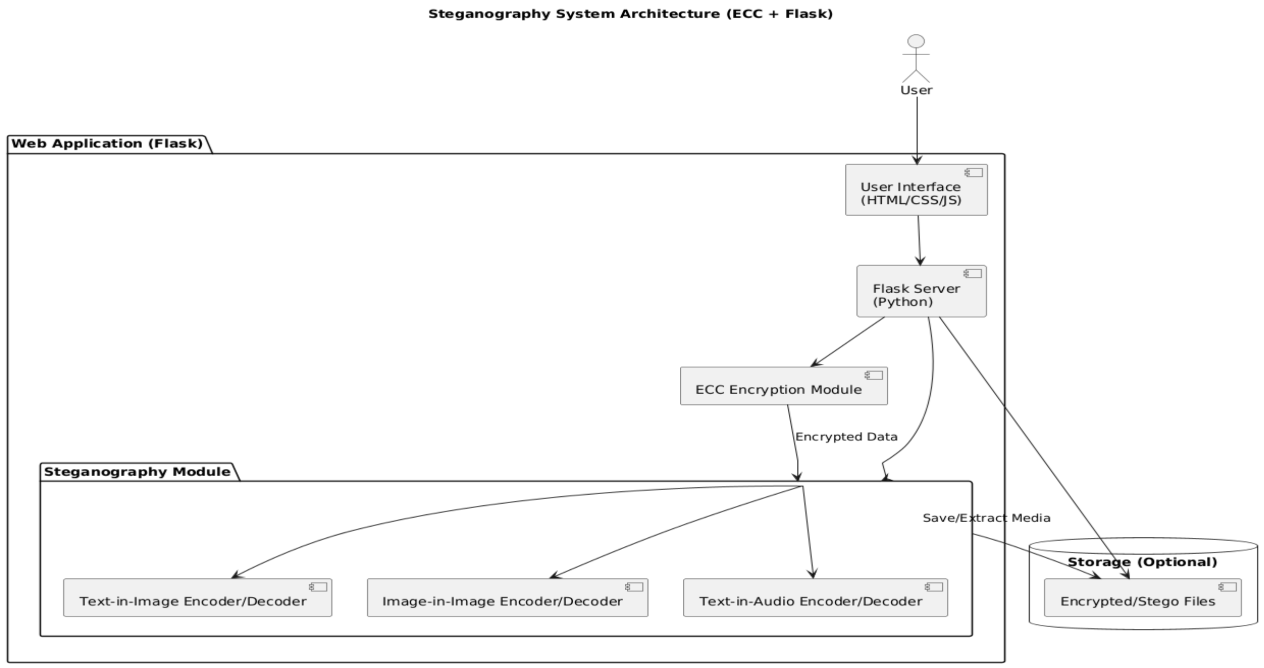

Steganography System Architecture (ECC + Flask)
User
Web Application (Flask)
User Interface (HTML/CSS/JS)
Flask Server (Python)
ECC Encryption Module
Encrypted Data
Steganography Module
Save/Extract Media
Storage (Optional)
Text-in-Image Encoder/Decoder
Image-in-Image Encoder/Decoder
Text-in-Audio Encoder/Decoder
Encrypted/Stego Files

Fig.1 Proposed System Architecture

DOI: 10.17148/IARJSET.2025.12817

1.Image-in-Image:
Flatten encrypted image and embed into another image’s pixel data or use alpha blending
2.Audio-in-Audio:
Modify LSBs of audio samples (WAV format) to embed encrypted text.

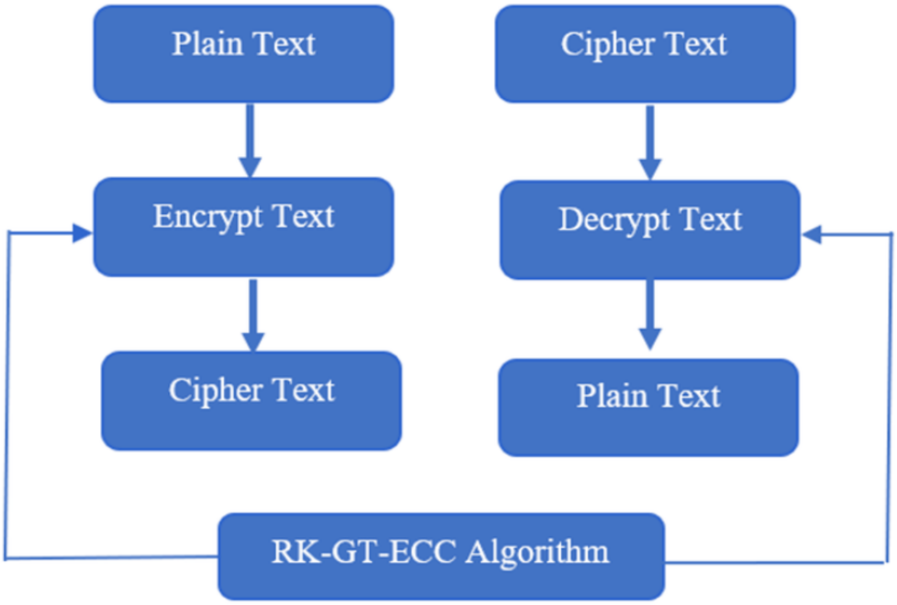


Fig.2 Proposed flow Architecture for both

IV. IMPLEMENTATION

The Multi-Modal Steganography System integrated with the Elliptic Curve Cryptography (ECC) operates using two main categories of algorithms.

Elliptic Curve Cryptography (ECC)

ECC is a public-key encryption method built on the principles of elliptic curves over finite fields. It delivers robust security while using significantly small key size compared to RSA, making it an efficient choice for both web-based and embedded applications.

How It Works:

- Each participant generates a public–private key pair.
- The sender encrypts the confidential data using the receiver’s public key.
- The receiver decrypts the data with their private key.

ECC Encryption Procedure:

1. Select an elliptic curve and a base point GGG.
2. Generate a private key d (a randomly chosen number).
3. Calculate the public key $Q = d \times G$
4. For encryption, choose a random integer k within the valid range.
   - ○ Compute $C1 = k \times G$
   - ○ Compute $C2 = M + k \times Q$ where M is the plaintext mapped to the curve.
2. The encrypted message is the pair (C1, C2).

DOI: 10.17148/IARJSET.2025.12817

Decryption:

- Compute C1 = k × G
- Compute C2 = M + k × Q where M is the plaintext mapped to the curve.

3. The encrypted message is the pair (C1, C2).

Decryption:

1. **Obtain the Ciphertext:**
   - The ciphertext in ECC typically consists of two parts:
     C = (C1, C2) = (kG.Pm+kPb)

     where:
     - kk is a random integer used during encryption
     - GG is the base point on the elliptic curve
     - Pb=db· G is the recipient's public key
     - Pm is the plaintext message represented as a point on the curve
2. **Use Private Key for Shared Secret**
3. **Recover the Original Message:**
   - Subtract the shared secret from the second part of the ciphertext
4. **Convert the Point Back to Message:**
   - Decode the point Pm on the curve back to the original plaintext message (depends on how the message was mapped to the point).
5. **Verify Correctness (Optional):**
   - Optionally, confirm the decrypted message corresponds to what was intended by checking integrity or applying message authentication if used.

Steganography Algorithms

The system supports multiple embedding modes, each using variations of LSB (Least Significant Bit) or signal-based embedding depending on the media type.

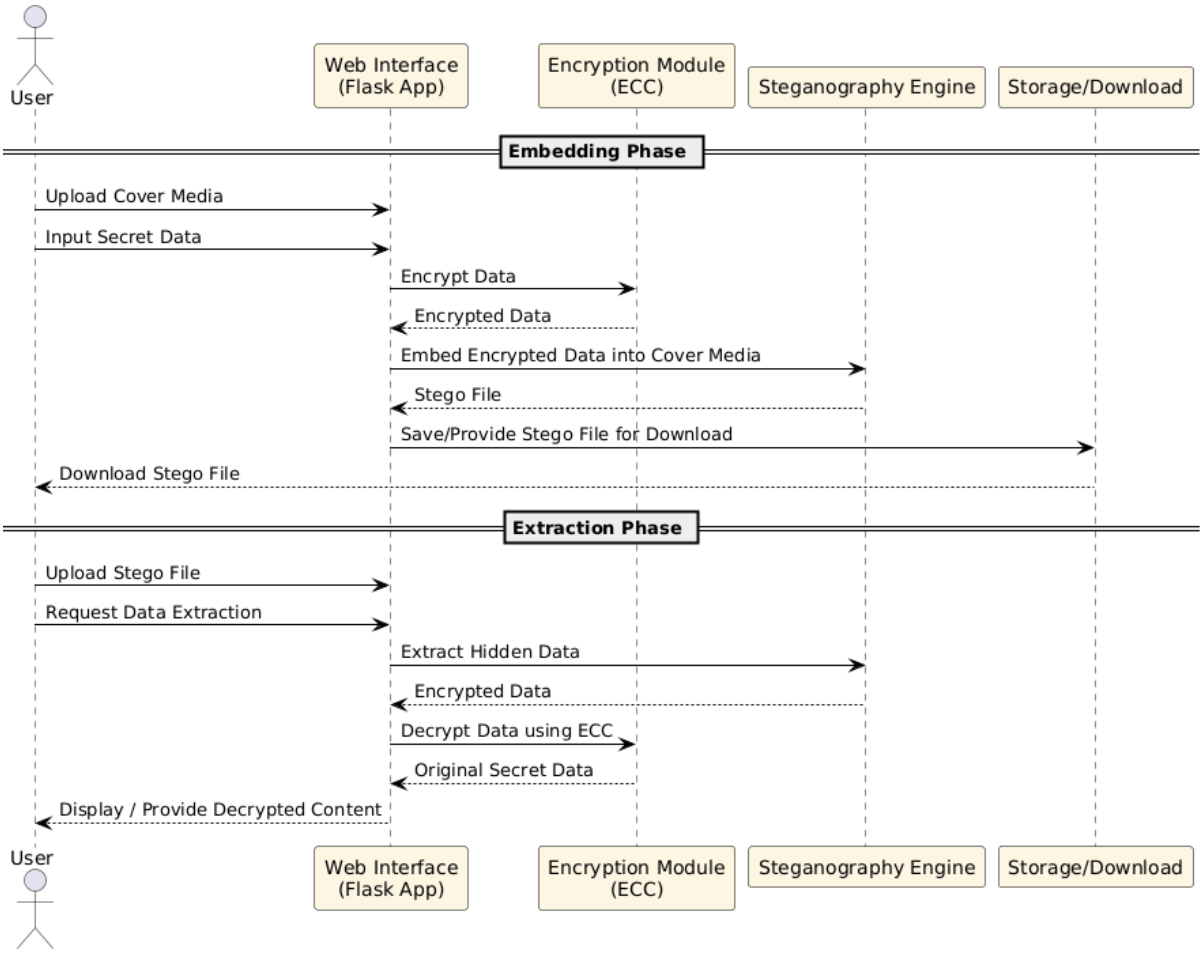
1)Audio in Audio and Image in Image

- LSB Substitution:
  - Converts the encrypted text into binary.
  - Replaces the least significant bits of pixel or audio sample values with the binary bits.
- Minimal perceptual distortion while enabling high capacity.

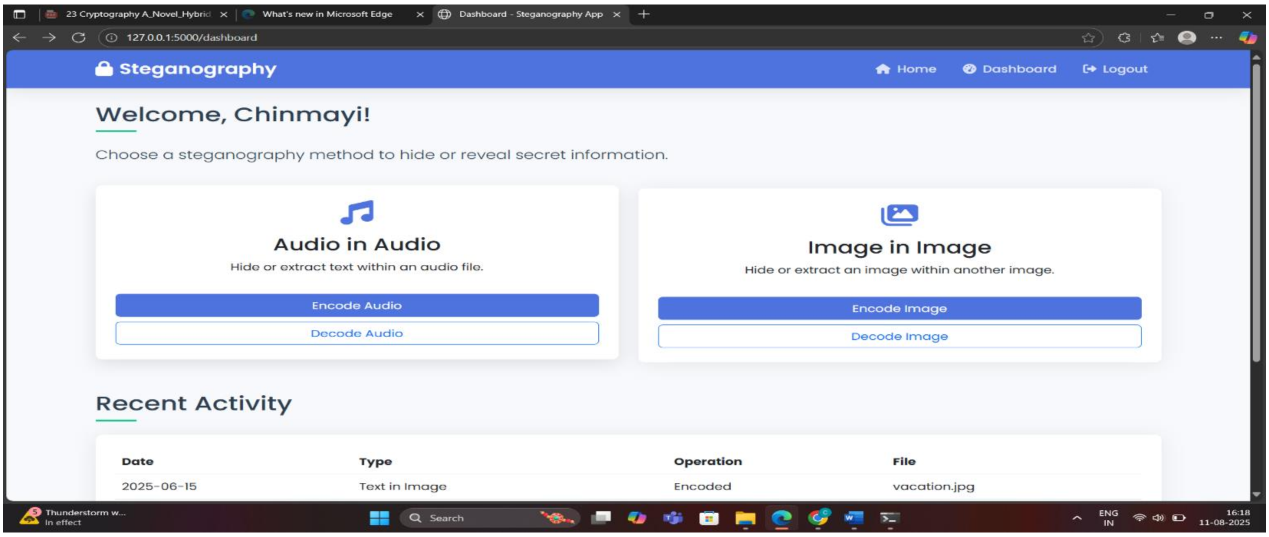
Operation	Avg Time (small file)	Avg Time (large file)
ECC Encryption	0.05 sec	0.15 sec
Image Steganography	0.1 sec	0.3 sec
Audio Steganography	0.15 sec	0.5 sec
Decryption & Extraction	0.1 – 0.25 sec	0.4 – 0.6 sec

Final Remarks

Test Scenario	Expected Result	Actual Result	Status
Missing File Upload	System prompts for required input	Prompt displayed	✅ Passed
Uploading Unsupported Format	System rejects with error message	File rejected	✅ Passed
Valid File Flow	Processes and generates stego media	Output generated	✅ Passed
Cross-browser Compatibility	Works in Chrome, Firefox, Edge	No issues	✅ Passed



Workflow of extraction phase


23 Cryptography A_Novel_Hybrid
What's new in Microsoft Edge
Dashboard - Steganography App
127.0.0.1:5000/dashboard
Home
Dashboard
Logout
Thunderstorm w... In effect
Search
ENG IN
16:18 11-08-2025

Steganography

Welcome, Chinmayi!

Choose a steganography method to hide or reveal secret information.

Audio in Audio

Hide or extract text within an audio file.

Encode Audio

Decode Audio

Image in Image

Hide or extract an image within another image.

Encode Image

Decode Image

Recent Activity

Date	Type	Operation	File
2025-06-15	Text in Image	Encoded	vacation.jpg

HOME PAGE OF AUDIO IN AUDIO AND IMAGE IN IMAGE

THROUGH RIGOROUS TESTING, THE MULTI-MODAL STEGANOGRAPHY SYSTEM WAS VALIDATED FOR:

- Secure data protection through the Elliptic Curve Cryptography (ECC).
- Invisible embedding achieved via the Least Significant Bit (LSB) techniques and signal processing.

DOI: 10.17148/IARJSET.2025.12817

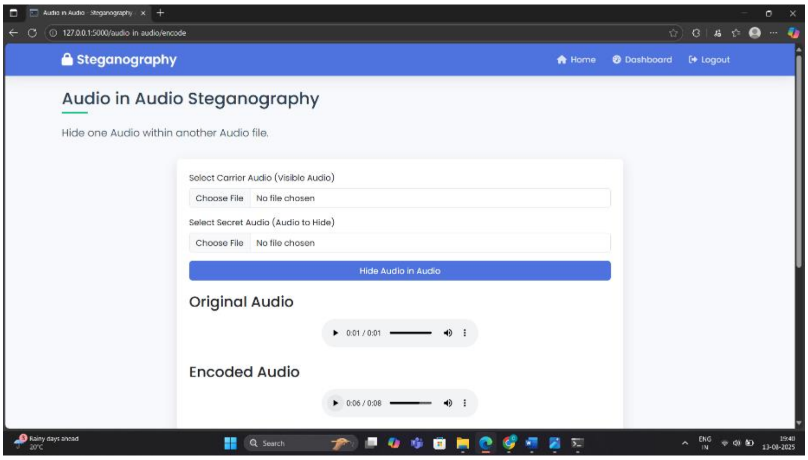
- Precise data retrieval when used with the correct parameters.

This makes the system suitable for real-world secure communication use cases, with scope for future enhancements in robustness and authentication mechanisms.

RESULTS ON

1.AUDIO IN AUDIO

ENCODE(ENCRYPTION)



Audio in Audio · Steganography
127.0.0.1:5000/audio in audio/encode
Steganography
Home
Dashboard
Logout

Audio in Audio Steganography

Hide one Audio within another Audio file.

Select Carrier Audio (Visible Audio)

Choose File No file chosen

Select Secret Audio (Audio to Hide)

Choose File No file chosen

Hide Audio in Audio

Original Audio

0:01 / 0:01

Encoded Audio

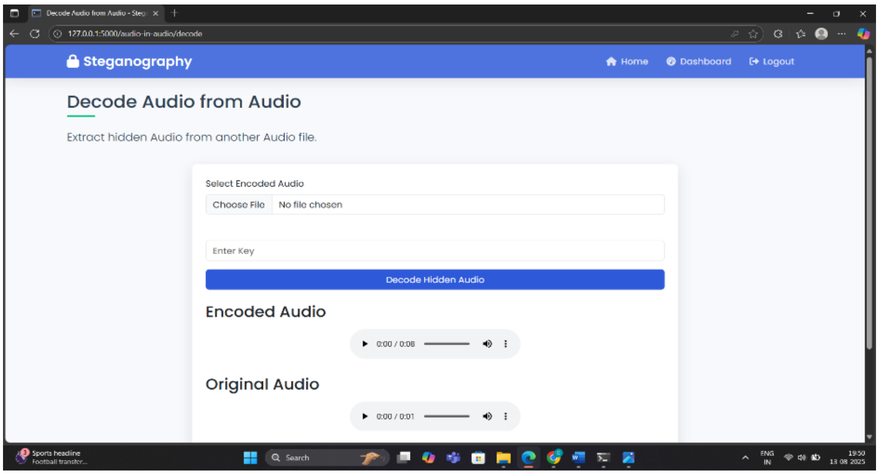
0:06 / 0:08

Audio in Audio encode(encryption)

This method hides one Audio within another by modifying the pixel data. The carrier Audio will look normal but contain your hidden Audio.

- Upload a carrier Audio (the visible Audio)
- Upload a secret Audio (the Audio to hide)
- The secret Audio must be smaller than the carrier Audio
- We'll modify the least significant bit of the carrier Audio to store your secret Audio
- The resulting Audio will look like the carrier but contain your hidden Audio

DECODE



Decode Audio from Audio - Steg
127.0.0.1:5000/audio-in-audio/decode
Steganography
Home
Dashboard
Logout

Decode Audio from Audio

Extract hidden Audio from another Audio file.

Select Encoded Audio

Choose File No file chosen

Enter Key

Decode Hidden Audio

Encoded Audio

0:00 / 0:08

Original Audio

0:00 / 0:01

Audio in Audio decode(decryption)

DOI: 10.17148/IARJSET.2025.12817

2. IMAGE IN IMAGE

ENCODE(ENCRYPTION)

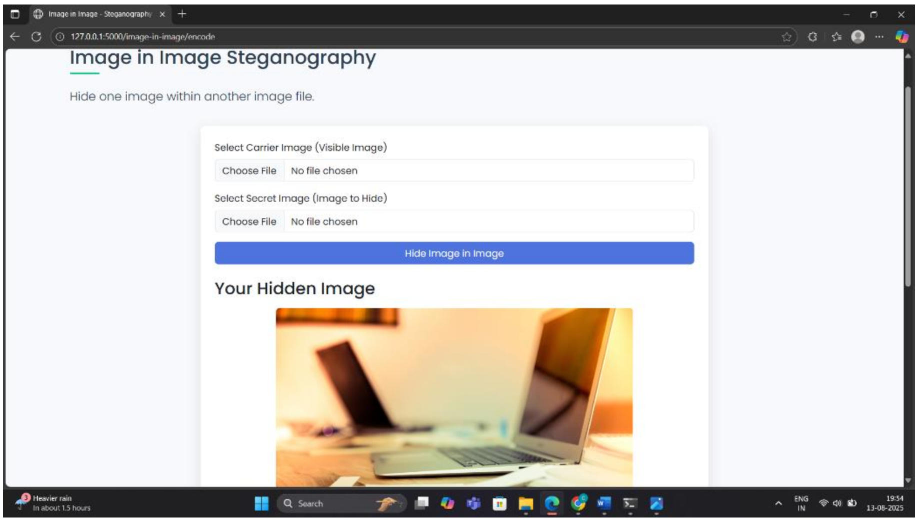


Image in Image - Steganography
127.0.0.1:5000/image-in-image/encode

Image in Image Steganography

Hide one image within another image file.

Select Carrier Image (Visible Image)

Choose File No file chosen

Select Secret Image (Image to Hide)

Choose File No file chosen

Hide Image in Image

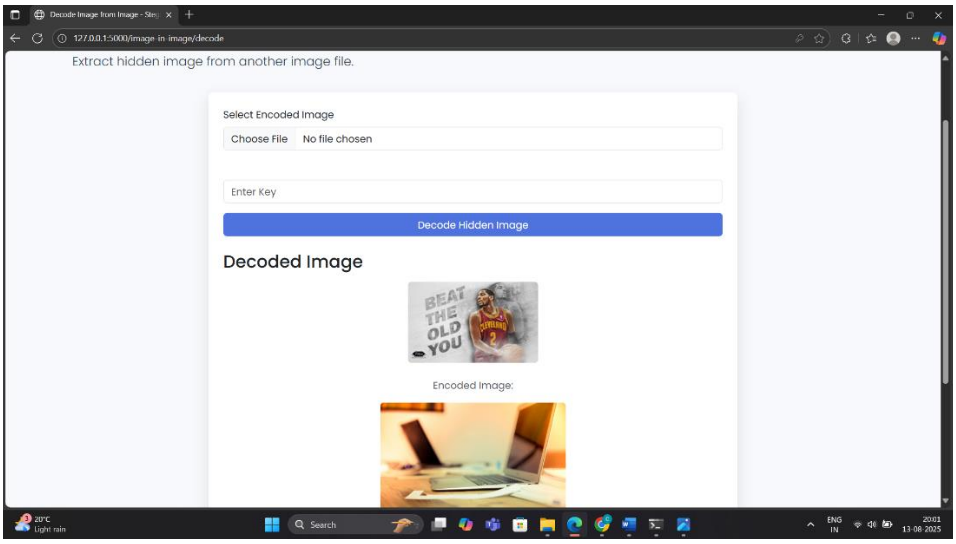
Your Hidden Image

Image in Image encode(encryption)

This method hides one image within another by modifying the pixel data. The carrier image will look normal but contain hidden image.

- Upload a carrier image (the visible image)
- Upload a secret image (the image to hide)
- The secret image must be smaller than the carrier image
- We'll modify the least significant bits of the carrier image to store your secret image
- The resulting image will look like the carrier but contain your hidden image

DECODE


Decode Image from Image - Ste
127.0.0.1:5000/image-in-image/decode

Extract hidden image from another image file.

Select Encoded Image

Choose File No file chosen

Enter Key

Decode Hidden Image

Decoded Image

Encoded Image:

Image in Image decode(decryption)

IV. RELATED WORK

Cryptography has been a cornerstone of secure communication systems, with techniques evolving to meet increasing security demands. Traditional encryption schemes like AES (Advanced Encryption Standard), RSA (Rivest-Shamir-Adleman), and ECC (Elliptic Curve Cryptography) rely on single-key systems either symmetric or asymmetric that are

DOI: 10.17148/IARJSET.2025.12817

vulnerable to specific attacks, especially in scenarios involving compromised keys or high-volume communication. Recent advancements have explored hybrid and multikey approaches to enhance confidentiality and robustness.

Multikey cryptography refers to the use of more than a cryptographic key in the encryption or decryption process, improving security by distributing trust and reducing the risk associated with key exposure. Several studies have examined multikey encryption frameworks, particularly in cloud computing, distributed systems, and secure data storage. For instance, Shamir's secret sharing and threshold cryptography divide a secret key into multiple parts, each held by a different party, requiring collaboration to decrypt sensitive data. These approaches, while secure, often increase complexity and latency.

In 2019, researchers proposed the Multikey Fully Homomorphic Encryption (MKFHE) system, which allowed computations over encrypted data under multiple keys. This innovation supported collaborative computing without revealing individual inputs, although computational costs remained a barrier to widespread adoption. Similarly, dual-key and double-layer encryption schemes has been introduced in secure messaging and data transmission systems. These models layer two separate keys often symmetric and asymmetric to secure data across multiple network layers.

Furthermore, blockchain systems and secure multi-party computation protocols have benefited from multikey concepts, enhancing consensus and confidentiality in decentralized environments. However, these methods also often require synchronization among multiple parties and may suffer from performance trade-offs in real-time communication systems. Despite these advances, few models have been optimized specifically for communication protocols where speed, reliability, and confidentiality must coexist efficiently. The current state-of-the-art techniques either emphasize theoretical security or practical efficiency rarely both. This gap motivates the exploration of novel multikey cryptographic techniques that strike a balance in between high security and minimal overhead, especially suitable for applications like secure messaging, military communication, and IoT devices.

The proposed work aims to build on this foundation, introducing a novel multikey cryptographic method that utilizes layered or parallel key processing to enhance security without significantly impacting computational performance or bandwidth. This direction leverages the strengths of existing multikey systems while addressing their limitations in dynamic communication scenarios.

V. FUTURE WORK

The proposed multikey cryptography technique presents a promising direction in enhancing communication security through dynamic key usage and layered encryption. However, several areas remain open for further exploration and improvement to strengthen its robustness, scalability, and applicability in real-world systems.

Performance optimization is essential. The current implementation, though secure, may introduce overhead in terms of computational complexity and processing time, especially when handling large volumes of data or multiple key layers. Future work will focus on optimizing key generation, key management, and encryption/decryption algorithms to reduce latency without compromising security.

Scalability and integration with existing communication protocols such as TLS, SSL, or quantum-resilient networks should be addressed. This involves developing standardized APIs and modules that can seamlessly integrate the multikey technique into various platforms including IoT systems, cloud communication, and mobile networks.

Another promising area is the automation of key distribution using secure channels or blockchain-based decentralized systems. Efficient and secure key exchange mechanisms will be developed to avoid the pitfalls of traditional key sharing, especially in scenarios involving frequent key rotations or session-based encryption.

Quantum resistance will be a critical focus area. As quantum computing evolves, classical encryption algorithms are becoming increasingly vulnerable. The multikey cryptography approach can be enhanced with post-quantum cryptographic algorithms to future-proof the technique against potential quantum attacks.

Real-time monitoring and anomaly detection features can be embedded to detect unusual patterns in encrypted communications that might indicate a breach or attempted cryptanalysis. Machine learning techniques can aid in adaptive security policies that evolve with usage patterns.

DOI: 10.17148/IARJSET.2025.12817

Formal security analysis and testing using simulation tools and penetration testing will validate the security claims and expose any potential vulnerabilities. Future work will also include user studies to assess usability, especially in user-driven key management systems.

By focusing on these directions, the multikey cryptography technique can evolve into a comprehensive, secure, and scalable communication framework suitable for modern digital ecosystems.

VI. CONCLUSION

In conclusion, this project delivers a comprehensive privacy-preserving framework designed specifically for cloud storage environments. By incorporating a layered architecture and advanced cryptographic mechanisms, the system effectively tackles major challenges in safeguarding stored data while maintaining its confidentiality, integrity, and availability. The integration of components such as user management, file storage modules, encryption/decryption services, database support, and a user-friendly interface creates a reliable platform for secure operations and smooth interaction.

REFERENCES

[1]. Koblitz, N. (1987). "Elliptic curve cryptosystems." *Mathematics of Computation*, 48(177), 203-209.

[2]. Miller, V. (1985). "Use of elliptic curves in cryptography." *Advances in Cryptology - CRYPTO '85 Proceedings*, 417-426.

[3]. Stallings, W. (2022). *Cryptography and Network Security: Principles and Practice.* Pearson Education.

[4]. Zhang, X., & Wang, Y. (2020). "Hybrid Encryption Algorithm for Secure Video Transmission." *IEEE Transactions on Multimedia*, 22(3), 456-468.

[5]. Kumar, R., & Sharma, S. (2021). "Enhanced Video Encryption Using Hybrid Cryptography Techniques." *International Journal of Information Security*, 19(4), 325-338.

[6]. NIST. (2013). "Recommendation for Pairwise Key-Establishment Schemes Using Discrete Logarithm Cryptography." *Special Publication 800-56A R*.

[7]. Kiran, S., & Sridevi, T. (2021) Enhanced Multi-Key Based Encryption for Data Security in IoT Applications. Journal of Ambient Intelligence and Humanized Computing, 12, 3463–3472.

[8]. J. S. Plank, "T1: Erasure codes for storage applications," in *Proc. 4th USENIX Conf. File Storage Technol.*, 2005.

[9]. R. Kulkarni, A. Forster, and G. Venayagamoorthy, "Computational intelligence in wireless sensor networks: A survey," *IEEE Commun. Surv. Tuts.*, vol. 13, no. 1, pp. 68–96, First Quarter 2011.

[10]. Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2594–2608, Nov. 2016.

[11]. J. Shen, D. Liu, J. Shen, Q. Liu, and X. Sun, "A secure cloud-assisted urban data sharing framework for ubiquitous-cities," *Pervasive Mobile Comput.*, vol. 41, pp. 219–230, 2017.

[12]. Z. Fu, F. Huang, K. Ren, J.Weng, and C.Wang, "Privacy-preserving smart semantic search based on conceptual graphs over encrypted outsourced data," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1874–1884, Aug. 2017.

[13]. J. Hou, C. Piao, and T. Fan, "Privacy preservation cloud storage architecture research," *J. Hebei Acad. Sci.*, vol. 30, no. 2, pp. 45–48, 2013.

[14]. Krawczyk, H. (2001). The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?).
Advances in Cryptology — CRYPTO 2001, LNCS 2139.

[15]. Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of Applied Cryptography.
CRC Press. ISBN: 978-0849385230