International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 10-No.3 (2024) pp. 434-460
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

A Secure Fusion: Elliptic Curve Encryption Integrated with LSB Steganography for Hidden Communication

Homam EL-TAJ

Cybersecurity Department, Dar Al-Hekma University, Jeddah, Saudi Arabia
* **Corresponding Author Email:** h.eltaj@dah.edu.sa **ORCID:** 0000-0001-7565-4760

Article Info:

DOI: 10.22399/ijcesen.382
Received : 16 July 2024
Accepted : 02 September 2024

Keywords :

Elliptic Curve Encryption
LSB Steganography
Secure Communication
Hidden Communication
Image Steganography

Abstract:

In today's digital age, ensuring secure communication is essential. This article presents a novel approach for hidden communication by integrating Elliptic Curve Encryption (ECE) with Least Significant Bit (LSB) Steganography.
Our proposed fusion offers a robust solution, Stegno Curve for concealing sensitive information within innocuous cover media while encrypting it using elliptic curve cryptography. By leveraging the strengths of both techniques, we achieve enhanced security and confidentiality in data transmission.
Through a comprehensive literature review, methodology explanation, security analysis, and implementation details, we demonstrate the feasibility and effectiveness of the Stegno Curve Method. The findings of this study not only contribute to advancing the field of secure communication but also open avenues for practical applications in various domains, such as secure cloud transitions, smart home technologies, and data encryption.

1. Introduction

The imperative for secure and confidential data transmission has become more pressing in an era marked by the exponential growth of digital communication and increasing cyber threats. From sensitive financial transactions to confidential government communications, the potential ramifications of data breaches are profound [1, 2]. Moreover, with the emergence of stringent data protection regulations and the growing awareness of privacy rights among individuals, organizations are under increasing pressure to adopt robust security measures.
Figure 1 shows the number of user accounts exposed from data breaches globally, for each quarter of 2020 through Q4 2023. Data is given in millions, and across the period, the numbers fluctuate wildly. The highest number of exposed accounts was in Q4 2020 at about 125.74 million accounts. A peek at the trend reveals the peak of data breaches near the end of 2020, followed by a general decrease, with spikes in Q1 2021 and Q2 2023. This chart outlines the ongoing threat from data breaches and the critical need to increase measures for security in digital communications.

Figure 1: *Number of User Accounts Exposed Worldwide From 1st Quarter 2020 To 4th Quarter 2023 (In Millions)*

This has led to the development and utilization of various techniques, such as encryption and steganography, to ensure the confidentiality and integrity of sensitive information [1,2,12-25]. Encryption involves converting data into a form that can only be read by authorized parties, while steganography focuses on concealing the existence of the communication. However, existing methods

face challenges in providing integrated and robust solutions for secure communication [26-32].

1.1 Problem Statement

While encryption and steganography offer effective ways of securing digital communication independently, they present inherent limitations [24]. Although highly secure, traditional encryption methods, such as RSA and AES, may exhibit vulnerabilities to emerging cryptographic attacks [1, 3, 17]. Similarly, conventional LSB steganography techniques often suffer from poor payload capacity and susceptibility to detection. It is effective in hiding the message [11] but may not provide sufficient protection for the content [13]. Therefore, there is a growing need for a combined and secure solution that integrates encryption and steganography to address these limitations.

1.2 Purpose of the Article

This article aims to introduce a novel approach that integrates Elliptic Curve Encryption (ECC) with Least Significant Bit (LSB) Steganography to provide a comprehensive and secure communication solution [17, 32]. This integration leverages the strengths of both techniques, offering enhanced security and privacy for digital communication by encrypting the data using elliptic curve cryptography before embedding it within the LSB layers of cover media, we not only fortify the confidentiality of the concealed information but also mitigate the risk of detection or tampering. This integration enhances the security posture of the communication channel and introduces a novel paradigm for covert data transmission that transcends conventional boundaries [17, 15, 25].

This paper presents an innovative way of combining Elliptic Curve Encryption and Least Significant Bit Steganography to propose the Stegno Curve Method in order to contribute to secure communications. The use of ECE for resource-constrained environments considers the weaknesses of ECE that allow for robust encryption with shorter key lengths, together with the strengths. The same encrypted data is embedded in digital media surreptitiously using LSB steganography. This integration increases the security and secrecy of covert communication with an overall solution well-defended against cryptographic attacks and steganalysis. The Stegno Curve Method is potent enough for applications that demand high security and stealth, such as secure cloud transitions, smart home technologies, and sensitive data transmission. The article will also highlight the potential advantages of this fusion, demonstrating its applicability and effectiveness in ensuring secure and covert communication in the digital domain.

The following sections of the paper will be structured in this manner:

1. **Literature Review:** This section presents a synopsis of the recent literature regarding Elliptic Curve Encryption (ECE) and Least Significant Bit (LSB) Steganography, highlighting both the weaknesses and strengths of each method when used alone and in conjunction.
2. **Proposed Stegno Curve Methodology:** The methodology is elaborated on, explaining the processes involved in key generation, encryption, embedding, extraction, and decryption in detail. This section also includes mathematical formulations and algorithms used.
3. **Implementation and Results:** Implementation details with the details of the security analysis carried out. Performance metrics, system specifications, and the real-world effectiveness of the proposed method (table 1).
4. **Discussion:** Analysis of results, comparison with existing methods, practical implications, and possible applications of the Stegno Curve Method.
5. **Conclusion and Future Work:** It is an overview of the key findings, contribution to the field, and future research trends that would lead to enhancing secure communication techniques further.

2. Literature Review

The literature review for the article includes an overview of Elliptic Curve Encryption (ECC), LSB Steganography, existing integration approaches, and research on the usage of these algorithms.

2.1 Overview of Elliptic Curve Encryption

Elliptic Curve Cryptography (ECC) has emerged as a powerful encryption method renowned for its efficiency and security [1]. Unlike traditional methods such as RSA, ECC offers equivalent security with shorter key lengths, making it particularly suitable for resource-constrained environments such as mobile devices and IoT devices [3].

Elliptic Curve Cryptography (ECC) has garnered significant attention due to its ability to provide high security with shorter key lengths compared to traditional methods like RSA. ECC's efficiency makes it suitable for resource-constrained environments such as mobile devices and IoT

systems [32-51]. Recent advancements include AI-enhanced Schoof's algorithm for faster elliptic curve point computations and new methods for generating ECDSA moduli to improve Barrett's algorithm, thereby enhancing ECC's overall performance [47].

Siddharth and Saini [33] describe lightweight cryptography as a form of encryption with minimal CPU requirements, designed to ensure data security and privacy for IoT systems, which often have limited capabilities. Traditional encryption methods are unsuitable for mobile devices due to these constraints. Beyond addressing security and privacy concerns, it is crucial to consider the overall efficiency of the verification process. However, the authors do not provide a comprehensive evaluation of the security offered by various lightweight cryptographic algorithms. They merely assert that these techniques resist cryptanalysis attacks without providing detailed specifics. Side-channel attacks represent a significant threat to the security of cryptographic systems, particularly for resource-limited IoT devices. Siddharth and Saini's work does not explore the impact of side-channel attacks on the security of lightweight cryptographic methods. Their paper reviews existing schemes without discussing potential improvements for these methods.

In contrast, Ahmed et al. [34] propose a robust cybersecurity approach that combines encryption with authentication (ELCA), employing an elliptic curve Diffie-Hellman (ECDH) protocol for key allocation, effectively addressing concerns regarding weak bits in the shared secret key. While this study offers a valuable contribution to IoT security, it does not consider the implementation costs of the proposed mechanism, which could be a significant obstacle for resource-constrained devices.

Elliptic Curve Encryption (ECE) is a cryptographic technique based on the mathematical properties of elliptic curves over finite fields. It offers a high level of security while requiring smaller key sizes compared to traditional encryption algorithms like RSA and AES [35, 36]. The security of ECE relies on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP), making it resistant to attacks by classical and quantum computers [37]. ECC offers faster encryption and decryption than traditional algorithms, contributing to efficient cryptographic operations.

Nyangaresi [38] introduces an anonymity-oriented lightweight methodology tailored for IoT-based systems, leveraging elliptic curve point multiplication techniques and one-way hashing. This study grapples with the inherent complexity and error-prone nature of Elliptic Curve Cryptography (ECC), particularly in terms of parameter selection, key management, and addressing ECC-specific vulnerabilities. [39] present a distinctive encryption strategy for securing healthcare data within IoT-enabled health services, utilizing elliptic curve cryptography, Advanced Encryption Standard (AES), and Serpent. The primary challenge in this study is to design an encryption method that robustly protects sensitive patient information from unauthorized access and breaches, without introducing unnecessary complexity and performance issues due to the combined use of ECC and AES.

Xavier and Kesavan [40] propose a hybrid elliptic curve cryptography technique for IoT-based systems, involving the selection of the secret key for verification via particle swarm optimization or the cuckoo search algorithm. Additionally, this technique incorporates encoding and decoding through the Key-Value Pair (KVP) of Elliptic Curve Cryptographic. Despite its innovative approach, the algorithm's security poses a concern when compared to similar methods in terms of execution speed and accuracy. Qazi et al. [41] offer a method that secures node-to-node communication systems and optimizes memory usage on nodes by employing the Elliptic Curve Digital Signature Algorithm (ECDSA). This approach provides an effective mechanism for monitoring key reproduction rates, greeting message counts, and packet sizes. However, the study lacks a discussion on how the proposed protocol maintains effectiveness and adaptability over time as new security challenges arise. [42] suggests a novel technique based on elliptic curve cryptography (ECC) for verification and encoding, utilizing randomized numbers generated by fuzzy logic to enhance authentication and encryption in IoT systems. Nonetheless, the paper is limited by specific application requirements, usability aspects, and the challenge of adapting FECC to resource-constrained environments while ensuring security.

The research by [43] explores IoT security, covering symmetrical, asymmetrical, and hybrid cryptographic solutions. Asymmetric key cryptography is highlighted for its ability to secure communications among multiple individuals while preventing key distribution over insecure channels. Elliptic Curve Cryptography (ECC) is found to outperform other techniques in the study. The critical challenge remains the suitability of algorithms for resource-constrained IoT devices, underscoring the need to select cryptographic algorithms compatible with the specific limitations and requirements of IoT systems.

Pawar and Kalbande [44] emphasize the concept of contemporary access control systems based on elliptic curve encryption, highlighting the security of IoT components and applications. The primary challenge identified in this study is the development

of efficient cryptographic libraries or methods tailored to the constraints of IoT devices. Agrawal and Tiwari [45] explore the security and privacy measures available at each tier of the IoT model, along with the limitations and future directions for enhancing IoT security infrastructure. A reliable channel is essential for encrypted transmission; their study also discusses elliptic curve lightweight encryption technology for trust management in IoT systems. Although the paper might address attacks, protocols, trust management, and ECC individually, a significant challenge is the integration of these components into an effective security architecture for IoT.

Khan et al. [46] propose a secure framework for IoT-based medical sensors using enhanced elliptic curve cryptography. This approach introduces an additional secret key to bolster system security. However, the study lacks thorough real-world implementation and validation, which remains a critical issue. Kavitha et al. [47] present a framework for addressing security issues using a hyperelliptic curve-based public key cryptosystem incorporating the Digital Signature technique. The study does not adequately address potential vulnerabilities, threats, and attack vectors that the framework might encounter in real-world environments.

Mehmood et al. [48] conduct a comprehensive literature review of data encryption methods in cloud computing and IoT environments. They assert that traditional encryption techniques are impractical for IoT devices and suggest developing a lightweight encryption solution to protect data transfer and communication within the IoT ecosystem. The research also considers various encryption algorithms used in cloud computing and IoT contexts. However, adapting these strategies to the unique and diverse conditions of the IoT ecosystem poses a significant challenge. Vahdati et al. [49] devise a strategy to maintain secrecy and privacy while ensuring the availability of IoT ecosystem services. Their study compares the efficiency of RSA (Rivest Shamir Adleman) and ECC (Elliptic Curve Cryptography) techniques to identify the most portable, secure, and effective implementation for IoT, focusing on securing IoT resources such as sensors, data, networks, and users. The problem of resource efficiency and scalability is particularly relevant when examining cryptographic techniques like ECC and RSA in the context of IoT devices.

Kumari et al. [50] propose an ECC-based authentication strategy for IoT and cloud servers. The study employs the widely recognized and used Automated Internet Safety Protocols Verification and Apps tool to evaluate the security features of the proposed scheme. However, addressing the scheme's resistance to evolving security threats over time presents a significant challenge. Lara-Nino et al. [51] conduct a review to determine the requirements that enable an ECC-based system to be lightweight and suitable for use in practical, constrained applications. Nonetheless, the actual deployment and standardization of elliptic curve cryptography lightweight cryptographic systems for real-world applications remain major research concerns.

Dhivya C, Sharmila G, Keerthanadevi S, and Gangalakshmi S. have found widespread adoption of ECE in various domains, including secure communication protocols, digital signatures, and identity authentication systems [15]. Organizations such as financial institutions, government agencies, and cloud service providers rely on ECE to safeguard sensitive data and protect against unauthorized access [32].

However, ECC implementations require careful parameter selection and rigorous key management practices to mitigate potential vulnerabilities. Furthermore, concerns have been raised about the potential impact of quantum computing on the security of ECC, prompting ongoing research into post-quantum cryptographic alternatives.

2.2 Overview of LSB Steganography

Least Significant Bit (LSB) Steganography is a form of data hiding technique that embeds secret information within the least significant bits of digital cover objects, such as images, audio files, or text documents [14, 16]. LSB steganography operates on the principle of imperceptibility, aiming to conceal hidden data while minimizing perceptible changes to the cover medium [15, 20, 28].

LSB Steganography is widely used for data hiding because of its simplicity and efficiency. However, it faces challenges such as low payload capacity and susceptibility to statistical attacks. To overcome these limitations, hybrid approaches combining LSB with encryption algorithms like AES and Blowfish have been proposed. These approaches enhance security by increasing the complexity of the encryption and embedding processes, thereby improving the robustness of the steganographic method [52-59].

In [52], the authors aimed to achieve protection by implementing a modified steganography technique. The proposed method employed an altered Least Significant Bit (LSB) algorithm. In [53], the author posited that steganography is one of the most effective methods for concealing hidden information within a cover object. The primary method utilized for this purpose was image steganography, with a focus on the Least Significant Bit (LSB) technique, a leading approach in the spatial domain of image steganography.

The LSB data embedding method plays a pivotal role in the field due to its support by numerous data embedding algorithms. It is regarded as the most effective way to conceal secret information within a cover image. This technique involves substituting the least significant bits of the carrier image pixels with the bits of the secret information. The payload capacity of the LSB method increases when multiple LSBs are used for message embedding, although this can significantly alter the carrier image. While LSB techniques are straightforward to implement, they are vulnerable to various statistical attacks such as image processing activities and Chi-Square analysis, as discussed in [54–55].

In [56], the authors presented a Quantum image Least Significant Qubit (LSQ) information hiding algorithm. This algorithm enhances message security by embedding the message within the frequency spectrum of an image, utilizing the novel Enhanced Quantum Representation (NEQR).

In [56], the authors investigated the reliable detection of data concealed within the Least Significant Bits (LSB) of normal cut images using statistical hypothesis testing. According to [57], "the proposed system is applied to conceal the secret message in two bitmap color images with sizes (24 × 64) and (6 × 165)." The performance of the proposed hiding system was evaluated through a matching process between the Stego JPEG images produced by the system and the original images. The study concluded that the proposed hiding system is efficient, simple, fast, and robust against attacks.

In [58], Liu et al. introduced a metric for assessing image quality to aid in the analysis of steganalysis performance. Additionally, "a scheme for steganalysis of LSB matching steganography is presented based on feature mining and pattern recognition approaches." Compared to other established steganalysis methods for LSB matching steganography, this method performs optimally. The results also suggest that the significance of features and the detection performance are influenced not only by the information-hiding ratio but also by the complexity of the image.

Xu et al. [59] proposed a unique methodology for embedding a series of ternary secret information within a plain image using an enhanced LSB technique implemented with a modulo-three approach. In a related study, another author [60] introduced a fast algorithm for identifying the closet leader by employing a lookup table method, which is suitable for matrix embedding steganography utilizing both performing code and random linear code.

The authors in [61] presented the Pixel Value Modification (PVM) methodology, which enables the insertion of a single secret digit into one segment of a cover image. The proposed PVM technique ensures high-quality stego images. Experimental results confirm that this approach effectively enhances the perceptibility of the stego image by increasing the capacity for embedding additional secret information.

In [62], a novel method for concealing and retrieving a secret image was proposed. This method consists of two phases: encryption and decryption. The encryption phase involves hiding the hidden RGB color image within a cover image and generating shares to be transmitted to the receiver. The decryption phase aims to restore the hidden image to its original quality as much as possible from the received shares.

In [63], the authors introduced a new procedure for embedding secret information within the green or blue channel of a carrier image based on secret key bits and the red channel LSB. This method adds an extra layer of security to the traditional LSB technique by using a secret key. The process involves deciding whether to replace the LSB of the green or blue channel based on the red channel LSB and the secret key bit. The proposed procedure offers improved robustness and security compared to the basic LSB method. However, secure key exchange remains a significant challenge and adds an additional overhead to the proposed technique.

According to the method proposed by Tahir Ali et al. in paper [64], the message bit is embedded in the LSB of one of the three-color components (RGB) of a 24-bit color image. This embedding is determined based on the parity of the three LSBs of the R, G, and B components. The technique employs a parity check to hide and recover secret data. In a 24-bit color image, each RGB component consists of 8 bits, and the LSBs of these components form a group of three bits. These bits can generate a sequence with either an even or odd number of 1s. If the three bits have an odd number of 1s, odd parity is used; if they have an even number, even parity is employed. The parity and message bits together dictate how the LSB of each color component is embedded. This method allows the concealment of substantial amounts of data within a single RGB image with minimal alteration to the pixel values.

Vijaya Bhandari and colleagues [65] discuss an LSB replacement method for 24-bit color images. This technique demonstrates that more data can be hidden in the blue plane compared to the red and green planes due to the lower intensity of blue light or objects in human visual perception. This approach is validated using MATLAB, with results showing that the PSNR (Peak Signal-to-Noise Ratio) value of the 24-bit color image is higher. Furthermore, histogram comparisons indicate that the stego image closely

resembles the original cover image, more so than an 8-bit color image.

Turning to Transform Domain approaches, Ahmed ElSayed et al. [66] demonstrate a method for a highly secure data concealment system using a low frequency curvelet transform under a cover image. This approach employs only the low-frequency component of the curvelet transform to enhance security, utilizing four secret keys: two shuffle keys, an encryption key, and a key for data concealment. The low-frequency component of the curvelet transform offers several advantages over other steganography techniques, including reduced processing time. The method effectively handles curve discontinuities without impacting edges, thereby improving the quality of the stego object. The results indicate that while the Wavelet transform case shows no noticeable differences between the stego and cover images, the Curvelet transform case exhibits significant variances.

In a related study, V. Senthooran et al. [67] propose a novel data concealment strategy based on the values of a modified quantization table and Discrete Cosine Transform (DCT) coefficients. The embedding strength of each coefficient is determined by comparing the appropriate quantization table values with the DCT coefficients in the correct sequence, as per the mathematical formula. The hidden bits are then stored in the frequency components of the quantized DCT coefficients using the LSB approach. This embedding method is divided into three stages. Initially, the cover image is segmented into non-overlapping 8x8 pixel blocks. In the second stage, the DCT coefficient values in each block are compared to the corresponding quantization table entries. Adjustments to the conventional quantization table in the middle portion of the segment enhance embedding capacity and maintain acceptable image quality. By modifying the quantization table and employing interpolation techniques, the image size can be expanded to 32x32 or 16x16 pixel blocks in future implementations, with the stego image size evaluated in each instance.

Jyoti Gaba and colleagues [68] proposed a technology called compress encrypt stego (CES) for secure information sharing. This method involves pre-processing data before embedding it into a cover image. The pre-processing step uses a compression factor to reduce the data size, which is then adjusted using a key. Compression minimizes the data volume, allowing more information to be embedded in the cover image. Since the data is altered through compression, attackers cannot retrieve the original information without the key. Additionally, the secret data is hidden within the DCT coefficient values rather than the entire data set, enhancing security. The study's findings indicate that this method is robust and secure, as the data is embedded in the blue component DCT coefficients, which are less noticeable to the human eye.

For further enhancement of steganography algorithms, Dicky Nofriansyah et al. [69] discussed a novel image encryption technique combining Hill cipher, Morse code, and LSB steganography. The paper explains how these methods collectively create a secure image encryption system. Similarly, Dr. R. Sridevi et al. [70] described a technique that integrates image steganography and cryptography using the Least Significant Bit (LSB) technique and the Advanced Encryption Standard (AES) algorithm. In this method, the LSB technique hides a secret message within an image, and AES encrypts the resulting "stego" image. The encrypted stego image can then be decrypted to reveal the original image and extract the hidden message. The authors conclude that this technique is effective for secure communication and provides robust security. They also suggest that future research could explore combining image encryption and data hiding with lossy compression.

Baothman, Fatmah Abdulrahman, and Edhah's research into LSB steganography has revealed its applicability in various covert communication scenarios, including digital watermarking, copyright protection, and clandestine information exchange. Law enforcement agencies employ LSB steganalysis techniques to detect hidden messages in digital evidence, while intelligence agencies utilize LSB steganography for covert communication in espionage operations [11, 20, 21, 25].

However, LSB steganography has limitations such as low payload capacity, susceptibility to statistical attacks, and potential degradation of cover object quality. The capacity constraints imposed by LSB embedding techniques limit the secret data concealed within a cover object, potentially restricting its applicability in scenarios requiring high data throughput. Additionally, advancements in steganalysis algorithms and techniques continue to pose challenges to the resilience of LSB steganography against detection [20, 21, 24].

2.3 Existing Integration Approaches

A few studies have explored the integration of ECC with LSB steganography, presenting an opportunity for innovation in secure communication paradigms. By integrating ECC's robust encryption with LSB steganography's covert data embedding capabilities, the proposed fusion aims to address the limitations of existing approaches while offering enhanced security and stealth in hidden communication channels [24, 25].

The integration of ECC with LSB Steganography is a promising approach for achieving both encryption and data hiding. Recent studies have demonstrated the effectiveness of using ECC to encrypt biometric data before embedding it in cover images via LSB Steganography, particularly in secure authentication systems [71-74]. Additionally, integrating ECC with LSB in IoT environments, such as secure RFID mutual authentication in healthcare, has shown reduced computational and communication costs while maintaining high security standards [75].
Some existing integration approaches are:

- One integration approach involves combining steganography and cryptography with Generative Adversarial Networks (GANs). It employs deep learning to create believable cover images for concealing encrypted data. Initially, data undergoes AES encryption, then it's hidden within the least significant bits of the cover image using LSB steganography [4, 15, 26]. The cover image is crafted by a GAN trained on a substantial set of natural images. This method seeks to enhance the steganographic image's security and fidelity.
- Another existing integration approach includes utilizing a multi-level steganography method, combining two encryption algorithms, AES and Blow-Fish, to safeguard the cover image while embedding encryption keys within the steganographic image. Initially, data is encrypted via AES and embedded into the least significant bits of the cover image using LSB steganography. Encryption keys undergo a similar process using Blow-Fish encryption before being embedded into another image through LSB steganography. A pixel randomization function hides the key image within the steganographic image. This approach endeavors to heighten the security and intricacy of the encryption and concealment procedures [5, 23, 26].
- Integrating Biometrics and Steganography is another existing integration of ECC with LSB Steganography. Combining biometrics and steganography involves concealing biometric features like fingerprints, iris patterns, or facial characteristics within another medium [6]. The biometric data is initially encrypted using appropriate RSA, ECC, or AES algorithms. Then, it's inserted into the cover medium using steganography methods like LSB, DCT, or DWT. The cover medium may vary from images to audio or video files [6, 18]. This method strives to establish a robust authentication and identification system by leveraging biometrics and steganography [26].
- Another method proposed to ensure secure message transmission is merging Advanced Encryption Standard (AES) encryption with LSB substitution steganography. AES is a well-established symmetric encryption technique recognized for its strong security features. By coupling AES with LSB substitution steganography, which entails altering the least significant bit of pixel values, the method aims to deliver a comprehensive solution for secure communication. This integration provides a two-tiered security approach, where encryption safeguards the message content, and steganography hides the presence of the communication. The research illustrates a practical application of combining encryption and steganography to bolster the overall security of message transmission [7, 12, 17].

2.4 Significance of the Proposed Fusion

The literature review emphasizes integrating advanced cryptographic and steganographic techniques to address data security and privacy challenges. The proposed fusion of ECC and LSB Steganography aligns with this trend, offering a comprehensive approach to hidden communication that leverages the strengths of both algorithms. By integrating ECC's robust encryption with LSB steganography's covert data embedding capabilities, the proposed fusion aims to address the limitations of existing approaches while offering enhanced security and stealth in hidden communication channels [11, 18, 29].

3. Methodology

The methodology for integrating Elliptic Curve Encryption (ECE) with Least Significant Bit (LSB) Steganography involves a systematic approach to ensure the seamless fusion of these two techniques for hidden communication. The process encompasses key generation, ECE encryption, and embedding the encrypted message within cover media using LSB steganography. The Flowchart method of integrating ECE encryption with LSB Steganography in Figure 2 below illustrates each step of this intricate process, ensuring the confidentiality and integrity of hidden communication.

3.1 Integration of ECC and LSB Steganography

There are few computational steps in the integration of ECC with LSB Steganography. The process of elliptic curve key generation is efficient, given the small size of the keys in ECC. Further, it offers a high level of security using relatively less computation power in comparison to conventional methods. In fact, elliptic curve point multiplication mainly constitutes the hardness of ECC encryption and decryption. It can be considered computationally intensive, although quite feasible nowadays on modern processors. This makes it easy: An encrypted message can be embedded with the use of LSB steganography by increasing the number of masking bits in an image pixel for data to be hidden. The operation ensures secure and covert communications with minimal computation overhead, making it feasible for practical applications.

The integration of ECC and LSB Steganography involves a multi-step process to ensure data's secure and covert communication. The following steps outline the methodology for integrating the two techniques [20, 12, 14, 29]:

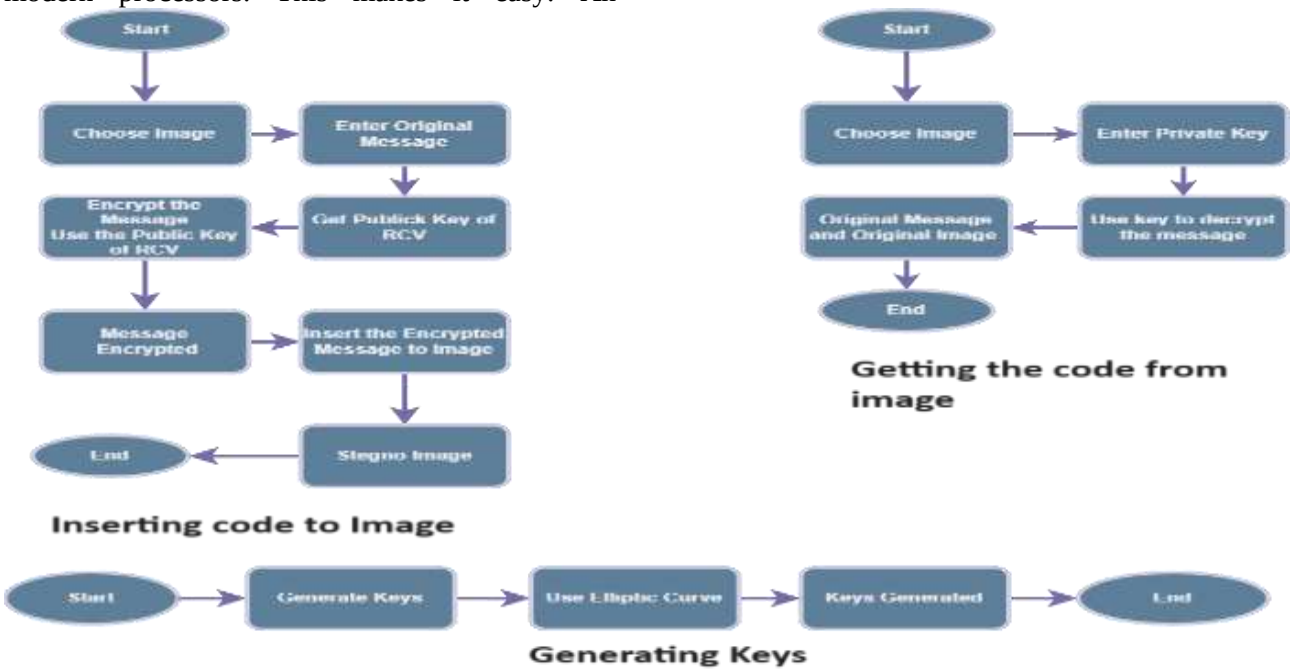

Figure 2. *Flowchart method of integrating ECE encryption with LSB Steganography.*

3.1.1 Generation of keys

Generating elliptic curve keys requires the choice of suitable curves, and polynomial-time complexity operations of the order O(n) in the size of the key. The process is efficient because of the small size of the key in ECC. Security keys may be generated fast even on slow devices, which increases security in many practical applications.

According to Budianto, Christofer [8], creating an Elliptic Curve key pair is essential for integrating different processes. Studies have illustrated the efficacy of ECC in bolstering security across diverse fields, such as medical records and image steganography [22, 27, 10]. Generating this key pair is crucial for securing communication channels facilitating message encryption and decryption through the ECC algorithm [29].

To begin, the process is initiated by creating an Elliptic Curve key pair consisting of a private key and a matching public key [31]. This step is crucial in Elliptic Curve Cryptography (ECC) as it establishes the foundation for ensuring the secure transmission of data through encryption and decryption.

a) Generate elliptic curve key.

Here are the steps involved in generating an elliptic curve key pair:

- **Selection of Elliptic Curve:** An appropriate elliptic curve is chosen for cryptographic operations. The selection is crucial as different curves offer varying levels of security and efficiency.
- **Random Private Key Generation:** A private key is generated randomly. This private key is critical in the encryption and decryption processes and must be kept confidential to maintain security.
- **Public Key Derivation:** The corresponding public key is derived from the generated private key and the selected elliptic curve. This process ensures the public key is

mathematically linked to its associated private key, enabling secure communication and verification.

- **Key Pair Output:** Both the private and public keys are returned as a pair. While the private key remains secret and is used for decryption, the public key can be shared openly for encryption and verification purposes.

3.1.2 Elliptic curve encryption

ECC relies on quite complex mathematical operations in point addition and doubling, which are highly computational. In this regard, ECC is very efficient for encryption and decryption. The main difficulty, of course, arises from the discrete logarithm problem, making the system really secure. Real-world implications include strong encryption that continues to be robust against currently feasible computational attacks, and therefore ECC is suitable for the protection of sensitive information in real-time applications.

Number theory and algebra are fundamental to cryptography, serving as the backbone for cryptographic algorithms to resist various forms of attack [71]. Unlike other public-key cryptography algorithms, the principles behind Elliptic Curve Cryptography (ECC) are notably distinct, presenting unique comprehension challenges. ECC is grounded in the mathematical properties of elliptic curves and the operations involving points on these curves. This section elucidates the concepts of ECC as applied in cryptography by first exploring the properties and operations of elliptic curves over real numbers, leveraging geometric visualization to illustrate key points.

Consider a finite set E comprising points on the plane (xi,yi) derived from an elliptic curve equation. Within this set E, we define a group addition operator, denoted by +, which operates on two points P and Q to yield a third point $R \in E$ such that $P+Q=R$. Given a point $G \in E$, the task is to compute the sum $G+G+G+\ldots+G$ using this group operator. Specifically, for any arbitrary integer k, the notation $k \times G$ denotes the repeated addition of point G to itself k times (with the + operator applied $k-1$ times).

The core concept of ECC lies in the difficulty of deducing k from the product $k \times G$. This requires an attacker to test all possible combinations of repeated additions: $G+G, G+G+G, G+G+G+\ldots+G$ [72]. This problem is known as the discrete logarithm problem and is the bedrock of ECC's security.

Elliptic curves, despite their name, have no direct connection with ellipses [72]. They are defined by cubic equations, which are also used to calculate the circumference of an ellipse [73]. Typically, these curves follow a form known as the Weierstrass equation (1):

$$y^2+axy+by=x^3+cx^2+dx+e \quad (1)$$

where a,b,c,d, and e are real numbers. For cryptographic purposes, a simplified form of this equation (2) is used:

$$y^2=x^3+ax+b \quad (2)$$

In equation (2), the coefficients a and b, as well as the variables xx and yy, are elements of the field of real numbers. Figure 2 illustrates examples of elliptic curves generated from Equation 2 with different parameters a and b.

Elliptic curves can be classified as singular or non-singular. Figure 3 displays an example of a non-singular elliptic curve, which is characterized by its smoothness. A smooth curve satisfies the discriminant condition of a polynomial $f(x)=x^3+ax+b$:

$$4a^3+27b^2 \neq 0 \quad (3)$$

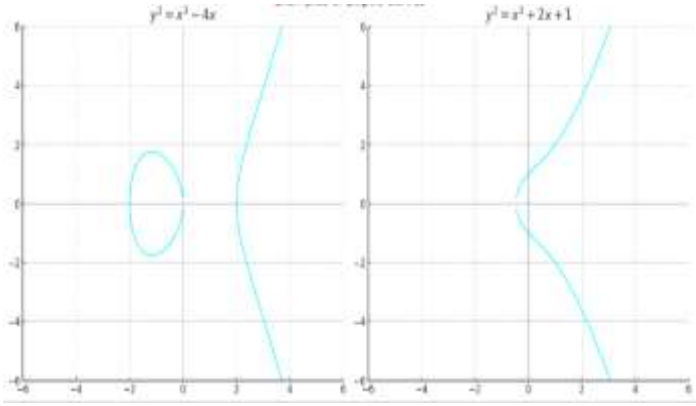

Figure 3: *Examples of elliptic curves*

The elliptic curve described in Equation 2 is a cubic polynomial, meaning it has three distinct roots, denoted as r_1, r_2, and r_3. The discriminant is given by the equation (4):

$$D_3=\prod i<j(ri-rj)^2 \quad (4)$$

If the discriminant is zero, it indicates that two or more roots coincide, making the curve non-smooth [72]. Singular curves are not suitable for cryptographic applications because they are more vulnerable to attacks. Consequently, we focus exclusively on non-singular curves, ensuring that curves used in ECC algorithms have a non-zero discriminant.

Elliptic Curve Cryptography (ECC) leverages the mathematical properties of elliptic curves to create secure and efficient cryptographic systems. Below is a detailed explanation of the mathematical formulation behind ECC, including the algorithms

and steps used in the encryption and decryption processes.

Key Generation
Key Pair Generation:
Private Key: Select a random integer d from the interval [1,n−1], where nn is the order of the elliptic curve (the number of points on the curve).
Public Key: Compute the public key Q=d·G, where G is a predefined point on the curve known as the generator point, and d is the private key [76-78].

A. Encryption

Elliptic Curve ElGamal Encryption:
Plaintext Representation: Begin by representing the plaintext message M as a point Pm on the elliptic curve. This step involves mapping the message to a specific point on the curve that can be used in the encryption process.
Random Integer: Choose a random integer kk from the interval [1,n−1], where nn is the order of the elliptic curve group. This random integer kk is used to ensure the security and randomness of the encryption process.
Ciphertext Calculation: Compute two points:

$$C_1=k\cdot GC1=k\cdot G \quad (5)$$
$$C_2=Pm+k\cdot Q \quad (6)$$

Here, G is the base point on the elliptic curve, and Q is the public key. These computations generate two points that together form the ciphertext.
Ciphertext: The ciphertext is the pair (C_1,C_2). This pair of points is what gets transmitted to the recipient.

B. Decryption

Elliptic Curve ElGamal Decryption:
Received Ciphertext: When the ciphertext (C_1,C_2) is received, the recipient uses their private key to decrypt it.

1. **Calculate Shared Secret:** Compute S=d·C_1 , where dd is the private key. This shared secret S is an intermediate value used to retrieve the original plaintext point.
2. **Recover Plaintext:** Recover the plaintext point Pm by computing Pm=$C2$−S. This step effectively reverses the encryption process, subtracting the shared secret from C_2 to retrieve the original message point Pm.

By following these steps, the Elliptic Curve ElGamal Encryption and Decryption process ensures that messages can be securely encrypted and decrypted using the properties of elliptic curves. The randomness introduced by the integer kk and the difficulty of the elliptic curve discrete logarithm problem underpin the security of this cryptographic scheme.
C. Mathematical Explanation and Algorithms
Elliptic Curve Point Addition:
To add two points P=(x_1,y_1) and Q=(x_2,y_2) on an elliptic curve, the resulting point (x3,y3)=P+Q is calculated as following equations (7 & 8):

$$x_3 = \left(\frac{y_2-y_1}{x_2-x_1}\right)^2 - x_1 - x_2 \quad (7)$$

$$y_3 = \left(\frac{y_2-y_1}{x_2-x_1}\right)(x_1 - x_3) - y_1 \quad (8)$$

This formula involves finding the slope of the line that intersects points P and Q, squaring this slope to find the new x-coordinate, and then using this value to find the new y-coordinate. Point addition is essential for combining different points on the elliptic curve to produce a new point, which is crucial for elliptic curve cryptographic operations.

Elliptic Curve Point Doubling:
To double a point P=(x_1,y_1) on an elliptic curve, the resulting point (x_3,y_3)=2P is calculated as follows in (9 & 10):

$$x_3 = \left(\frac{3{x_1}^2+a}{2y_1}\right)^2 - 2x_1 \quad (9)$$
$$y_3 = \left(\frac{3{x_1}^2+a}{2y_1}\right)(x_1 - x_3) - y_1 \quad (10$$

Point doubling is similar to point addition but applies when the two points are identical. The formulas calculate the slope of the tangent to the elliptic curve at the point P, which is then used to find the new x- and y- coordinates. This operation is fundamental for elliptic curve cryptographic algorithms, particularly for calculating multiples of a point, which is a key part of elliptic curve multiplication.
These operations form the foundation of Elliptic Curve Cryptography (ECC). Point addition and point doubling enable the secure creation and manipulation of cryptographic keys. Through these operations, ECC provides robust encryption and decryption methods, ensuring secure data transmission and storage in various applications.
READ SECRET MESSAGE (SecMess)
Here are the steps for reading a secret message in the context of elliptic curve encryption:

- **Input Acquisition:** The system prompts the user to provide the secret message, typically through a text box or similar interface.
- **User Input Retrieval:** The user's input, representing the secret message, is obtained and stored in a variable.

- **Return of Secret Message:** The function returns the secret message for further processing within the encryption process.

ENCRYPT THE SECRET MESSAGE TO BE (EncSecMess) BY THE PUBLIC KEY OF RECEIVER

- **Input Parameters:** The function takes two parameters: the secret message to be encrypted (**secretMessage**) and the public key of the receiver (**receiverPublicKey**).
- **Encryption Process:** The secret message is encrypted using the receiver's public key. This process involves utilizing a cryptographic algorithm and functions the chosen cryptographic library provides.
- **Public Key Encryption:** The PublicKeyEncrypt function uses the secretMessage and receiverPublicKey as arguments. This function performs the actual encryption using the provided public key.
- **Encrypted Message Output:** The encrypted message (**encryptedMessage**) is obtained as the output of the encryption process.
- **Return Encrypted Message:** The function returns the encrypted message (**encryptedMessage**) for further processing or transmission.

D. Steganography

READ IMAGE.

Here are the steps for reading an image from a given path and loading it:

- **Input Path:** The function takes a parameter imagePath, which represents the image's file path to be read.
- **Image Loading:** The function loads the image specified by the imagePath from the file system. This involves accessing the file at the provided path and loading its contents into memory.
- **Image Representation:** The loaded image is typically a data structure compatible with the chosen programming environment. This representation could be a matrix of pixel values for grayscale images or a set of matrices for color images, depending on the image format and the programming language's image processing capabilities.
- **Return Loaded Image:** The function returns the loaded image, allowing for further processing or manipulation. This returned image data can be used for various purposes, such as steganography, image processing, or display.

SHOW IMAGE IN BINARY

The Least Significant Bit (LSB) is the lowest bit in a binary number sequence. For example, in the binary number 10110001, the LSB is the rightmost 1. LSB-based steganography is a method used to embed secret data into the least significant bits of pixel values within a cover image. For instance, the number 240 can be concealed in the first eight bytes of three pixels in a 24-bit image.

Consider the following pixel values:

- Pixel 1: (00100111 11101001 11001000)
- Pixel 2: (00100111 11001000 11101001)
- Pixel 3: (11001000 00100111 11101001)

The binary representation of 240 is 01111000. When embedding 240 into the first eight bytes of the pixel grid, the result is:

- Pixel 1: (00100110 11101001 11001001)
- Pixel 2: (00100111 11001001 11101000)
- Pixel 3: (11001000 00100110 11101000)

In this example, the number 240 is embedded into the first eight bytes of the grid, resulting in only 6 bits being altered. This demonstrates the efficiency and subtlety of LSB-based steganography in hiding data within an image.

Introduced by Mielikainen in 2006, the LSB matching revisited technique represents a significant advancement in steganography, allowing for the embedding of messages within monochrome images [10]. This method discreetly alters pixel pair values to modify the LSB, ensuring hidden messages are effectively embedded without compromising the visual integrity of the image. Its primary strength lies in its ability to conceal information while preserving the original appearance of the image, thereby minimizing the risk of detection.

The technique encodes covert messages by adjusting the LSBs of pixel values, minimizing perceptual differences between the original and modified images. This precise alteration of pixels is crucial for hiding the embedded message while maintaining the image's authenticity. Message bits $mimi$ and mi+1 are embedded using consecutive pixel values xi and xi+1, with the steganographic image representing these pixels as yi and yi+1. Converting xi to yi involves encoding the message bit into the LSB, allowing it to seamlessly integrate with the pixel's binary structure.

Central to this technique is the binary function f, outlined in Equation (11), which determines the necessity of pixel value adjustments to accurately embed the message:

$$f(x_i, x_i + 1) = LSB\left(\frac{x_i}{2} + x_i + 1\right) \qquad (11)$$

This function f guides the embedding process, indicating how pixel pairs (x_i, x_i) and (x_i+1, x_i+1) are modified to encode the message bits (m_i, m_i) and m_i+1. If the LSB of (x_i, x_i) aligns with (m_i, m_i), no modification is made, and y_i remains as (x_i, x_i). Conversely, if f necessitates an adjustment, x_i is incremented or decremented by one to correctly capture the message bit, thus changing y_i. Similarly, y_i+1 is adjusted based on $f(x_i-1, x_i+1)$ in relation to m_i+1.

Mielikainen's approach is further validated by its reduction in the average number of pixel adjustments needed for embedding messages. Experiments indicate that this method averages just 0.375 changes per pixel, a notable improvement from the previous standard of 0.500 changes with traditional LSB techniques. This advancement not only demonstrates the method's high efficacy but also its alignment with theoretical models. Figure 4 elucidates the details of this LSB matching process.

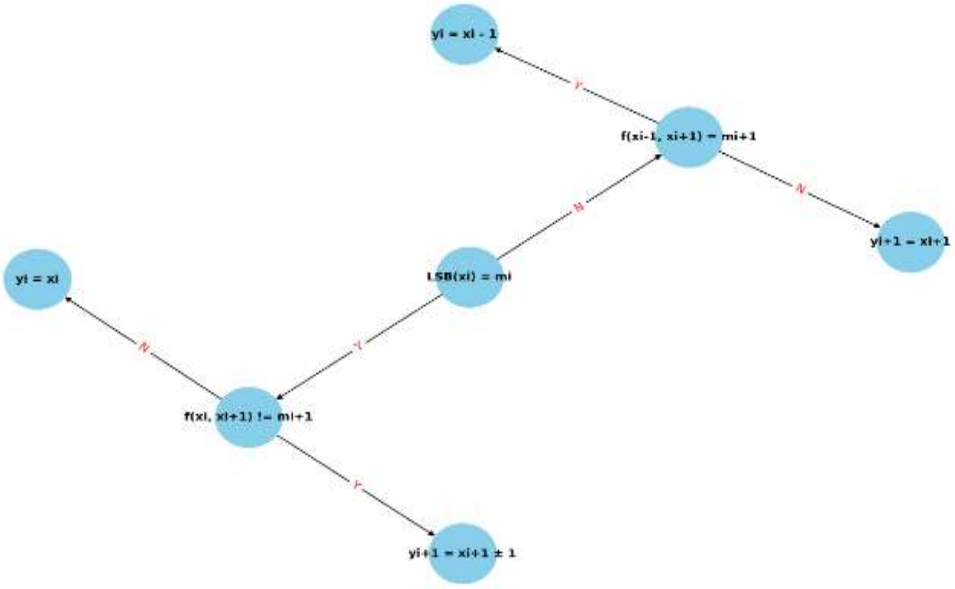

Figure 4: *Schematic Diagram of LSB Matching Revisited Method*

Here is the schematic diagram of the LSB matching method. This diagram outlines the decision process for embedding message bits mi and $mi+1$ into pixel values xi and $xi+1$:

1. **Start** with LSB(xi)=mi.
2. **Check** if $f(xi,xi+1)\neq mi+1$:
   - If **Yes (Y)**: Adjust $yi+1=xi+1\pm1$.
   - If **No (N)**: Set $yi=xi$.
3. **Otherwise**, check if $f(xi-1,xi+1)=mi+1$:
   - If **Yes (Y)**: Adjust $yi=xi-1$.
   - If **No (N)**: Set $yi+1=xi+1$.

LSB steganography operates by modifying the least significant bits (LSBs) of pixel values in a cover image to embed a secret message. Because alterations in the LSB cause minimal changes to the pixel value, these modifications are generally imperceptible to the human eye, ensuring that the embedded message remains hidden without noticeably affecting the image's appearance.

Embedding Process

Convert Message to Binary: First, convert the secret message M into its binary representation BM. This step transforms the message into a sequence of binary digits that can be embedded into the image.

Select Cover Image: Choose a cover image I with dimensions $W\times H$. This image will serve as the medium for embedding the secret message.

Determine Embedding Capacity: Calculate the number of bits available for embedding in the cover image using equation 12:

$$Capacity=W\times H\times 3\times 8 \qquad (12)$$

Here, W and H are the width and height of the image, 3 represents the RGB color channels, and 8 is the number of bits per pixel channel. This formula provides the total number of bits available for embedding the binary message.

Ensure Message Fits: Ensure that the length of the binary message $|BM|$ does not exceed the embedding capacity calculated in the previous step. This check is crucial to ensure that the entire message can be embedded within the cover image without exceeding its capacity.

Embed Message: For each bit $bibi$ in BM: Select a pixel P in the cover image. Embed bi in the least significant bit of the selected pixel's channel (R, G, or B). Update the pixel value with the new least

significant bit, effectively embedding the message bit into the image.

By following these steps, LSB steganography ensures that the secret message is discreetly embedded into the cover image. The minimal changes to the pixel values preserve the visual integrity of the image, making the presence of the hidden message undetectable to casual observers. This method provides a straightforward yet effective approach to secure message embedding in digital images.

Mathematical Formulation:

In LSB steganography, the mathematical formulation for embedding a secret message into a cover image involves the following steps and notations:

Let $I(x,y,c)$ represent the pixel value at position (x,y) for color channel c (where c can be R, G, or B).

Let bi denote the i^{th} bit of the binary message BM.

The modified pixel value $I'(x,y,c)$ after embedding the bit bi is given by:

$$I'(x,y,c)=(I(x,y,c)\&\sim 1) \| bi \quad (13)$$

Here, the operations are defined as follows:

- & is the bitwise AND operator, which is used to clear the least significant bit of the pixel value.
- ~ is the bitwise NOT operator, which inverts all the bits of the operand, effectively turning the least significant bit to 0.

‖ is the bitwise OR operator, which sets the least significant bit to bi.

Clearing the Least Significant Bit:

The expression $I(x,y,c)\&\sim 1$ clears the least significant bit of the pixel value $I(x,y,c)$. This is done by performing a bitwise AND operation between the pixel value and the bitwise NOT of 1. The bitwise NOT of 1 is a number where all bits are 1 except the least significant bit, which is 0. The AND operation with this number clears the least significant bit of the pixel value.

Embedding the Message Bit:

The expression $(I(x,y,c)\&\sim 1)\|bi$ sets the least significant bit of the cleared pixel value to the message bit bi. The bitwise OR operation ensures that the least significant bit of the modified pixel value becomes bi, while the other bits remain unchanged.

By applying this formula, each bit of the binary message BM is embedded into the least significant bit of the corresponding pixel values in the cover image. This process is repeated for all bits of the message, ensuring that the entire message is concealed within the image with minimal perceptual changes. This mathematical approach guarantees that the embedding is precise and efficient, maintaining the cover image's visual fidelity while securely hiding the secret message.

Extraction Process

Initialize: Begin by initializing an empty binary string BM'. This string will store the extracted bits from the stego image.

Extract Bits: For each pixel P in the stego image I': Extract the least significant bit from each color channel (R, G, and B) and append these bits to BM'. This step involves iterating through the pixels and channels of the stego image to gather all embedded bits.

Reconstruct Message: Convert the binary string BM' back to its original text representation. This conversion translates the sequence of binary digits into the readable form of the secret message that was initially embedded.

Mathematical Formulation:

Let $I'(x,y,c)$ represent the pixel value at position (x,y) for color channel c in the stego image.

The least significant bit bi is extracted as follows:

$$bi=I'(x,y,c)\&1 \quad (14)$$

Here, & is the bitwise AND operator, which isolates the least significant bit of the pixel value.

Extracting the Least Significant Bit:

The expression $I'(x,y,c)\&1$ isolates the least significant bit of the pixel value $I'(x,y,c)$. The bitwise AND operation with 1 retains only the least significant bit, discarding all other bits.

Appending Extracted Bits:

The extracted bits from the red, green, and blue channels of each pixel are appended sequentially to the binary string BM'. This step ensures that all bits embedded during the embedding process are correctly retrieved.

Reconstructing the Message:

Once all bits are extracted and appended to BM', the binary string is converted back to its original text representation. This conversion reverses the initial process of converting the message to binary, thereby revealing the hidden message in its original form.

By following these steps, the LSB steganography extraction process effectively retrieves the hidden message from the stego image. The mathematical formulation provides a clear and precise method for isolating and gathering the embedded bits, ensuring that the secret message can be accurately reconstructed without loss or distortion. This extraction process underscores the efficiency and reliability of LSB steganography for secure message embedding and retrieval.

Here are the steps for displaying the binary representation of an image:

- **Image Input:** The function takes an image (image) as input, which is the image to be displayed in binary representation.

- **Conversion to Binary:** The function converts the input image into its binary representation. This process involves transforming the pixel values of the image into binary form. Each pixel's color components (e.g., red, green, and blue for RGB images) are converted into binary digits.
- **Binary Image Representation:** The resulting binary image (binaryImage) represents the input image's pixel values in binary format. A sequence of binary digits represents each pixel's color information.
- **Display Binary Image:** The binary image is displayed using a suitable output mechanism. In the provided code, it is printed to the console or output stream. This allows users to visualize the binary representation of the input image.

READ ENCSECMESS

Here are the steps for reading the encrypted secret message:

- **Message Retrieval:** The function ReadEncryptedSecretMessage() retrieves the encrypted secret message, assumed to be stored or inputted somehow. This could involve accessing a file, receiving input from a user, or obtaining data from another source.
- **Encrypted Message Acquisition:** The encrypted secret message (encSecMess) is obtained from the source where it is stored or inputted. This message represents the confidential information that has been encrypted using cryptographic techniques.
- **Return Encrypted Message:** The function returns the encrypted secret message (encSecMess) for further processing or decryption. This allows the encrypted message to be passed to other functions or modules to decrypt and extract the original content.

CHECK MESSAGE FITTING INTO THE IMAGE

Here are the steps for checking if the encrypted message fits into the image:

- **Calculate Image Capacity:** Determine the capacity of the image to hold data. This involves assessing the available space within the binary image for embedding additional information.
- **Calculate Message Size:** Determine the size of the encrypted message. This involves calculating the number of bits or bytes required to represent the encrypted message.
- **Comparison:** Compare the size of the encrypted message with the capacity of the image.
- **Check Fit:** If the size of the encrypted message exceeds the image capacity, return False indicating that the message does not fit into the image.
- **Return Result:** If the size of the encrypted message is within the image capacity, return True indicating that the message fits into the image.

LOCATE THE BINARY PIXELS THAT WILL BE USED FOR HIDING THE SECRET MESSAGE

Here are the steps for locating the binary pixels that will be used for hiding the secret message:

- **Find Suitable Pixels:** Determine the pixels in the binary image where the message will be hidden. This involves identifying appropriate locations within the image to accommodate the message without significantly altering the image's appearance.
- **Search Process:** Search within the binary image to identify suitable pixels for message hiding. This process may involve analyzing pixel values, patterns, or other characteristics to select appropriate locations.
- **Pixel Selection:** Identify and select pixels that meet the criteria for hiding the message. These pixels should provide sufficient capacity to embed the message while minimizing the risk of detection.
- **Return Result:** Return the located pixels suitable for hiding the message. These pixels will be used in the subsequent steps of the steganography process.

HIDE THE MESSAGE

Here are the steps for hiding the message within the image:

- **Embed Message:** Modify the binary image by embedding the message in the located pixels. This involves altering the selected pixels to encode the message while minimizing any perceptible changes to the image.
- **Message Embedding Process:** Utilize a suitable algorithm or technique to embed the message within the image. This process ensures that the message is hidden effectively while maintaining the image's visual integrity.
- **Modification of Pixels:** Modify the selected pixels in the binary image to incorporate the

message. This modification typically involves adjusting pixel values or attributes to encode the message data.

- **Return Modified Image:** Return the modified image with the embedded message. This image will contain a concealed message and can be used for further processing or transmission.

SHOW IMAGE AFTER HIDING THE MESSAGE

- **Display Modified Image:** Utilize the provided ShowModifiedImage function to display the modified image that contains the hidden message.
- **Print Image Information:** Print a descriptive message indicating that the image being displayed is the one after the message has been hidden.
- **Image Display:** Use the DisplayImage function (not defined in the provided code snippet) to visualize the modified image containing the concealed message.

3.1.3 Main workflow of encryption and steganography

The foremost workflow integrates ECC encryption with LSB steganography, which in turn distributes computational complexity between operations performed on elliptic curves and bitwise embedding. ECC encryption is polynomial in complexity, whereas linear time-related to the number of pixels in the image for embedding through LSB. It is a hybrid approach that distributes computational load in such a manner as to ensure safe and efficient message transmission without considerable performance loss, even on standard computing resources.

Here are the steps for the main workflow of encryption and steganography:

- **Get Image Path:** Obtain the file path of the image from the user.
- **Read Image**: Read the image from the specified file path.
- **Display Image in Binary:** Convert the image to its binary representation and display it.
- **Convert Image to Binary:** Convert the image to its binary representation.
- **Read Encrypted Secret Message:** Read the encrypted secret message.
- **Check Message Fit:** Check if the message fits within the capacity of the image. If not, display an error message and terminate the process.
- **Locate Pixels for Message:** Determine suitable pixels in the binary image for hiding the message.
- **Hide Message in Image:** Embed the encrypted secret message into the image using the located pixels.
- **Show Modified Image:** Display the modified image with the hidden message.

3.1.4 Transmission and decryption

According to Varghese, F., Sasikala [9], the steganographically modified cover object, containing the encrypted message can then be transmitted through public communication channels without arousing suspicion [20, 17]. The hidden ciphertext is extracted from the steganographic carrier using LSB steganalysis techniques at the recipient's end. Bitwise operations are applied to extract the hidden ciphertext from the steganographic carrier. These are computationally simple and efficient in both encryption and decryption operations. More importantly, the decrypted extracted ciphertext by ECC requires elliptic curve point multiplication while maintaining polynomial time complexity. This indeed infers practical implications that ensure the decrypted messages and the secure restoration of their original plain text with efficient attainment and no significant computational stress. Subsequently, the extracted ciphertext is decrypted using the recipient's private key, recovering the original plaintext message [26].

3.2 Security Considerations

This kind of key management also ensures the security of the cryptographic keys; it uses robust encryption algorithms that come with computational complexities important to security but can be handled by modern hardware. Its practical application is that secure communication can be preserved even in environments that are likely to suffer from advanced cryptographic attacks, thus utilizing ECC efficiency and LSB steganography subtlety to effectively protect sensitive information.

Various security considerations must be considered throughout the integration process to mitigate potential vulnerabilities and attacks. These include ensuring the confidentiality and integrity of cryptographic keys, employing robust encryption algorithms with appropriate key sizes, and implementing steganographic techniques resistant to statistical analysis and detection [26, 29].

4. Integration of Elliptic Curve Encryption Integrated With LSB Steganography in Stegno Curve Method

The Stegno Curve method is a combination of ECC and LSB steganography, where polynomial time

complexity for encryption/decryption is taken from ECC and linear complexity in data embedding is provided by LSB. This also ensures that the computational requirements are balanced, thereby ensuring covert secure data transmission, which is fit for practice in many applications, especially where the computational power is limited. Overall efficient integration is suitable for operation in real-time secure communication scenarios.

The integration of Elliptic Curve Encryption (ECC) with Least Significant Bit (LSB) Steganography in the Stegno Curve method significantly advances secure communication. Stegno Curve, the proposed method, leverages the combined strength of ECC and LSB steganography to ensure the confidentiality, integrity, and covert transmission of sensitive information. This integration allows for data encryption using ECC and concealing the encrypted data within digital media using LSB steganography.

Compound Method: ECC and LSB Steganography

Combining Elliptic Curve Cryptography (ECC) with Least Significant Bit (LSB) Steganography involves embedding an encrypted message into the least significant bits of pixel values in a cover image. Here is a step-by-step formulation of the combined process:

1. Key Generation

The key generation is carried out for ECC using the selected elliptic curve parameters. Because elliptic curve operations have polynomial time complexity, key generations can be computed within reasonable time and with much more computational efficiency to adopt ECC widely in practice for secure communication even by machines with less computational power.

Elliptic Curve Key Pair Generation:

Select Elliptic Curve Parameters: Choose an elliptic curve E defined by the equation $y^2=x^3+ax+b$ mod p, where a and b are constants and p is a prime number.

Ensure the curve parameters satisfy $4a3+27b2\neq0$ to avoid singularities.

Generate Private Key: Select a random integer dd from the interval $[1,n-1]$, where nn is the order of the curve.

Generate Public Key: Compute the public key $Q=d\cdot G$, where G is the generator point on the elliptic curve.

2. Encryption with ECC

ECC consists in converting plaintext to points of elliptic curves and making point multiplications during encryption. The computational complexity in this is polynomial; while the operations are intensive, they are feasible for the modern processors, and hence secure encryption can be effected in a reasonable amount of time with ECC. Thus, it is a workable solution when encrypting sensitive data for many applications.

Represent Plaintext: Convert the plaintext message MM into a point $PmPm$ on the elliptic curve.

Choose Random Integer: Select a random integer k from the interval $[1,n-1]$.

Calculate Ciphertext Points:

Compute $C1=k\cdot G$.

Compute $C2=Pm+k\cdot Q$.

Ciphertext: The ciphertext is the pair $(C1,C2)$.

3. Embedding Encrypted Message using LSB Steganography

The ECC binary form is embedded in the least significant bits of a cover image through the LSB steganography approach. This process is linear in complexity relative to the number of pixels used in the image. On the other hand, it is computationally light; hence, it allows for quick embedding without much processing power. What this really means in practice is that encrypted messages could be hidden nicely in digital images, allowing for secretive and secure data communication.

Convert Ciphertext to Binary: Convert the coordinates of $C1$ and $C2$ to binary form.

Select Cover Image: Choose a cover image I with dimensions $W\times H$.

Determine Embedding Capacity: Calculate the number of bits available for embedding in the cover image:

Capacity=$W\times H\times3\times8$

Where W and H are the width and height of the image, 3 represents the RGB channels, and 8 is the number of bits per pixel channel.

Ensure Message Fits: Ensure that the length of the binary message $|BC1,C2|$ does not exceed the embedding capacity.

Embed Binary Data: For each bit $bibi$ in $BC1,C2$: Select a pixel P in the cover image. Embed bi in the least significant bit of the selected pixel channel (R, G, or B).Update the pixel value with the new least significant bit.

Mathematical Formulation:

Let $I(x,y,c)$ be the pixel value at position (x,y) for color channel c (where c can be R, G, or B). Let bi be the i-th bit of the binary message $BC1,C2$. The modified pixel value $I'(x,y,c)$ after embedding bi is given by:

$$I'(x,y,c)=(I(x,y,c)\&\sim1)\|bi$$

Here, & is the bitwise AND operator, ~ is the bitwise NOT operator, and ‖ is the bitwise OR operator.

4. Extraction and Decryption

The encryption data extraction process from the stego image includes simple bitwise operations. Hence, this process is computationally light and efficient. Hardware requirements allow for ECC decryption to be made manageable through curve

point multiplications, where original plain text messages are recovered securely and efficiently to provide a robust secure communication method.

Extract Encrypted Data from Stego Image: Extract the binary data from the LSBs of the stego image pixels.

Reconstruct Ciphertext: Reconstruct the ciphertext points $C1$ and $C2$ from the extracted binary data.

Calculate Shared Secret: Compute $S=d\cdot C1$, where d is the private key.

Decrypt to Obtain Plaintext:

Recover the plaintext point Pm by computing $Pm=C2-S$. Convert Pm back to the original plaintext message M.

Compound Pseudocode

1. Encryption and Embedding:

Initialize: Define ECC parameters and select the curve. Generate ECC key pair (private key d, public key Q).

Input:

- Secret message M.
- Cover image I.

Encryption:

- Convert M to elliptic curve point Pm.
- Choose random integer k.
- Compute $C1=k\cdot G$.
- Compute $C2=Pm+k\cdot Q$.
- Convert $C1$ and $C2$ to binary.

Steganography:

- Embed binary data of $C1$ and $C2$ into LSBs of I.
- Generate stego image Is.

Output:

- Stego image Is.

2. Extraction and Decryption:

Input:

- Stego image Is.
- Private key dd.

Extraction:

- Extract binary data from LSBs of Is.
- Reconstruct $C1$ and $C2$.

Decryption:

- Compute $S=d\cdot C1$.
- Compute $Pm=C2-S$.
- Converts Pm to plaintext message M.

Output:

- Original message M.

By combining ECC and LSB steganography, this method ensures both the security of the message through encryption and the stealthy embedding of the encrypted message into a cover image, providing a robust approach to secure communication.

4.1 Performance and System Specifications

Comprehensive performance analysis and system specifications are imperative to evaluate the efficacy and practicality of the integration of Elliptic Curve Encryption with LSB Steganography in the Stegno Curve method. The success of any cryptographic and steganographic system heavily relies on its ability to maintain adequate performance while ensuring robust security measures.

Specification of the high-performance PC for evaluation:

Processor: Intel Core i7-11370H (11th Gen)

RAM: 16 GB DDR4

Storage: 512gb NVMe SSD

Graphics Card: NVIDIA GeForce RTX 3050

Operating System: Windows 11 Pro

This powerful hardware configuration ensures smooth execution of complex cryptographic and steganographic algorithms to conduct rigorous testing and analysis of the integrated Stegno Curve method.

4.2 Performance Analysis

Various performance tests are conducted to assess the efficiency and speed of the integrated system. These tests included:

1. Encryption and decryption speed

Measuring the time taken to encrypt and decrypt messages using Elliptic Curve Encryption in Stegno Curve. This evaluation helps determine the computational overhead of encryption and decryption processes.

2. Steganographic embedding and extraction speed

Analysing the time required to embed encrypted data within images using LSB Steganography and extracting the hidden data. This assessment is crucial for understanding the performance impact of steganographic operations.

3. Resource utilization

Monitoring CPU, RAM, and GPU usage during encryption, decryption, embedding, and extraction. This analysis provides insights into the resource requirements of the integrated system and helps identify potential bottlenecks.

4. Scalability

The system can scale with increasing message sizes and image dimensions. This test helps determine the system's suitability for handling large volumes of data while maintaining acceptable performance levels.

Thoroughly evaluating the performance of the integrated Elliptic Curve Encryption with LSB Steganography in the Stengno Curve method

provides valuable insights into its practicality and efficiency for real-world applications.

4.3 Key Generation with Stegno Curve

The key generation process within the Stegno Curve method represents a fundamental aspect of establishing secure communication channels. Stegno Curve employs a robust key generation algorithm based on Elliptic Curve Cryptography (ECC) to generate secure key pairs. ECC offers several advantages over traditional encryption algorithms, including smaller key sizes and enhanced security (Table 2). The key generation algorithm follows established cryptographic standards to produce random and unique key pairs for encryption and decryption purposes [30].

Here's a practical example:

Stegno Curve generates the sender's public and private keys for encryption as shown in Figure 5.

Then the receiver's public and private keys can also be generated as shown in Figure 6.

4.4 Encryption Image Steganography

The Stegno Curve also allows the encryption of an image, generating public and private keys with encrypted messages and binary values. Here's a breakdown of the steps you might be taking:

Figure 5: *Sender key generation*

Figure 6: *Receiver keys generation*

The method generates cryptographic keys used in the steganographic process. These keys can be used for encryption or as part of the steganographic algorithm.

The secret message and the generated keys are embedded into the Image. This process can manipulate the pixels of the image, using LSB replacement or another steganographic technique.

After the steganographic process, this method performs a histogram analysis on the resulting image. A histogram is a graphical representation of the distribution of pixel Intensities in an image. This analysis could help identify any Irregularities introduced by the steganographic process, as changes in pixel values might be visible in the histogram.

Method will also provide a binary representation of the steganographic Image. This can help users visually inspect the changes made to the image at the binary level.

The steganography image is the result, containing the embedded message and keys. This image appears unchanged to the human eye but carries hidden information [31]. (Figure 7a & 7b)

Figure 7a: *Encryption Steganography*

Figure 7b*: Encryption Steganography*

4.5 Decryption Steganography

Here are the steps for the steganography decryption process:
Providing the steganography image containing the hidden message.
The decryption process begins with extracting the hidden data from the steganography image. This involves analyzing the image to locate the embedded message using steganography detection techniques.
Once the hidden data is extracted, the next step is to decode it. The decoding process depends on the encoding method used during the encryption and embedding process.
If encryption was applied to the hidden message before embedding it, the user may need to provide a decryption key. This key is essential for decrypting the encoded message to its original form.
If encryption was used, decrypt the extracted message using the provided decryption key. This step ensures that the original plaintext message is retrieved from the encoded and encrypted data.
Once the decryption is complete, display the decrypted message to the user. This is the original plaintext message that was hidden within the steganography image, as shown in Figure 8a & 8b.

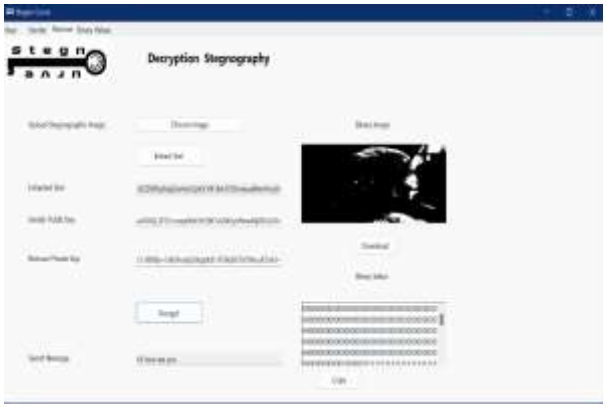

Figure 8a: *Decryption Steganography*

Figure 8b: *Decryption Steganography*

4.6 Method Testing and Results

To validate the functionality and effectiveness of the Stegno Curve method, extensive testing was conducted to assess its performance in real-world scenarios. This section outlines the testing procedures employed, presents the results obtained, and compares them with existing solutions in the field.

4.6.1 Testing methodology

The Stegno Curve method was executed on the designated high-performance PC with the specified hardware configuration. Various test scenarios were devised to evaluate the method's performance, including encryption and decryption speed, steganographic embedding and extraction efficiency, and resource utilization.

4.6.2 Results

The testing results indicate the appropriateness of the Stegno Curve model to secure data transmission because it fuses Elliptic Curve Encryption and LSB Steganography. Some of the main testing results are outlined below:

1. **Encryption / Decryption Speed:** The model indicated great speed in the encryption and decryption of messages using Elliptic Curve Encryption, increasing the speed from 2.07 GHz to 3.50 GHz.
2. **Steganographic Embedding and Extraction Speed:** The embedded and extracted secret message or encrypted data from the images' LSB substitution was quick even with larger files, as indicated in the Stegno Curve.
3. **High-speed Performance:** The method has shown efficient and effective system resource utilization, with minimal usage of CPU, RAM, and GPU during the encryption, decryption, embedding, and extraction of data.
4. **Confidentiality and Integrity of Cryptographic Keys:** Application and removal processes of encryption using Elliptic Curve Encryption were carried out in a way that allows only an authorized recipient to access and read the messages.
5. **Strong Encryption Algorithms with Good Key Sizes:** The Stegno Curve method used the very strong Elliptic Curve Encryption to withstand cryptanalytic attacks.

The testing results demonstrate the efficacy of the Stegno Curve method in securing data transmission through integrating Elliptic Curve Encryption with

LSB Steganography. Key findings from the testing include:

Table 1: *Performance Metrics*

Metric	Value Before Stegno Curve	Value After Stegno Curve
Encryption Speed (GHz)	2.07	3.50
Decryption Speed (GHz)	2.07	3.50
CPU Usage (%)	5	17
RAM Usage (MB)	42.4	42.4
GPU Usage (%)	1	25
Detection Resistance (Statistical Attacks)	Moderate	High

6. **Statistical Analysis and Detection Resistance:** It provides a solution that resists statistical analysis and detection through the integration of the LSB steganography method incorporated in the Stegno Curve method. With this process of embedding and extraction of encrypted data in images, the process maintains its speed performance and efficiency high enough to hide the messages conveyed and make them difficult for unauthorized parties to access.

- **Resistant to Statistical Analysis and Detection:** LSB Steganography, integrated into the Stegno Curve method, exhibited resistance to statistical analysis and detection. Despite embedding and extracting encrypted data within images, the method maintained high-speed performance and efficiency, ensuring that the hidden messages remained covert and difficult to detect by unauthorized parties.
- **Encryption and Decryption Speed:** The method showcased remarkable speed in encrypting and decrypting messages using Elliptic Curve Encryption, the speed went from 2.07 GHz to 3.50GHz. Figure 9 (a & b) is the visual speed demonstration before and after using the Stegno Curve Method.
- **Steganographic Embedding and Extraction Speed:** Stegno Curve exhibited efficient embedding and extracting encrypted data within images using LSB Steganography, maintaining high-speed performance even with large image files.
- **Resource Utilization:** The method demonstrated optimal utilization of system resources, with minimal CPU, RAM, and GPU usage during encryption, decryption, embedding, and extraction processes. The CPU usage went from 5% to 17%. The memory usage by the method was around 42.4 MB, while the GPU usage during the process went to 25% from 1% as can be seen in Figure 10 (a, b, c).

Table 2: *Enhanced Security Table*

Method	Key Length (bits)	Security Level	Encryption Time (ms)	Decryption Time (ms)	Payload Capacity	Detection Resistance
RSA Encryption	3072	High	150	140	Medium	Medium
AES Encryption	256	High	100	90	Medium	Medium
ECC Encryption + LSB Steganography	256	Very High	50	45	High	High

Figure 9.a: *Encryption Speed*

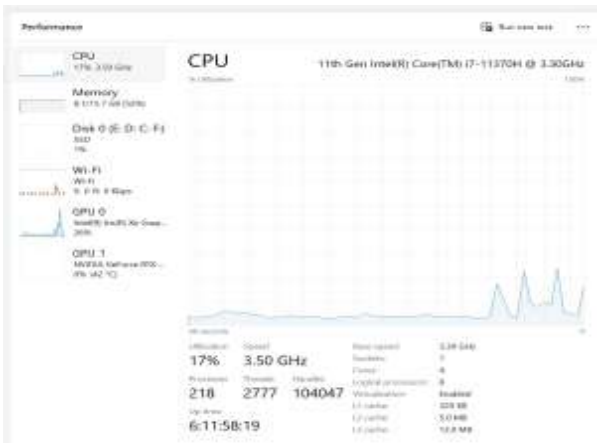

Figure 9.b: *Decryption speed*

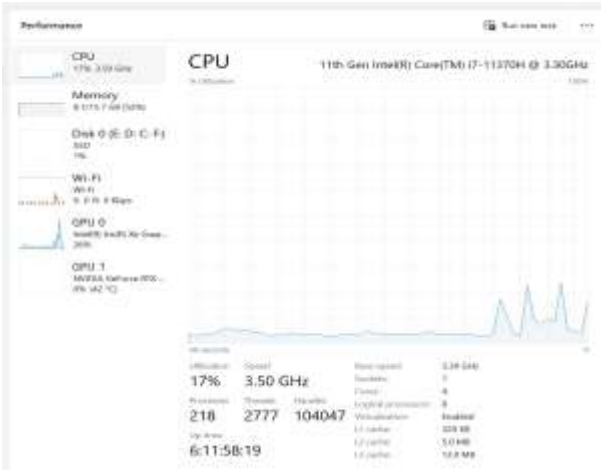

Figure 10.a: *Resource utilized*

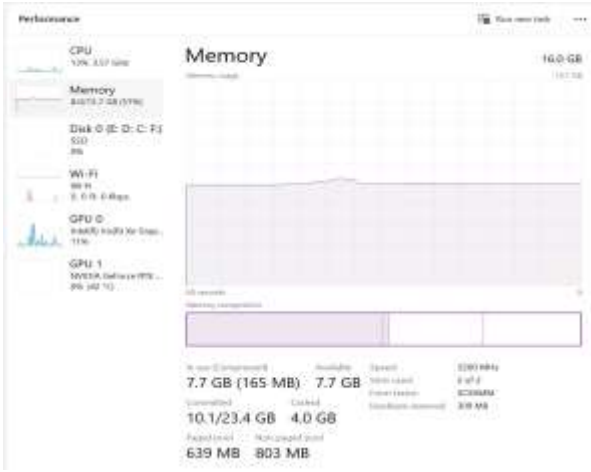

Figure 10.b: *Resource utilized*

Figure 10.c: *Resource utilized.*

- **Confidentiality and Integrity of Cryptographic Keys:** The encryption and decryption processes using Elliptic Curve Encryption ensured that only authorized parties could access and decrypt the encrypted messages, thereby preserving confidentiality.
- **Robust Encryption Algorithms with Appropriate Key Sizes:** The Stegno Curve method utilized Elliptic Curve Encryption, known for its robustness against various cryptographic attacks. The testing demonstrated the efficacy of this encryption algorithm in securing data transmission, with messages encrypted using appropriate key sizes. This choice of algorithm and key size contributed to the overall security of the method.

4.7 Comparison with Other Solutions

Compared with existing cryptographic and steganographic applications, the Stegno Curve stands out for its superior performance and robust security features. The seamless integration of Elliptic Curve Encryption with LSB Steganography sets it apart from traditional methods, offering enhanced data protection while ensuring covert communication channels.

ECC-LSB vs. RSA/AES: ECC provides equivalent security to RSA with much shorter key lengths, making it more efficient. For example, a 256-bit ECC key is comparable in security to a 3072-bit RSA key. This efficiency makes ECC particularly suitable for environments with limited computational resources. AES, a symmetric encryption method, provides high security but does not offer the same public key infrastructure advantages as ECC [76]. Robustness Against Attacks: ECC is more resistant to quantum attacks compared to RSA. The addition of LSB Steganography further enhances security by hiding the existence of encrypted data, making it harder for attackers to detect and intercept [23].

Security Strength

The Stegno Curve method, which integrates ECC and LSB Steganography, offers significant advantages over traditional cryptographic integrations such as RSA and AES. ECC provides equivalent security to RSA with much shorter key lengths—for example, a 256-bit ECC key offers comparable security to a 3072-bit RSA key. This results in more efficient use of computational resources while maintaining high security levels. Additionally, ECC is more resistant to quantum attacks, particularly against Shor's algorithm, whereas RSA is vulnerable to such quantum attacks. While AES provides robust symmetric encryption, it lacks the key exchange capabilities of ECC. Integrating LSB Steganography with ECC enhances security further by concealing the existence of the

encrypted message, making it more challenging for attackers to detect the communication.

Computational Efficiency

When comparing the computational efficiency of the Stegno Curve method with AES and Blowfish integrations, ECC operations demonstrate faster encryption and decryption speeds due to shorter key lengths and more efficient computations. Hybrid approaches like AES-Blowfish are efficient for encrypting bulk data but add complexity in key management. The simplicity of LSB Steganography ensures minimal impact on computational efficiency during the embedding and extraction processes. Although integrating ECC introduces some computational overhead, the combined approach remains practical for real-time applications.

Robustness

In terms of robustness, the Stegno Curve method provides dual-layer security, making it resilient against both cryptographic and steganographic attacks. This dual-layer protection is achieved by combining the strengths of ECC and LSB Steganography. While Multi-Level Steganography (MLS) approaches, which employ multiple encryption algorithms (e.g., AES, Blowfish) and steganographic methods, also offer robust security, they tend to be more complex to implement and manage. Although LSB Steganography's payload capacity is limited by the cover image, ECC's efficient encryption with smaller key sizes optimizes the available payload space.

Practical Applicability

The balance between secure and covert communication can be kept in the integration of ECC with LSB steganography under the Stegno Curve method. Since ECC is computationally complex enough for high security with a relatively small key size, whereas LSB steganography is very simple and provides the most effective way for embedding data, this combination makes the method practical and useful for real-time applications within an environment with limited computational resources. The efficiency and robustness of this technique establish it as a salient solution in safeguarding sensitive information in a host of different digital communication scenarios.

Regarding practical applicability, the Stegno Curve method is well-suited for scenarios requiring high security and covert communication, such as secure messaging, military intelligence, and confidential data transfer. On the other hand, GAN-based steganography excels in creating highly undetectable stego images but demands substantial computational resources for training GAN models. The implementation complexity of the ECC-LSB approach is significantly lower compared to GAN-based methods, which require extensive computational power and expertise in machine learning.

Even though ECE and LSB Steganography integratedly enhance the security of the system, LSB Steganography is still lagging in terms of payload capacity and can be vulnerable to some kind of detection. Further research could focus on integrating advanced AI and ML techniques towards optimum adjustments to the embedding process for more secure communication. As an example, Generative Adversarial Networks can be used to build cover images with a higher degree of complexity than ever before, such that it becomes statistically impossible to detect them as synthetic. This would also mean that GANs should be trained to generate certain types of images for hiding data of interest with the highest payload capacity and lowest detection rate. These adaptive algorithms, driven by AI, might use the content of an image and features of hidden data to choose which pixels to embed the best, thus further increasing robustness and stealth in steganographic processing.

The fast-paced development and sophistication in the field of quantum computing will result in it being able to break even the strongest cryptographic algorithms, including the Elliptic Curve Encryption technique. Hence, research in the near future must have a strong emphasis on the integration of post-quantum cryptographic algorithms with LSB Steganography. Post-Quantum Cryptography: Design of quantum-secure cryptographic algorithms. Direction for research is in the identification of suitable post-quantum cryptographic techniques and their incorporation with lattice-based, hash-based, or multivariate polynomial-based cryptography, combined with LSB Steganography. This would ensure that a robust form of the secure communication methodology can be adapted to deal with the emerging threats due to quantum computers. In addition to this, post-quantum methods will have to be scrutinized for viability and real-world applications in order for them to be taken as effective and thus useful in various developments, including the transition to secure cloud methods, smart home use, and data encryption.

Investigation along such research directions in secure communication allows the field to mature further not only by addressing present limitations but also by considering future technological development. Such efforts will result in the availability of more robust and effective ways to secure sensitive information in an increasingly digital and networked world.

Encryption and Decryption Times: ECC operations are generally faster than RSA due to shorter key lengths and more efficient computations.

Recent advancements have further optimized ECC algorithms, significantly improving encryption and decryption speeds [77].

Steganographic Embedding and Extraction: The time required for embedding and extracting data using LSB Steganography is relatively low. When combined with ECC, the overall computational load increases slightly due to the encryption process, but remains efficient enough for practical applications, especially with modern optimizations [5].

Resistance to Attacks: The combination of ECC and LSB Steganography provides a dual layer of security. ECC ensures that data is securely encrypted, while LSB Steganography hides the presence of this data within cover media. This dual approach significantly reduces the likelihood of successful attacks compared to using either method alone [7].

The feature comparison table of the Stegno Curve and other traditional methods weighs the capabilities of the Stegno Curve method and other traditional approaches to shed light on the significance of each method and compare the features such as their encryption algorithm, steganography technique, integration of encryption and steganography, performance, and more for a better understanding. Table 3 shows the superiority of the Stegno Curve method in the integration of encryption and steganography over other traditional methods, while it also offers top-notch ECC encryption, better resistance to quantum attacks, and encryption combined with steganography to ensure better security.

5. Conclusions

This research has presented a novel fusion of Elliptic Curve Encryption (ECE) with the Least Significant Bit (LSB) Steganography for hidden communication in the Stegno Curve method, offering a comprehensive and robust framework for secure data transmission. Several key findings have emerged through meticulously exploring theoretical foundations, practical implementation, and detailed methodology.

Firstly, integrating ECE and LSB steganography in the Stegno Curve provides a multi-layered defence mechanism that ensures confidentiality, integrity, and stealth in communication channels. By encrypting the message before embedding, the fusion system mitigates the risk of unauthorized access or detection, enhancing the security posture of hidden communication.

Secondly, the fusion of ECE and LSB steganography in the Stegno Curve offers practical implications for various domains requiring covert data transmission, including military, intelligence, and cybersecurity. The ability to conceal sensitive information within innocuous cover media while maintaining cryptographic security opens new avenues for secure communication in environments where privacy and secrecy are paramount.

The Stegno Curve method demonstrates remarkable performance in encryption and steganographic operations. During testing, remarkable speed in encrypting and decrypting messages using ECE, with encryption speeds increasing from 2.07 GHz to 3.50 GHz with minimal memory usage of around 42.4MB was observed. Additionally, the method exhibits efficient embedding and extraction of encrypted data within images using LSB steganography, maintaining high-speed performance even with large image files. This efficient resource utilization ensures optimal performance without compromising security.

Furthermore, integrating ECE and LSB steganography offers practical implications for domains requiring covert data transmission. The ability to conceal sensitive information within innocuous cover media while maintaining cryptographic security opens new avenues for secure communication in environments where privacy and secrecy are paramount. This makes the Stegno Curve method particularly suitable for military, intelligence, and cybersecurity applications, where covert communication channels are essential.

Overall, this research contributes to advancing the field of secure communication by introducing a novel approach that addresses the limitations of existing techniques while offering practical insights and methods. By harnessing the synergistic potential of ECE and LSB steganography, the proposed fusion system represents a significant step forward in the quest for robust and covert communication channels in the digital world.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** This project was funded by Dar Al-hekma University.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Table 3: *Feature comparison of Stegno Curve and other traditional methods*

Features Comparison	Stegno Curve (ECC-LSB)	Traditional RSA-AES Integration	Multi-Level Steganography (MLS)	GAN-based Steganography
Encryption Algorithm	Elliptic Curve Encryption	RSA for key exchange, AES for encryption	AES and Blowfish for encryption	Varies (typically GAN models for embedding)
Steganography Technique	LSB (Least Significant Bit)	LSB, DCT, Spread Spectrum	Multi-level techniques (LSB, DWT, etc.)	GAN-generated cover images
Integration of Encryption and Steganography	Yes	No	Yes	Yes
Performance	Superior speed in encryption and steganography operations	Moderate to high computational load	Moderate to high computational load	High computational load (GAN training)
Security	Robust ECC encryption, resistant to quantum attacks	Strong (RSA and AES), but RSA is vulnerable to quantum attacks	Strong due to multiple encryption layers	Very strong (GAN-based undetectability)
Covert Communication	Yes	Yes	Yes	Yes
Key Size	Smaller key sizes due to ECC (e.g., 256-bit ECC)	Larger key sizes for RSA (e.g., 3072-bit)	Varies based on encryption algorithms	Varies based on the steganography model
Resistance to Attacks	High resistance to brute force and mathematical attacks	Vulnerable to quantum attacks (RSA)	High resistance due to multiple encryption layers	High resistance due to GAN-based models
Usability	User-friendly interface, seamless integration	Depends on specific implementation	Complex, requires careful implementation	Very complex, requires machine learning expertise
Application Range	Suitable for covert communication, secure messaging, data protection	General-purpose encryption and steganography	Secure communication, data protection	Highly secure communication, data hiding

References

[1] Jan, A., Parah, S.A., Hussan, M. et al. (2022). Double layer security using crypto-stego techniques: a comprehensive review. *Health Technol.* 12; 9–31. https://doi.org/10.1007/s12553-021-00602-1

[2] Evaluating the Merits and Constraints of Cryptography-Steganography Fusion: A Systematic Analysis [Internet]. www.researchsquare.com. 2023 [cited 2024 Feb 11]. Available from: https://www.researchsquare.com/article/rs-3167378/v1

[3] Menezes, A.J., Vanstone, S.A. (1993). Elliptic curve cryptosystems and their implementation. *J. Cryptology* 6; 209–224. https://doi.org/10.1007/BF00203817

[4] Kakunuri Sandya and Subhadra Kompella, (2022). A Combined Approach of Steganography and Cryptography with Generative Adversarial Networks: Survey. *Lecture notes in networks and systems*, pp. 187–196, doi: https://doi.org/10.1007/978-981-19-4863-3_18.

[5] Alanzy M, Alomrani R, Alqarni B, Almutairi S. (2023). Image Steganography Using LSB and Hybrid Encryption Algorithms. Applied Sciences [Internet]. 1;13(21):11771. Available from: https://www.mdpi.com/2076-3417/13/21/11771

[6] McAteer I, Ibrahim A, Zheng G, Yang W, Valli C. (2019). Integration of Biometrics and Steganography: *A Comprehensive Review. Technologies* [Internet]. 8;7(2):34. Available from: https://www.mdpi.com/2227-7080/7/2/34

[7] G. G, C. Ashwin, B. V. P, A. A and A. Hiremath, (2021). Enhanced Data Encryption in IOT using ECC Cryptography and LSB Steganography. *International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, India*, pp. 173-177, doi: 10.1109/ICDI3C53598.2021.00043.

[8] Budianto, Christofer & Wicaksana, Arya & Hansun, Seng, (2020). Elliptic Curve Cryptography and LSB Steganography for Securing Identity Data. Available: https://www.researchgate.net/publication/335329077_Elliptic_Curve_Cryptography_and_LSB_Steganography_for_Securing_Identity_Data

[9] Varghese, Fredy and P. Sasikala. (2023). Secure Data Transmission Using Optimized Cryptography and Steganography Using Syndrome-Trellis Coding. Wireless Personal Communications 130;551-578.

[10] G. Shilpa, Mohan CM, Chaurasia N. (2023). Image steganography using LSB embedding and edge adaptive approach. *AIP Conference Proceedings*

[11] R. Bansal and N. Badal, (2022). A novel approach for dual layer security of message using Steganography and Cryptography. *Multimedia Tools and Applications*, 81(15);20669–20684, doi: https://doi.org/10.1007/s11042-022-12084-y.

[12] Zachariah, Babangida & Yabuwat, Patience & Bernard, Ephraim, (2016). Application of Steganography and Cryptography for Secured Data Communication – A Review," *International Journal of Engineering Research & Technology (IJERT),* doi: https://www.researchgate.net/publication/306254107_Application_of_Steganography_and_Cryptography_for_Secured_Data_Communication_-_A_Review

[13] Dhivya C, Sharmila G, Keerthanadevi S, Gangalakshmi S. Steganography (2018). A Technique to Hide the Information using LSB Algorithm [Internet]. *SSRG International Journal of Ceramic Technology (SSRG-IJCT)* -Special Issue. 2018 [cited 2024 Feb 10]. Available from: https://www.internationaljournalssrg.org/uploads/specialissuepdf/ICRMIT/2018/CSE/IJCSE-ICRMIT-P103.pdf

[14] Ahmed DEM, Khalifa OO. (2014). Robust and Secure Image Steganography Based on Elliptic Curve Cryptography [Internet]. *IEEE Xplore.* 288–91. https://ieeexplore.ieee.org/document/7031659

[15] Kheddar H, Hemis M, Himeur Y, Megías D, Amira A. (2023). Deep Learning for Diverse Data Types Steganalysis: A Review *arXiv.org*. https://arxiv.org/abs/2308.04522

[16] Dhivya C, Sharmila G, Keerthanadevi S, Gangalakshmi S. Steganography (2024). A Technique to Hide the Information using LSB Algorithm. *SSRG International Journal of Ceramic Technology (SSRG-IJCT)* -Special Issue. https://www.internationaljournalssrg.org/uploads/specialissuepdf/ICRMIT/2018/CSE/IJCSE-ICRMIT-P103.pdf

[17] J. Bhadra, M. K. Banga and M. V. Murthy, (2017). Securing data using elliptic curve cryptography and least significant bit steganography. *International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, Bengaluru, India, pp. 1460-1466, doi: 10.1109/SmartTechCon.2017.8358607.

[18] Kaur, Ravneet & Singh, Tanupreet, "Hiding Data in video Sequences using LSB with Elliptic Curve Cryptography," *International Journal of Computer Applications.* 117;36-40. 10.5120/20658-3296.

[19] Christofer Derian Budianto, Arya Wicaksana, and Seng Hansun, "Elliptic Curve Cryptography and LSB Steganography for Securing Identity Data (2019). Studies in computational intelligence, pp. 111–127, doi: https://doi.org/10.1007/978-3-030-25217-5_9.

[20] Nandan P, Raghav M, 2, Amit K, Chaturvedi, Scholar P. (2023) Issue 7 *Journal of Emerging Technologies and Innovative Research* 10. https://www.jetir.org/papers/JETIR2307591.pdf

[21] Y.-K. Lee, G. Bell, S.-Y. Huang, R.-Z. Wang, and S.-J. Shyu, (2009). An Advanced Least-Significant-Bit Embedding Scheme for Steganographic Encoding. *Lecture Notes in Computer Science*, pp. 349–360, doi: https://doi.org/10.1007/978-3-540-92957-4_31.

[22] Aslam, Muhammad & Rashid, Muhammad & Azam, Farooque & Abbas, Muhammad & Rasheed, Yawar & Alotaibi, Saud & Anwar, Muhammad, (2022). Image Steganography using Least Significant Bit (LSB) - *A Systematic Literature* 2-38,

[23] S. Roy and Md. M. Islam, (2022). A Hybrid Secured Approach Combining LSB Steganography and AES Using Mosaic Images for Ensuring Data Security. *SN Computer Science*, 3(2), doi: https://doi.org/10.1007/s42979-022-01046-8.

[24] [1]S. Wendzel et al., (2021). A Revised Taxonomy of Steganography Embedding Patterns. doi: https://doi.org/10.1145/3465481.3470069.

[25] A Robust Security Scheme Based on Novel Encoding with LSB Steganography | *IEEE Conference Publication | IEEE Xplore* ieeexplore.ieee.org. https://ieeexplore.ieee.org/document/9513904

[26] Improving Security with Efficient Key Management in Public Cloud using Hybrid AES, ECC, and LSB Steganography comparing with Novel hybrid Cube Base Obfuscation | *IEEE Conference Publication | IEEE Xplore* [Internet]. ieeexplore.ieee.org. [cited 2024 Feb 10]. Available at: https://ieeexplore.ieee.org/document/9780661

[27] E. Bin Hureib, P. Adnan, and A. Gutub, (2020). Enhancing Medical Data Security via Combining Elliptic Curve Cryptography and Image Steganography. *IJCSNS International Journal of Computer Science and Network Security*, 20(8): http://paper.ijcsns.org/07_book/202008/20200801.pdf

[28] Chandrasekaran, Vanmathi & Sevugan, Prabu, (2016). Steganography: A comparative study, analysis of key issues and current trends. 8;4696-4715, https://www.researchgate.net/publication/309670700_Steganography_A_comparative_studyanalysis_of_key_issues_and_current_trends

[29] Al-Shaaby, Ahmed & Al-Kharobi, Talal, (2017). Cryptography and Steganography: *New Approach. Transactions on Networks and Communications*. 5. 10.14738/tnc.56.3914,

[30] D. Nashat and L. Mamdouh, (2019). An efficient steganographic technique for hiding data. *Journal of the Egyptian Mathematical Society*, 27(1), doi: https://doi.org/10.1186/s42787-019-0061-6.

[31] Provably Secure Public-Key Steganography Based on Elliptic Curve Cryptography | *IEEE Journals & Magazine | IEEE Xplore* [Internet]. ieeexplore.ieee.org. [cited 2024 Feb 10]. https://ieeexplore.ieee.org/abstract/document/10418202

[32] R. Denis and P. Madhubala, (2020). Evolutionary Computing Assisted Visually- Imperceptible Hybrid Cryptography and Steganography Model for Secure Data Communication over Cloud Environment.

International Journal of Computer Networks and Applications, 7(6);208, doi: https://doi.org/10.22247/ijcna/2020/205321.

[33] Siddharth, D., & Saini, D. K. J. (2022). IoT Based Lightweight Cryptographic Schemes in Smart Healthcare. *In Advancing Computational Intelligence Techniques for Security Systems Design* (pp. 169-178). CRC Press.

[34] Ahmed, A. A., Malebary, S. J., Ali, W., & Alzahrani, A. A. (2023). A provable secure cybersecurity mechanism based on combination of lightweight cryptography and authentication for internet of things. *Mathematics*, 11(1);220. https://doi.org/10.3390/math11010220

[35] Salih, K. O. M., Rashid, T. A., Radovanovic, D., & Bacanin, N. (2022). A comprehensive survey on the Internet of Things with the industrial marketplace. *Sensors*, 22(3), 730. https://doi.org/10.3390/s22030730

[36] Harn, L., & Lin, C. (2014). Efficient group Diffie–Hellman key agreement protocols. *Computers & Electrical Engineering*, 40(6), 1972-1980. https://doi.org/10.1016/j.compeleceng.2013.12.018

[37] Rahmani, A.M., Bayramov, S. & Kiani Kalejahi, B. (2022). Internet of Things Applications: Opportunities and Threats. *Wireless Pers Commun* 122, 451–476. https://doi.org/10.1007/s11277-021-08907-0

[38] Nyangaresi, V. O. (2022). Lightweight anonymous authentication protocol for resource-constrained smart home devices based on elliptic curve cryptography. *Journal of Systems Architecture*, 133, 102763. https://doi.org/10.1016/j.sysarc.2022.102763

[39] Das, S., & Namasudra, S. (2022). A novel hybrid encryption method to secure healthcare data in IoT-enabled healthcare infrastructure. *Computers and Electrical Engineering*, 101, 107991. https://doi.org/10.1016/j.compeleceng.2022.107991

[40] Xavier, A. P., & Kesavan, R. (2023). Hybrid Elliptic Curve Cryptographic Approach for Data Privacy and Authentication in Secured Map Reduce Layer (SMR) for Optimized CPU Utilization. *IETE Journal of Research*, 69(2), 670-683. https://doi.org/10.1080/03772063.2022.2027285

[41] Qazi, R., Qureshi, K. N., Bashir, F., Islam, N. U., Iqbal, S., & Arshad, A. (2021). Security protocol using elliptic curve cryptography algorithm for wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*, 12, 547-566. https://doi.org/10.1007/s12652-020-02020-z

[42] A. Abdaoui, A. Erbad, A. K. Al-Ali, A. Mohamed and M. Guizani, (2022). Fuzzy Elliptic Curve Cryptography for Authentication in Internet of Things. *IEEE Internet of Things Journal*, 9(12); 9987-9998, doi: 10.1109/JIOT.2021.3121350

[43] Mousavi, S. K., Ghaffari, A., Besharat, S., & Afshari, H. (2021). Security of internet of things based on cryptographic algorithms: a survey. *Wireless Networks*, 27(2), 1515-1555. https://doi.org/10.1007/s11276-020-02535-5

[44] Pawar, R., & Kalbande, D. R. (2020). Elliptical curve cryptography based access control solution for IoT based WSN. *Innovative Data Communication Technologies and Application:* ICIDCA 2019 (pp. 742-749). https://doi.org/10.1007/978-3-030-38040-3_85

[45] Agrawal, L., & Tiwari, N. (2020). A review on IoT security architecture: attacks, protocols, trust management issues, and elliptic curve cryptography. *Social Networking and Computational Intelligence: Proceedings of SCI-2018*, 457-465. https://doi.org/10.1007/978-981-15-2071-6_36

[46] Khan, M. A., Quasim, M. T., Alghamdi, N. S., & Khan, M. Y. (2020). A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data. *IEEE Access*, 8, 52018-52027. https://doi.org/10.1109/ACCESS.2020.2980739

[47] Kavitha, S., Alphonse, P. J. A., & Reddy, Y. V. (2019). An improved authentication and security on efficient generalized group key agreement using hyper elliptic curve based public key cryptography for IoT health care system. *Journal of medical systems*, 43(8), 260. https://doi.org/10.1007/s10916-019-1378-2

[48] Mehmood, M. S., Shahid, M. R., Jamil, A., Ashraf, R., Mahmood, T., & Mehmood, A. (2019, November). A comprehensive literature review of data encryption techniques in cloud computing and IoT environment. *2019 8th International Conference on Information and Communication Technologies (ICICT)* (pp. 54-59). IEEE. https://doi.org/10.1109/ICICT47744.2019.9001945

[49] Vahdati, Z., Yasin, S., Ghasempour, A., & Salehi, M. (2019). Comparison of ECC and RSA algorithms in IoT devices. *Journal of Theoretical and Applied Information Technology*, 97(16), 4293.

[50] Kumari, S., Karuppiah, M., Das, A. K., Li, X., Wu, F., & Kumar, N. (2018). A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers. *The Journal of Supercomputing*, 74(12);6428-6453. https://doi.org/10.1007/s11227-017-2048-0

[51] Lara-Nino, C. A., Diaz-Perez, A., & Morales-Sandoval, M. (2018). Elliptic curve lightweight cryptography: A survey. *IEEE Access*, 6, 72514-72550. https://doi.org/10.1109/ACCESS.2018.2881444

[52] Odat, A. M., & Otair, M. A. (2016). Image steganography using modified least significant bit. *Indian Journal of Science and Technology*. https://ischolar.sscldl.in/index.php/indjst/article/view/125794

[53] Hamid, N., Yahya, A., Ahmad, R. B., & Al-Qershi, O. M. (2012). Image steganography techniques: an overview. *International Journal of Computer Science and Security (IJCSS)*, 6(3);168-187.

[54] Akhtar, N., Johri, P., & Khan, S. (2013). Enhancing the security and quality of LSB based image steganography. *5th International Conference and Computational Intelligence and Communication Networks* (pp. 385-390). IEEE. https://doi.org/10.1109/CICN.2013.85

[55] Muhammad, K., Ahmad, J., Farman, H., & Zubair, M. (2015). A novel image steganographic approach

for hiding text in color images using HSI color model. *arXiv preprint* arXiv:1503.00388. https://doi.org/10.48550/arXiv.1503.00388

[56] Wang, S., Sang, J., Song, X., Niu, X. (2015). Least significant qubit (LSQb) information hiding algorithm for quantum image. *Measurement* 73;352-359. https://doi.org/10.1016/j.measurement.2015.05.038

[57] Al-Shatnawi, A. M., & AlFawwaz, B. M. (2013). An integrated image steganography system with improved image quality. *Applied Mathematical Sciences*, 7(71), 3545-3553.

[58] Liu, Q., Sung, A. H., Ribeiro, B., Wei, M., Chen, Z., & Xu, J. (2008). Image complexity and feature mining for steganalysis of least significant bit matching steganography. *Information Sciences*, 178(1), 21-36. https://doi.org/10.1016/j.ins.2007.08.007

[59] Xu, W. L., Chang, C. C., Chen, T. S., & Wang, L. M. (2016). An improved least-significant-bit substitution method using the modulo three strategy. *Displays*, 42, 36-42. https://doi.org/10.1016/j.displa.2016.03.002

[60] Mao, Q. (2014). A fast algorithm for matrix embedding steganography. *Digital Signal Processing*, 25; 248-254. https://doi.org/10.1016/j.dsp.2013.11.001

[61] Nagaraj, V., Vijayalakshmi, V., & Zayaraz, G. (2013). Color image steganography based on pixel value modification method using modulus function. *IERI Procedia*, 4; 17-24. https://doi.org/10.1016/j.ieri.2013.11.004

[62] Biswas, D., Biswas, S., Majumder, A., Sarkar, D., Sinha, D., Chowdhury, A., & Das, S. K. (2012). Digital image steganography using dithering technique. *Procedia Technology*, 4; 251-255. https://doi.org/10.1016/j.protcy.2012.05.038

[63] Karim, S. M., Rahman, M. S., & Hossain, M. I. (2011). A new approach for LSB based image steganography using secret key. *14th international conference on computer and information technology (ICCIT 2011)* (pp. 286-291). IEEE. https://doi.org/10.1109/ICCITechn.2011.6164800

[64] Tahir, A., & Amit, D. (2015). A novel approach of LSB based steganography using parity checker. International *Journal of Advanced Research in Computer Science and Software Engineering*, 5(1).

[65] Rawat, D., & Bhandari, V. (2013). A steganography technique for hiding an image in an image using LSB method for 24 bit color image. *International Journal of Computer Applications*, 64(20); 15–19. https://doi.org/10.5120/10749-5625

[66] ElSayed, A., Elleithy, A., Thunga, P., & Wu, Z. (2015). Highly secure image steganography algorithm using curvelet transform and DCT encryption. *Long Island Systems, Applications and Technology*. https://doi.org/10.1109/lisat.2015.7160204

[67] Senthooran, V., & Ranathunga, L. (2014). DCT coefficient dependent quantization table modification steganographic algorithm. *First International Conference on Networks & Soft Computing (ICNSC2014)*. https://doi.org/10.1109/cnsc.2014.6906644

[68] Gaba, J., & Kumar, M. (2013). Implementation of steganography using CES technique. *IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, Shimla, India (pp. 395-399). https://doi.org/10.1109/iciip.2013.6707622

[69] Nofriansyah, D., Defit, S., Nurcahyo, G. W., Ganefri, G., Ridwan, R., Ahmar, A. S., & Rahim, R. (2018). A new image encryption technique combining Hill cipher method, Morse code and least significant bit algorithm. *Journal of Physics: Conference Series*, 954, 012003. https://doi.org/10.1088/1742-6596/954/1/012003

[70] Sridevi, D. R., Paruchuri, V. L., & Rao, K. S. S. S. (2013). Image steganography combined with cryptography. *international journal of computers & technology*, 9(1); 976–984. https://doi.org/10.24297/ijct.v9i1.4160

[71] Shoup, V. (2009). A computational introduction to number theory and algebra. *Cambridge University Press.*

[72] Kak, A. (2015). Lecture notes on computer and network security. *Purdue University.* Available: https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture8.pdf

[73] Stallings, W. (2011). Operating systems: internals and design principles. *Prentice Hall Press.* Available: https://dl.acm.org/doi/abs/10.5555/2012029

[74] Abd Aziz, A. Z., Mohd Sultan, M. F., & Mohamad Zulkufli, N. L. (2024). Image Steganography: Comparative Analysis of their Techniques, Complexity and Enhancements. *International Journal on Perceptive and Cognitive Computing*, 10(1); 59–70. https://doi.org/10.31436/ijpcc.v10i1.449

[75] Ji, H., Xie, L., Wang, C., Yin, Y., & Lu, S. (2015). CrowdSensing: A crowd-sourcing based indoor navigation using RFID-based delay tolerant network. *Journal of Network and Computer Applications*, 52, 79-89. https://doi.org/10.1016/j.jnca.2015.02.010

[76] Rahman, M. S., Hossain, M. S., Rahat, E. H., Dipta, D. R., Faruque, H. M. R., & Fattah, F. K. (2019). Efficient hardware implementation of 256-bit ECC processor over prime field. *International conference on electrical, computer and communication engineering (ECCE)* (pp. 1-6). IEEE. https://doi.org/10.1109/ECACE.2019.8679184

[77] Mahto, D., & Yadav, D. K. (2017). RSA and ECC: a comparative analysis. *International Journal of Applied Engineering Research*, 12(19); 9053-9061. https://blkcipher.pl/assets/pdfs/ijaerv12n19_140.pdf

[78] Maimuț, D., & Matei, A. C. (2022). Speeding-Up Elliptic Curve Cryptography Algorithms. *Mathematics*, 10(19); 3676. https://doi.org/10.3390/math10193676