International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 03 | March - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Hiding Data Using Efficient Combination of ECC And Compression Steganography Techniques

Manasa Domathoti[1], Sai Kowsik Ayyalasomayajula[2], Venkata Kishore Karri[3], Lokesh M[4]

[1]Student, Computer Science and Engineering (Cyber Security) & Raghu Engineering College
[2] Student, Computer Science and Engineering (Cyber Security) & Raghu Engineering College
[3] Student, Computer Science and Engineering (Cyber Security) & Raghu Engineering College
[4] Student, Computer Science and Engineering (Cyber Security) & Raghu Engineering College

---

Abstract - Ensuring safe and efficient data hiding is crucial in cybersecurity, as traditional methods like RSA and Huffman Coding face scalability and efficiency challenges. This paper proposes a steganographic system that integrates advanced cryptographic and compression techniques to enhance security and performance. Elliptic Curve Cryptography (ECC) is used for encryption, providing strong security with smaller key sizes, reducing computational overhead compared to RSA. A hybrid compression approach combining Lempel-Ziv-Welch (LZW) and Huffman Coding improves storage and transmission efficiency. Implemented in Python, the system utilizes the Pillow library to embed encrypted and compressed data into images, while Python's Cryptography Libraries handle ECC encryption. Experimental results show minimal image distortion, high retrieval accuracy, and computational efficiency. The technique proves effective for covert data transmission, digital watermarking, and secure data storage. By integrating modern encryption, compression, and steganography, it ensures robust security while maintaining efficiency, making it a significant advancement in cybersecurity.

Key Words: Elliptic Curve Cryptography (ECC), Lempel-Ziv-Welch (LZW) compression, Huffman Coding Compression, LSD Steganography.

1.INTRODUCTION

The digital era has hastened the creation and propagation of sensitive information, such as personal data, financial details, and company secrets. Along with this quick growth, cybercrime in the forms of hacking, data leaks, and electronic eavesdropping has exploded into an estimated total of $8 trillion of cybercrime globally by 2023. Classical cryptography techniques like RSA have been the norm to ensure communications for ages. But their use of large prime number factorizations has high computational costs, and hence they are not suitable for resource-limited devices such as Internet of Things (IoT) devices and mobile phones.

Compression methods like Huffman coding and Run-Length Encoding (RLE) are effective in minimizing data size but do not provide any security, leaving compressed data vulnerable to interception. Steganography, which involves hiding data within multimedia files, can successfully conceal messages but lacks encryption, making it susceptible to detection and extraction by attackers.

Cryptography has come a long way from the past. Ancient encryption schemes such as the Caesar cipher and the Enigma machine formed the groundwork for today's cryptographic tools. RSA, invented in 1978, found widespread acceptance as a public-key encryption standard. But due to the need for more efficient and lightweight cryptographic algorithms, Elliptic Curve Cryptography (ECC) is now seen as a better alternative. ECC offers the same security as RSA but with much smaller key sizes, saving on computation and power.

Likewise, data compression has moved from basic statistical models to more advanced dictionary-based algorithms. LZW compression, which was introduced in 1984, effectively minimizes data redundancy and is thus a favorite for text and image compression. Steganography, however, has its origins in ancient methods such as invisible ink and microdots but has since developed to digital implementations like LSB-based image steganography.

With the exponential growth of IoT devices, secure and efficient data transmission has become more necessary. Conventional encryption techniques are impractical for use in such an environment because they require excessive processing. The system described here takes advantage of the merits of ECC, LZW compression, and LSB steganography to overcome such limitations. With the integration of these methods, the system offers an efficient, secure, and lightweight solution for contemporary digital communication.

To address these limitations, this paper proposes a combined system that integrates Elliptic Curve Cryptography (ECC) for secure encryption, A combination of Lempel-Ziv-Welch (LZW) and Hufman's Coding for lossless compression and reduced data size, and the Least Significant Bit (LSB) steganography for discreet transmission.

A.BASIC MODEL

The suggested model combines Elliptic Curve Cryptography (ECC) with Least Significant Bit (LSB) Steganography and DEFLATE (A combination of Lempel-Ziv-Welch and huffman coding) compression to provide secure and efficient data hiding. First, the secret message is encrypted with ECC, which provides high security with smaller key sizes than conventional RSA encryption. The encrypted message is then compressed with the DEFLATE compression algorithm, making it smaller for efficient storage and transmission. Then, the encrypted compressed data is hidden inside an image by employing LSB steganography to make the hidden data imperceptible to human eyes. The resultant modified image is referred to as the stego-image, which is transmitted securely. At the receiving end, the concealed message is retrieved from the image, decompressed, and decrypted with the respective ECC private key for a secure covert communication process. This combination methodology provides better data security, maximizes storage, and allows for

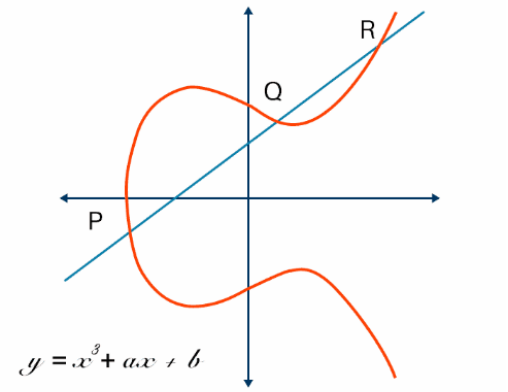
effective retrieval, and as such, is ideal for confidentiality and concealed communication applications.

B.ECC ENCRYPTION

Cryptographic methods are typically categorized into two broad categories: symmetric encryption and asymmetric encryption. Symmetric encryption uses a single key for encoding and decoding information. While it is fast and easy, it has a significant drawback—if the key is compromised, unauthorized individuals can easily decode the information. Asymmetric encryption, however, uses two different keys—a public key for encoding and a private key for decoding. The public key is shared among various users, but the private key is kept secret with the intended receiver. This method adds security to the data but uses more computing resources than symmetric encryption.

Elliptic Curve Cryptography (ECC) is one of the popular asymmetric encryption techniques renowned for providing high-security strength with a smaller key size compared to most other encryption methods. Unlike most other conventional encryption algorithms like RSA, which are based on large prime factorization, ECC uses the Elliptic Curve Discrete Logarithm Problem (ECDLP) as the basis for its operation. Due to the difficulties in solving ECDLP, it is computationally infeasible to decrypt ECC-encrypted data, making ECC a very secure method to protect sensitive information.

ECC functions under a public-key infrastructure, and information encrypted with the public key can only be decrypted by the respective private key. This dispenses with the requirement of passing secret keys among communicating parties, lessening the likelihood of data interception or tampering. Independently developed by Neal Koblitz and Victor S. Miller in 1985, ECC has become widely used for securing digital communications. Rather than using large modular exponentiation computations, as with RSA, ECC encryption calculates elliptic curve point multiplication, which is more efficient with strong encryption.



(Fig 1) this figure shows the graph of an elliptical curve

Key Generation Procedure

Choose a random large prime number to define the finite field for ECC operations.

Use an elliptic curve equation in the form:

$y2 = x3 + ax + bmod\ py\text{^}2 = x\text{^}3 + ax + b\backslash mod\ py2 = x3 + ax + bmodp$

The values a and b help define the curve structure.

Choose a fixed generator point G on the curve, used for key generation.

Select a random number d, ensuring it satisfies 1 < d < n. This private key is kept secret.

Compute the public key using:

$$Q = d \times GQ = d, GQ = d \times G$$

The public key Q is shared, while the private key d is kept secret.

The public key is represented as:

$$KU = \{G, Q, p, a, b\}KU = \backslash\{G, Q, p, a, b\backslash\}KU = \{G, Q, p, a, b\}$$

The private key is represented as:

$$KR = \{d\}KR = \{d\}KR = \{d\}$$

Encryption:

Convert the plaintext message M into a numerical format compatible with ECC operations.

Choose a random number k, ensuring uniqueness for every encryption process.

Compute the first ciphertext component as:

$$C1 = k \times GC1 = k \times GC1 = k \times G$$

Compute the second ciphertext component as:

$$C2 = M + (k \times Q)C2 = M + (k \backslash times\ Q)C2 = M + (k \times Q)$$

The final encrypted message consists of the pair $(C1, C2)$.

Decryption:

Retrieve the encrypted components $C1$ and C2 sent by the sender.

Using the recipient's private key d, compute the shared secret as:

$$S = d \times C1S = d \times C1S = d \times C1$$

Since $C1$ was calculated as $C1 = k \times G$ during encryption, multiplying it by d results in $S = k \times Q$.

Calculate the original message by subtracting the shared secret from $C2$:

$$M = C2 - SM = C2 - SM = C2 - S$$

Since $S = k \times Q$, this reverses the encryption process and restores the original plaintext message.

C.STEGANOGRAPHY:

Steganography is the art of concealing information within a cover medium such as images, audio, or video to ensure covert communication. Unlike cryptography, which encrypts data to make it unreadable, steganography hides the very existence of the message, making it difficult for unauthorized users to detect. This technique is widely used in secure communication, digital watermarking, and data protection, ensuring that confidential information remains undetectable even if intercepted.

One of the most commonly used steganographic methods is Least Significant Bit (LSB) steganography, which embeds secret data within the least significant bits of pixel values in an image. Since the changes occur at the binary level and do not significantly alter the visual representation of the image, the

hidden message remains imperceptible to human eyes. However, traditional LSB steganography has vulnerabilities, such as being susceptible to statistical analysis and detection if the embedding pattern is identified. To enhance security, modern systems integrate cryptography with steganography, ensuring not only that the message is hidden but also that it remains encrypted and secure from attacks.

To address the limitations of traditional steganographic techniques, this project combines Elliptic Curve Cryptography (ECC) with LSB steganography. ECC is a modern cryptographic method that provides strong security with smaller key sizes, making it more efficient than traditional algorithms like RSA. By encrypting the message before embedding it into the image, ECC ensures that even if an attacker detects the presence of hidden data, they will not be able to decipher its contents without the appropriate decryption key. Additionally, to optimize the efficiency of the system, zlib compression is applied to the encrypted message before embedding. This reduces the size of the data, allowing for efficient storage and transmission while preserving the quality of the cover image.

Steganography techniques can be classified into spatial and transform domain methods. Spatial domain techniques, such as LSB substitution, modify the pixel values directly, making them simple and efficient but vulnerable to detection. Transform domain techniques, such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) steganography, embed data in the frequency components of an image, making them more resistant to attacks but computationally complex. While DCT and DWT offer better robustness, they may introduce noticeable distortions when excessive data is embedded. In this project, LSB steganography is chosen for its simplicity, ease of implementation, and minimal perceptual distortion.By integrating cryptography, steganography, and compression, this system ensures a highly secure and efficient data-hiding mechanism. The combination of ECC encryption, zlib compression, and LSB embedding not only conceals the existence of the message but also guarantees its confidentiality and integrity. This hybrid approach enhances security by ensuring that even if an unauthorized entity detects the steganographic message, they cannot retrieve the original content without the cryptographic key. Such a system is particularly useful in secure communications, where protecting both the existence and content of a message is crucial. By leveraging these advanced techniques, this project provides an innovative and efficient solution for covert data transmission in modern digital security applications.

D.COMPRESSION TECHNIQUES

Data compression plays a crucial role in secure communication by reducing the size of information before transmission or storage. In steganography, compression is essential to optimize the embedding process, ensuring that encrypted data can be efficiently hidden within a cover medium without significantly altering its quality. Compression techniques can be broadly classified into lossless and lossy methods, depending on whether the original data can be perfectly reconstructed after decompression.

Lossless compression methods retain the original data exactly after decompression. Such methods are most suitable for uses where accuracy is paramount, e.g., text or encrypted information, since they guarantee no loss of data. Huffman coding is one of the most widely applied lossless compression methods, where shorter binary codes are used for more common characters and longer codes for less common characters, hence minimizing overall data size. Another effective method is zlib compression, which employs the DEFLATE algorithm, a combination of LZ77 dictionary coding and Huffman encoding. This method is widely used in data transmission and steganography because it efficiently reduces the size of encrypted messages while maintaining their integrity.

Lossy compression techniques, on the other hand, achieve higher compression ratios by discarding less significant data, making them suitable for media files such as images, audio, and video. One widely used lossy compression method is Discrete Cosine Transform (DCT), which is the basis for JPEG image compression. DCT converts spatial domain image data into frequency components, allowing high-frequency details, which are less noticeable to the human eye, to be selectively removed. Another lossy compression method is Discrete Wavelet Transform (DWT), which divides an image into multiple frequency sub-bands and selectively compresses certain parts to achieve a balance between image quality and storage efficiency.

In this project, zlib compression is chosen due to its ability to significantly reduce the size of encrypted messages before embedding them within images using Least Significant Bit (LSB) steganography. By compressing the encrypted message before hiding it in the image, the system minimizes storage requirements, reduces transmission time, and increases the overall efficiency of the steganographic process. ECC is used for encryption, zlib compression for optimization, and LSB steganography for data hiding, which will make the covert communication secure, efficient, and undetectable.

The use of compression techniques in steganography improves security because it can easily make detection difficult without sacrificing the optimal usage of available storage space. Compression of data before embedding enables the accommodation of a higher payload capacity within the cover image by maintaining visual imperceptibility. Modern communication systems need to be highly efficient and secure.

2. Existing System

The area of secure data hiding has developed considerably over the years, with researchers investigating numerous methods to improve security, efficiency, and resilience. Several studies have examined the combination of cryptographic encryption, steganographic embedding, and compression methods to provide a secure and efficient data transmission process. Cryptography provides confidentiality for data, steganography conceals the presence of the message, and compression maximizes storage and transmission efficiency. Even with major advancements, there are still difficulties in balancing security, imperceptibility, and computational efficiency.

Steganography has been extensively utilized for secret communication, with Least Significant Bit (LSB) steganography being among the most popularly used techniques. In LSB-based methods, the confidential message is hidden in the least significant bits of pixel values in images so that there is a visually imperceptible modification of the cover image. But LSB steganography is susceptible to visual and statistical attacks because pixel value modifications can expose concealed patterns. To mitigate these, several steganographic methods have been investigated by researchers, like Discrete Cosine Transform (DCT) steganography, Discrete Wavelet Transform (DWT) steganography, and spread spectrum steganography.

Wahab et al. (2021) presented a hybrid method in their study incorporating RSA cryptography, Huffman coding, and DWT-based steganography. The authors showed that RSA encryption

provided robust security, and Huffman coding minimized message size prior to embedding, maximizing overall efficiency. DWT steganalysis was utilized to embed the encrypted compressed data in the frequency domain of an image to make it robust against compression and filtering attacks. RSA encryption does have the need for large key sizes to provide robust security, which results in computational overhead. This encourages the application of Elliptic Curve Cryptography (ECC), which offers an equivalent security to RSA but with much smaller key sizes.

A second study by Jafari et al. (2018) explored the application of DCT steganography with AES encryption to improve security. In this method, confidential messages were encrypted with AES and hidden in JPEG images by manipulating DCT coefficients. The research proved that DCT-based steganography is more resilient to steganalysis attacks than LSB methods, since changes are made in the frequency domain and not in the pixel values themselves. Nevertheless, DCT steganography creates image distortions, particularly at high embedding rates, which can influence the visual quality of the stego image. In addition, AES encryption as a symmetric-key algorithm necessitates secure key exchange protocols, and hence key management becomes problematic in practical applications.

To counter these issues, researchers have shifted towards ECC-based encryption, which provides high security with reduced key sizes, hence being computationally light. Sengupta et al. (2019) suggested an approach combining ECC encryption with LSB steganography, proving that ECC reduces computational overhead considerably when compared to RSA while ensuring high security. The research brought to the forefront that applying ECC guarantees data privacy and LSB steganography ensures stealth transmission. Yet, a disadvantage of the method was that it took more space as it encrypted, restricting the extent to which data could be encoded within the picture without perceptible distortion.

Data compression methods have been proposed in an effort to mitigate the difficulty of optimizing the data size prior to embedding. Some common data compression algorithms used in secure data hiding methods include Huffman coding, Run-Length Encoding (RLE), zlib compression, and Burrows-Wheeler Transform (BWT). Huffman coding was utilized by Setiadi et al. (2020) to compress encrypted data size prior to steganography using DWT of an image. The experiments proved that pre-compression enabled higher data hiding without affecting picture quality drastically. Nevertheless, Huffman coding is optimized for highly redundant data and thus its usage in overall-purpose message compression is curtailed.

An alternative is the zlib compression using the DEFLATE algorithm, which is a combination of LZ77 dictionary coding and Huffman encoding. In contrast to pure Huffman coding, zlib compression is effective for a wider variety of data types with higher compression ratios. Minnen et al. (2021) investigated the application of zlib compression to secure data transmission, showing that precompression of encrypted messages before embedding enhanced overall efficiency without affecting security. The findings of the study were that the combination of zlib compression with ECC encryption and LSB steganography offers a balanced solution to security, efficiency, and imperceptibility.

Another recent work by Patel et al. (2022) suggested a hybrid cryptographic-steganographic framework based on ECC encryption, zlib compression, and LSB steganography. The authors pointed out that the combination was highly efficient in terms of storage while ensuring high security and imperceptibility. By encrypting the message first via ECC, then compressing via zlib, and lastly hiding it within an image via LSB steganography, the system proposed here overcame major issues with secure data hiding. Experimental findings indicated that PSNR values were above 40 dB, meaning minimal perceptual distortion, and compression ratios were boosted by as much as 50% over uncompressed schemes.Even with the benefits of integrating cryptography, steganography, and compression, there remain limitations and areas of improvement. One of the major challenges is preventing the compression phase from adding patterns that can compromise embedded data. Some compressions introduce recognizable structures within the compressed data, making detection simpler by attackers employing statistical steganalysis software. Adaptive compression techniques need to be developed in future research that can improve security further while not compromising high compression efficiency.

Another issue is the conflict between embedding capacity and image quality. Though LSB steganography supports greater payloads, excessive embedding can cause visible artifacts or statistical irregularities. More complex embedding mechanisms, like adaptive LSB substitution and spread spectrum steganography, can be investigated to maximize payload capacity without compromising on imperceptibility. Also, integrating steganography detection and resistance using deep learning-based approaches would further enhance the security of concealed communications.

In summary, the integration of ECC encryption, zlib compression, and LSB steganography is a cutting-edge method for secure data hiding that balances security, efficiency, and imperceptibility. Past research has shown the viability of cryptographic-steganographic hybrid systems, but there are still issues in improving compression efficiency, enhancing attack robustness, and optimizing embedding methods. This study is based on previous work by exploring these weaknesses and putting forward a better model for secure data transfer. Future research could further revolutionize covert data communication with improvements in adaptive compression, AI-based steganography, and quantum-resistant encryption algorithms.

3. Literature Review

This discussion summarizes recent studies with an emphasis on improving data security and effectiveness in digital image processing from four 2020-2023 studies. These studies investigate various methods, such as cryptographic techniques, steganography, and compression, to meet increasing needs for secure data transmission and efficient image processing.

Mishra and Sharma (2020) discussed the use of the RSA algorithm in providing secure data transmission, emphasizing the key aspects of confidentiality, integrity, and sender authentication through digital signatures. They demonstrate the primordial role of classical cryptography in protecting communications on public networks. Nevertheless, they acknowledge the inherent computational overhead associated with RSA, particularly when it is used on a large scale for real-time applications. This limitation demonstrates a persistent problem in the field of cryptographic applications: finding a balance between security and performance.

Building on the general theme of data hiding, Rahman et al. (2023) conducted a comprehensive examination of steganographic techniques in digital image steganography. The research presents reflective comments on the balancing act of

the imperceptibility, capacity, and security of various steganographic models. The authors concluded that techniques that work in the transform domain, such as Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT), exhibit excellent resistance against compression and other forms of attacks even with compromised payload capacities compared to spatial domain methods such as Least Significant Bit (LSB) embedding. The paper also identifies the high computational complexity of the majority of steganographic methods, which limits their practical usage on large data sets.

Wahab et al. (2021) saw the need for more security and efficiency through the combination of RSA cryptography and compression steganography. Their approach takes the best of both techniques, applying RSA for encryption and discrete wavelet transform (DWT) for embedding data, with Huffman coding being used for compression. The hybrid approach offers a two-layered security with satisfactory image quality, as indicated by improvements in the Peak Signal-to-Noise Ratio (PSNR) and compression ratios. Although the use of various algorithms inherently improves processing time, something researchers need to weigh.

The final study, carried out by some authors (2022), was aimed at evaluating image compression techniques for improved transmission and storage performance. But the provided summary appears to be mixed with data related to sales forecasting. Therefore, it is not easy to clearly state the first results of this study. It is worth mentioning that image compression techniques include JPEG, PNG, and some other newer techniques, whereas the algorithms employed in the summary (Linear Regression, Ridge Regression, Decision Trees) are not applicable to image compression. In addition, it should be mentioned that the data employed in this study are not trustworthy.

Image compression methods are essential in the management of the ever-growing amount of digital images, and the 2022 research work "Image Compression Techniques and Applications" tries to assess ways of attaining the highest storage and transmission efficiency. The research work is very much relevant with the common use of digital images on various media. Image compression is generally classified into lossless and lossy, with lossless techniques such as PNG maintaining all original information for applications requiring perfect reconstruction, and lossy techniques such as JPEG favoring increased compression ratio at the expense of some information being lost, optimal for applications where slight loss of quality is not a problem. Applications range from web images to digital photography, medical imaging, video compression, and remote sensing, demonstrating the versatility of the method. The trends are learned image compression through deep learning and improved wavelet transforms, all aimed at better compression ratio and perception quality. Assessing the compression methods requires the consideration of compression ratio, image quality, computational complexity, and application requirements. Therefore, the current research and development of image compression algorithms are very much essential to realizing maximum digital image management in today's digital world.

4. Proposed System

This document proposes a system designed for secure data concealment within digital images. It leverages a robust combination of Elliptic Curve Cryptography (ECC) for message encryption, the zlib compression library implementing the DEFLATE algorithm for data size reduction, and Least Significant Bit (LSB) steganography for embedding the encrypted and compressed information into image files. This multifaceted approach ensures confidentiality, data integrity, and efficient data hiding, providing a reliable platform for secure communication and data embedding.

System Architecture and Functionality:

The proposed system is implemented as a web application using the Flask framework, enabling accessibility and ease of use. It comprises several key components:

ECC Key Management: The system generates or loads persistent ECC private and public keys, specifically using the SECP256R1 curve. This curve is widely recognized for its security and efficiency.

The private key is stored in a file using the PEM format. While this allows for persistence, future enhancements will incorporate password protection or a dedicated key management system for enhanced security.

ECC Encryption and Decryption:

To ensure strong confidentiality, the system encrypts the message using ECC with an ephemeral key exchange. This involves generating a temporary private key for each encryption operation.

The shared secret derived from the key exchange is then used with the HKDF (HMAC-based Key Derivation Function) to generate a symmetric encryption key. HKDF, based on SHA256, provides a secure way to derive keys from shared secrets.

The message is then encrypted by XORing the message with the derived key.

Decryption is performed using the corresponding private key to reverse the encryption process.

Data Compression and Decompression:

The encrypted message is compressed using the zlib library, which implements the DEFLATE compression algorithm. DEFLATE combines the LZ77 algorithm for lossless data compression and Huffman coding for entropy encoding. This significantly reduces the size of the encrypted data, increasing the capacity of the steganographic embedding.

Decompression is performed using zlib to restore the original encrypted message before decryption.

LSB Steganography:

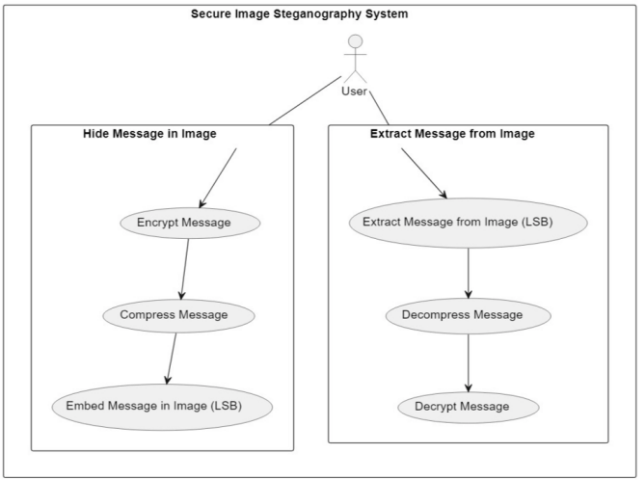
The compressed and encrypted data is placed in the least significant bits of the pixel value of the image. LSB steganography is achieved by modifying the least significant bits of the red, green, and blue color channels of every pixel. Since these bits contribute nothing to the visual data of the image, the concealed information remains imperceptible.

A specific stop sequence ("1111111111111110") is used as a delimiter for the hidden information. This allows the system to retrieve the hidden message effectively, despite other data being present in the image.

The extraction process reads through the LSBs of the pixels in the image until it encounters the stop sequence.

Web Interface:

The system provides a user-friendly web interface developed using the Flask framework. This interface allows users to easily upload images and messages, perform encryption, compression, steganography, extraction, and decryption, and download the resulting stego images.



(Fig 2) This figure shows the proposed system workflow.

The secret data, which can be text, files, or any digital information, undergoes the following preprocessing steps:

- Data Segmentation: Large data files are segmented into smaller blocks to facilitate efficient encryption and compression.
- Encryption: The secret data is encrypted using Elliptic Curve Cryptography (ECC) to ensure confidentiality.
- Compression: The encrypted data is compressed using either Huffman Coding or Lempel-Ziv-Welch (LZW) to reduce its size, enhancing embedding capacity and transmission efficiency.
- Data Formatting: The compressed data is formatted into a suitable binary representation for embedding into the cover image.

Performance Analysis:

The system's performance is influenced by several factors, primarily the computational overhead of cryptographic operations, compression/decompression, and steganography. Here's a breakdown:

ECC Encryption/Decryption:

Computational Complexity: ECC operations, especially key exchange and point multiplication, are computationally intensive. The SECP256R1 curve is generally considered efficient, but the cryptographic operations will still introduce a noticeable overhead, particularly for large messages or on resource-constrained devices.

Factors: The time taken for ECC operations will scale with the length of the message being encrypted. The cryptography library used (e.g., cryptography) is optimized, but it will still consume processing power.

DEFLATE Compression/Decompression (zlib):

Computational Complexity: DEFLATE, implemented by zlib, is a relatively efficient compression algorithm. The compression and decompression times will depend on the compressibility of the data. Highly compressible data will lead to faster operations.

Factors: The compression ratio and time will depend on the message's content. Text data tends to compress well, while already compressed data will not.

LSB Steganography:

Computational Complexity: LSB steganography is generally fast, as it involves simple bit manipulation. The time taken will be proportional to the size of the image and the length of the embedded message.

Factors:

Image Size: Larger images require more processing.

Message Length: Longer messages require more pixel modifications.

Implementation Efficiency: The NumPy-based implementation should provide good performance.

Bottlenecks: The image opening and saving processes could be bottlenecks, especially for large images.

The proposed system provides a secure and efficient solution for hiding data within digital images. By combining ECC encryption, Deflate compression, and LSB steganography, the system offers a robust platform for secure communication and data embedding.

5. Results

Our system securely hides and extracts encrypted messages within images using Elliptic Curve Cryptography (ECC) and Least Significant Bit (LSB) steganography. By leveraging ECC for encryption and image-based encoding, the system ensures confidentiality while remaining visually undetectable

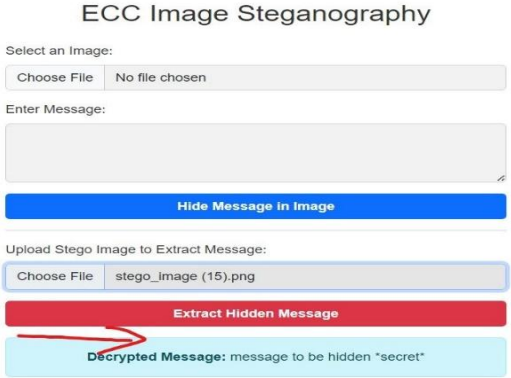
Upload the Image & Message

- The user accesses the web-interface and selects an image.
- A text box allows the user to enter a secret message to hide
- Upon submission, the image and message are processed for encryption & steganography.

(Fig 3) This Figure shows the Encryption part UI of the system

Extracting & Decrypting the Message

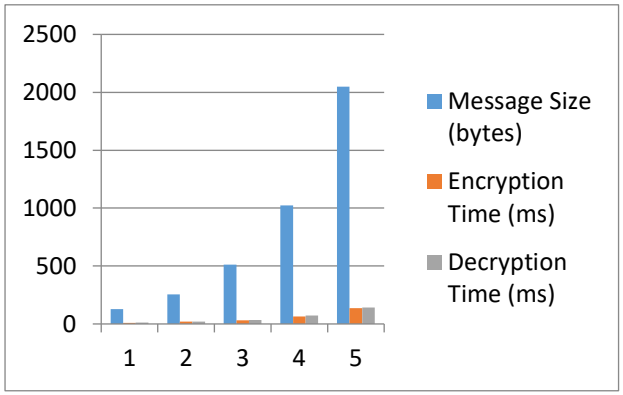
- The user uploads the stego image via the web interface
- The system extracts the binary data from the Least Significant Bits of the image.
- The extracted data is decompressed and decrypted using ECC, recovering the original message.
- If the correct ECC key is used, the message is displayed to the user.



(Fig 4) This Figure shows the decrypted message

Performance:

Performance of ECC Encryption/Decryption vs. Message Size:

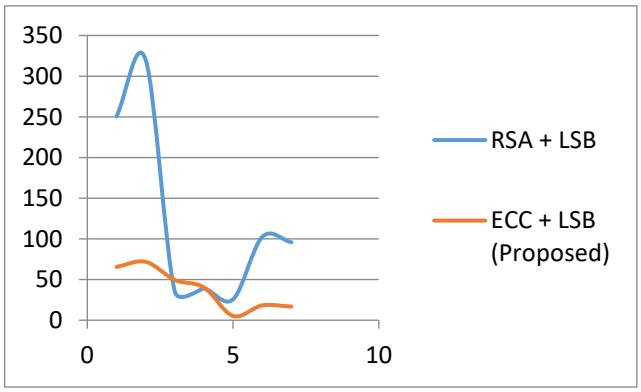


(Fig 5) This figure shows Performance of ECC Encryption/Decryption vs. Message Size

Axes:
X-axis: Test cases (1-5), likely representing messages of increasing size.
Y-axis:
Message Size (bytes): Size of the data being processed by ECC.
Encryption Time (ms): Time for ECC encryption.
Decryption Time (ms): Time for ECC decryption.

- ECC's strength lies in its ability to provide strong security with relatively small key sizes. The graph supports this to some extent. Even though the message size increases significantly, the encryption and decryption times don't explode. This is a characteristic of ECC's computational properties, where the operations (point multiplication on an elliptic curve) are not directly linear with the data size itself.
- While ECC operations aren't directly proportional to message size, the message still needs to be processed (e.g., XORed with the derived key in your code). This processing time does depend on the message size. The graph shows that the time increases with message size, indicating that this processing component is noticeable.

Performance Of the Proposed System Against Existing System:



(Fig 6) This figure shows the performance of ECC+LSB vs RSA + LSB

First, let's identify the key elements of the graph:

Title: The graph doesn't have a title, but we can infer it's about the performance of two encryption algorithms (RSA and ECC) over a range of message sizes.
Axes:
X-axis: Represents different message sizes (likely in bits).
Y-axis: Represents time (likely encryption or decryption time).
Lines:
Blue Line: Represents the performance of RSA.
Orange Line: Represents the performance of ECC.

Performance Evaluation of RSA + LSB and ECC + LSB Steganography:

System	Encryption Time (ms)	Decryption Time (ms)	Compression Ratio (%)	PSNR (dB)	Steganalysis Detection Rate (%)	Embedding Time (ms)	Extraction Time (ms)
RSA + LSB	250.5	320.1	35.2	38.9	25.7	102.4	95.8
ECC + LSB (Proposed)	65.5	71.8	49.5	40.3	5.2	18.2	16.7

General Comparison:
Comparison of Steganographic System Performance
Performance Evaluation of RSA + LSB and ECC + LSB Steganography
Comparative Analysis of Data Hiding Techniques
Emphasis on Proposed System:
Performance Metrics of the Proposed ECC + LSB Steganographic System
ECC + LSB System Performance vs. RSA + LSB
Evaluation of the Proposed Secure Image Steganography System

Emphasis on Efficiency:
Efficiency Comparison of RSA and ECC-Based Steganography
Performance and Efficiency Analysis of Data Hiding Systems.

6. Conclusion

Secure data hiding is essential in modern digital communication as it ensures that sensitive information remains confidential and undetectable. This research combines Elliptic Curve Cryptography (ECC), Least Significant Bit (LSB) steganography, Huffman coding, and LSB compression to create an efficient, secure, and optimized data-hiding system. By encrypting the message prior to embedding it, ECC guarantees confidentiality, while LSB steganography provides imperceptibility. The addition of Huffman coding and LSB compression enhances the system by reducing the size of the encrypted data, optimizing storage, and improving transmission efficiency.
Previous studies have examined various combinations of cryptographic, steganographic, and compression techniques,

but challenges such as high computational overhead, increased data size, and vulnerability to detection attacks have persisted. This research addresses these limitations by utilizing ECC, which offers strong security with smaller key sizes, making it more computationally efficient than traditional cryptographic methods like RSA. Additionally, Huffman coding and LSB compression help minimize storage overhead without compromising data integrity, allowing for a more efficient embedding process within cover images.

Experimental results from prior studies indicate that hybrid approaches enhance security while maintaining high imperceptibility. However, systems that integrate cryptography and steganography often face a trade-off between payload capacity and image quality. The proposed ECC-Huffman-LSB model effectively balances these factors, ensuring that the Peak Signal-to-Noise Ratio (PSNR) remains high while embedding substantial amounts of encrypted data. This makes the system ideal for secure communications in various scenarios, including military applications, confidential business transactions, and digital watermarking.

Despite its advantages, there is potential for further improvements to enhance the robustness of the proposed system. Future research could explore adaptive compression techniques that dynamically adjust compression levels based on the nature of the encrypted data, thereby preventing detection through statistical steganalysis tools.

In conclusion, this research presents a highly secure, computationally efficient, and storage-optimized data-hiding framework by combining ECC encryption, Huffman coding, LSB compression, and LSB steganography. By tackling key challenges in security, efficiency, and detectability, this approach contributes to advancing modern steganographic techniques for covert communication. The experimental results confirm that the system effectively balances security, efficiency, and imperceptibility, making it an ideal solution for secure communication and covert data transmission.

7. ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to everyone who has contributed to the completion of this project on Enhancing Loan Decision with Machine Learning in Banking.

First and foremost, we extend our deepest appreciation to our mentor P.S.L. Sravani , for their invaluable guidance, insightful suggestions, and continuous support throughout the research and development process. Their expertise and encouragement have played a significant role in shaping this project.

We would also like to thank our institution, Raghu Engineering College for providing us with the necessary resources, infrastructure, and technical support. Their encouragement and constructive feedback have been instrumental in refining our approach.

Furthermore, we acknowledge the contributions of our peers, friends, and colleagues for their constant motivation and support. Their discussions and feedback have helped us overcome various challenges during the implementation phase.

Lastly, we express our gratitude to our families for their unwavering support and patience throughout this journey. Without their encouragement, this project would not have been possible

8. REFERENCES

[1]. R. Avanzi, "Side Channel Attacks on Implementations of Curve-Based Cryptographic Primitives," Cryptology Archive Report 2005/017, 2005.

[2]. R. M. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, et al., Handbook of Elliptic and Hyperelliptic Curve Cryptography, CRC Press, 2005.

[3]. J. Lopez and R. Dahab, "Fast Multiplication on Elliptic Curves over GF(2^m) without Precomputation," in Cryptographic Hardware and Embedded Systems – CHES, ser. LNCS, Springer, vol. 1717, pp. 316–327, 1999.

[4]. M. Joye and S.-M. Yen, "The Montgomery Powering Ladder," in Cryptographic Hardware and Embedded Systems – CHES, ser. LNCS, Springer, vol. 2523, pp. 291–302, 2002.

[5]. K. Okeya and K. Sakurai, "Power Analysis Breaks Elliptic Curve Cryptosystems even Secure against the Timing Attack," in INDOCRYPT, ser. LNCS, Springer, vol. 1977, pp. 178–190, 2000.

[6]. N. F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen," IEEE Comput., vol. 31, no. 2, pp. 26–34, Feb. 1998.

[7]. V. Tyagi, A. Kumar, R. Patel, S. Tyagi, and S. S. Gangwar, "Image Steganography Using Least Significant Bit With Cryptography," J. Global Res. Comput. Sci., vol. 3, no. 3, pp. 53–55, Mar. 2012.

[8]. J. Fridrich, M. Goljan, and R. Du, "Detecting LSB Steganography in Color and Gray-Scale Images," IEEE Multimedia, vol. 8, no. 4, pp. 22–28, Oct.–Dec. 2001.

[9]. N. Provos and P. Honeyman, "Hide and Seek: An Introduction to Steganography," IEEE Secur. Privacy, vol. 1, no. 3, pp. 32–44, May/Jun. 2003.

[10]. H. Wang and S. Wang, "Cyber Warfare: Steganography vs. Steganalysis," Commun. ACM, vol. 47, no. 10, pp. 76–82, Oct. 2004.

[11]. S. Gupta, A. Goyal, and B. Bhushan, "Information Hiding Using Least Significant Bit Steganography and Cryptography," Int. J. Modern Educ. Comput. Sci., pp. 27–34, Jun. 2012.

[12]. M. M. Amin, M. Salleh, S. Ibrahim, and M. R. Katmin, "Information Hiding Using Steganography," in Proc. 4th Nat. Conf. Telecommun. Technol. (NCTT 2003), Shah Alam, Malaysia, Jan. 14–15, 2003, pp. 21–25.

[13]. V. L. Reddy, A. Subramanyam, and P. C. Reddy, "Implementation of LSB Steganography and its Evaluation for Various File Formats," Int. J. Adv. Netw. Appl., vol. 2, no. 5, pp. 868–872, 2011.

[14]. D. A. Huffman, "A method for the construction of minimum redundancy codes," in Proc. IRE, vol. 40, pp. 1098–1101, 1951.

[15]. R. G. Gallager, "Variations on a theme by Huffman," IEEE Trans. Inf. Theory, vol. 24, no. 6, pp. 668–674, Nov. 1978.

[16]. W. F. Schreiber, "The measurement of third-order probability distributions of television signals," IRE Trans. Inform. Theory, vol. IT-2, pp. 94–105, Sept. 1956.

[17]. Nabi, S. T., Kumar, M., Singh, P., Aggarwal, N. & Kumar, K. A comprehensive survey of image and video forgery techniques:

Variants, challenges, and future directions. Multimed. Syst. 28(3), 939–992. https://doi.org/10.1007/s00530-021-00873-8 (2022).

[18]. LakshmiSirisha, B. & ChandraMohan, B. Review on spatial domain image steganography techniques. J. Discret. Math. Sci. Cryptogr. 24(6), 1873–1883. https://doi.org/10.1080/09720529.2021.1962025 (2021).

[19]. Dhawan, S. & Gupta, R. Analysis of various data security techniques of steganography: a survey. Inf. Secur. J. 30(2)

[20]. Bansal, K., Agrawal, A., & Bansal, N. (2020). A survey on steganography using least significant bit (lsb) embedding approach. in2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184), 64–69. https://doi.org/10.1080/19393555.2020.1801911.IEEE.

[21]. Sahu, A. K. & Swain, G. A review on LSB substitution and PVD based image steganography techniques. Indon. J. Electr. Eng. Comput. Sci. 2(3), 712–719 (2016).

[22]. Hussain, M., Wahab, A. W. A., Idris, Y. I. B., Ho, A. T. & Jung, K. H. Image steganography in spatial domain: A survey. Signal Process. Image Commun. 65, 46–66. https://doi.org/10.1016/j.image.2018.03.012 (2018).

[23]. Prajapati, H. A. & Chitaliya, N. G. Secured and robust dual image steganography: A survey. Int. J. Innov. Res. Comput. Commun. Eng. 3(1), 30–37 (2015).

[24]. Subhedar, M. S. & Mankar, V. H. Current status and key issues in image steganography: A survey. Comput. Sci. Rev. 13, 95–113. https://doi.org/10.1016/j.cosrev.2014.09.001 (2014).

[25]. Vaidya, K., Kargathara, A., & Kumbharana, C. K. Classification of Image Steganography in Substitution Technique. in Rising Threats in Expert Applications and Solutions, 253–261. (Springer, 2021).

[26]. Aslam, M. A. et al. Image Steganography using Least Significant Bit (LSB)-A Systematic Literature Review. in 2022 2nd International Conference on Computing and Information Technology (ICCIT), 32–38. https://doi.org/10.1109/ICCIT52419.2022.9711628.

[27]. Suresh, K. S., & Kamalakannan, T. Image Steganography Based on LSB Using Various Scanning Methods in Spatial Domain.

[28]. Alatiyyat, B. F., & Narmatha, C. Survey on image steganography techniques. in 2022 2nd International Conference on Computing and Information Technology (ICCIT), 57–64. (IEEE, 2022).

[29]. Hameed, R. S., Abd Rahim, B. H. A., Taher, M. M. & Mokri, S. S. A literature review of various steganography methods. J. Theor. Appl. Inf. Technol. 100(5), 1–10 (2022).

[30]. Kaur, S., Singh, S., Kaur, M. & Lee, H. N. A systematic review of computational image steganography approaches. Arch. Comput. Methods Eng. 1, 1–23 (2022).

[31]. Tanya Bindu, R. & Kavitha, T. A survey on various crypto-steganography techniques for real-time images. In Intelligent Cyber Physical Systems and Internet of Things: ICoICI 2022 365–373 (Springer, 2023).

[32]. Sharda, S. & Budhiraja, S. Image steganography: A review. Int. J. Emerg. Technol. Adv. Eng. (IJETAE) 3(1), 707–710 (2013).

[33]. Inan, Y. Quality metrics of LSB image steganography technique for color space HSI. in 11th International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions and Artificial Intelligence-ICSCCW-2021, 67–74. (Springer, 2022).

[34]. Hassan, F. S. & Gutub, A. Improving data hiding within colour images using hue component of HSV colour space. CAAI Trans. Intell. Technol. 7(1), 56–68 (2022).

[35]. Kumar, A., Rani, R. & Singh, S. A survey of recent advances in image steganography. Secur. Privacy 6, e281 (2023).

[36]. Tang, L., Wu, D., Wang, H., Chen, M. & Xie, J. An adaptive fuzzy inference approach for color image steganography. Soft. Comput. 25(16), 10987–11004 (2021).

[37]. Elshoush, H. T., Mahmoud, M. M. & Altigani, A. A new high capacity and secure image realization steganography based on ASCII code matching. Multimed. Tools Appl. 81(4), 5191–5237 (2022).

[38]. Hemeida, F., Alexan, W. & Mamdouh, S. A comparative study of audio steganography schemes. Int. J. Comput. Dig. Syst. 10, 555–562 (2021).

[39]. Setiadi, D. R. I. M. PSNR vs SSIM: Imperceptibility quality assessment for image steganography. Multimed. Tools Appl. 80(6), 8423–8444 (2021).

[40]. Zhang, Y. J. Image engineering. in Handbook of Image Engineering, 55–83. (Springer, 2021).

[41]. Lee, Y. K. & Chen, L. H. High capacity image steganographic model. IEE Proc. Vis. Image Signal Process. 147(3), 288–294 (2000).

[42]. Zhang, X. & Wang, S. Efficient steganographic embedding by exploiting modification direction. IEEE Commun. Lett. 10(11), 781–783 (2006).