Capstone ID : 20250905

A Secure Elliptic Curve Based Public-Key Steganography

Team:
Nikhita Moncy - 22BCE8527
Pallavi Sankar - 22BCE8138
Meghana V Patil - 22BCE9123

FALL SEM 2025-26

Introduction

- Focuses on developing a secure communication system that combines Elliptic Curve Cryptography (ECC) with image-based steganography.
- The primary goal is to enhance the security of hidden data by encrypting it using ECC before embedding it into digital media files (images).
- The ECC algorithm will be used for generating public-private keys and encrypting messages, ensuring that even if the steganographic layer is compromised, the message remains unreadable.

What is ECC ?

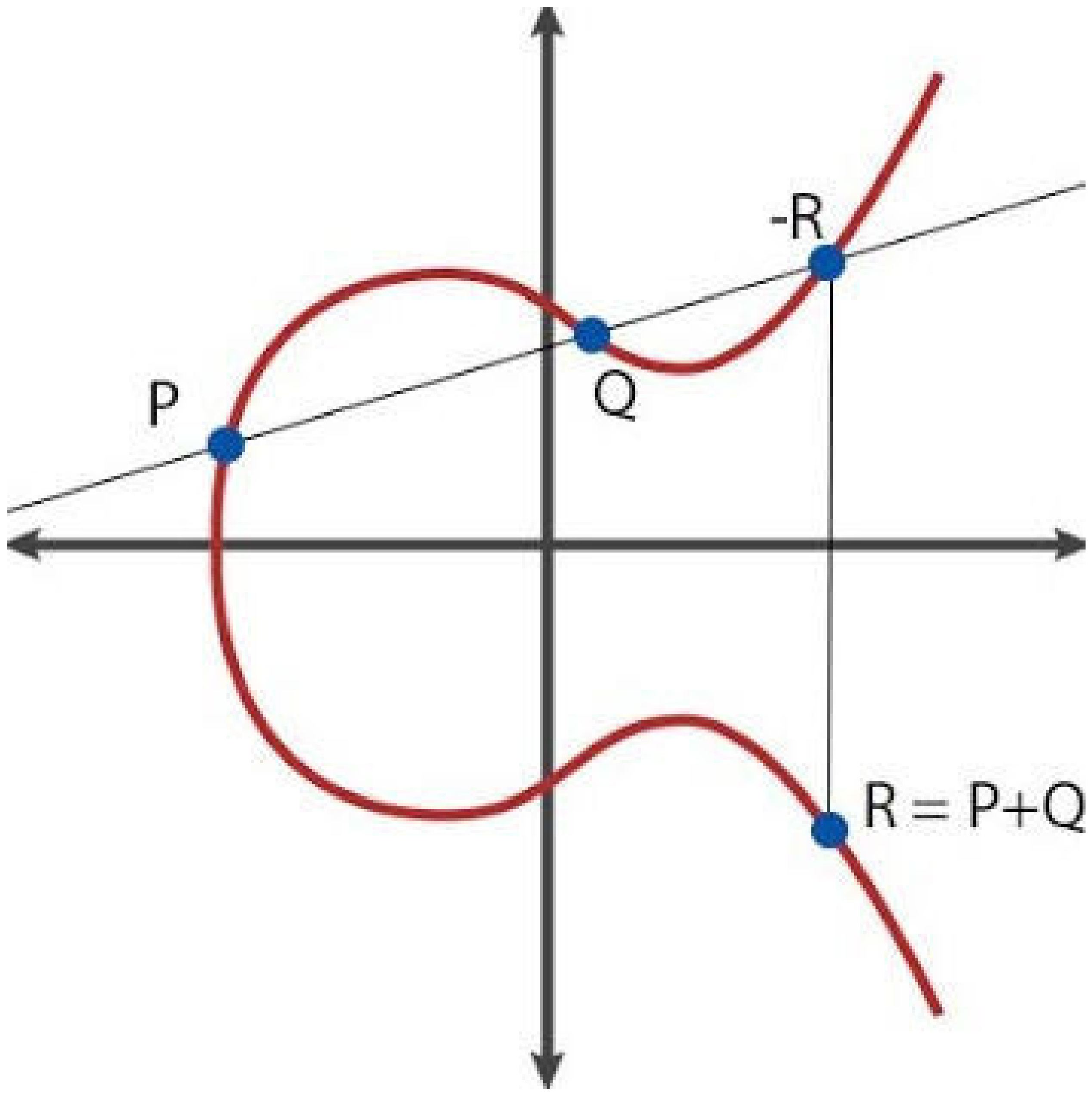
ECC Definition

In cryptography, ECC (Elliptic Curve Cryptography) is a public-key encryption method that uses the mathematical properties of elliptic curves over finite fields to provide security.

- **Equation: $y^2 = x^3 + ax + b$**
- Cryptographic operations:
  Point Addition : P + Q = R
  Scalar Multiplication: k * P

ECC Uses

- Enables secure communication over the internet and other digital systems, particularly in low-resource environments like smartphones and IoT devices.
- Secures web traffic, provides digital signatures, facilitates mutual authentication, and is crucial for blockchain technologies



Extra Security of ECC over Existing Methods

1. Plain Steganography (without encryption):

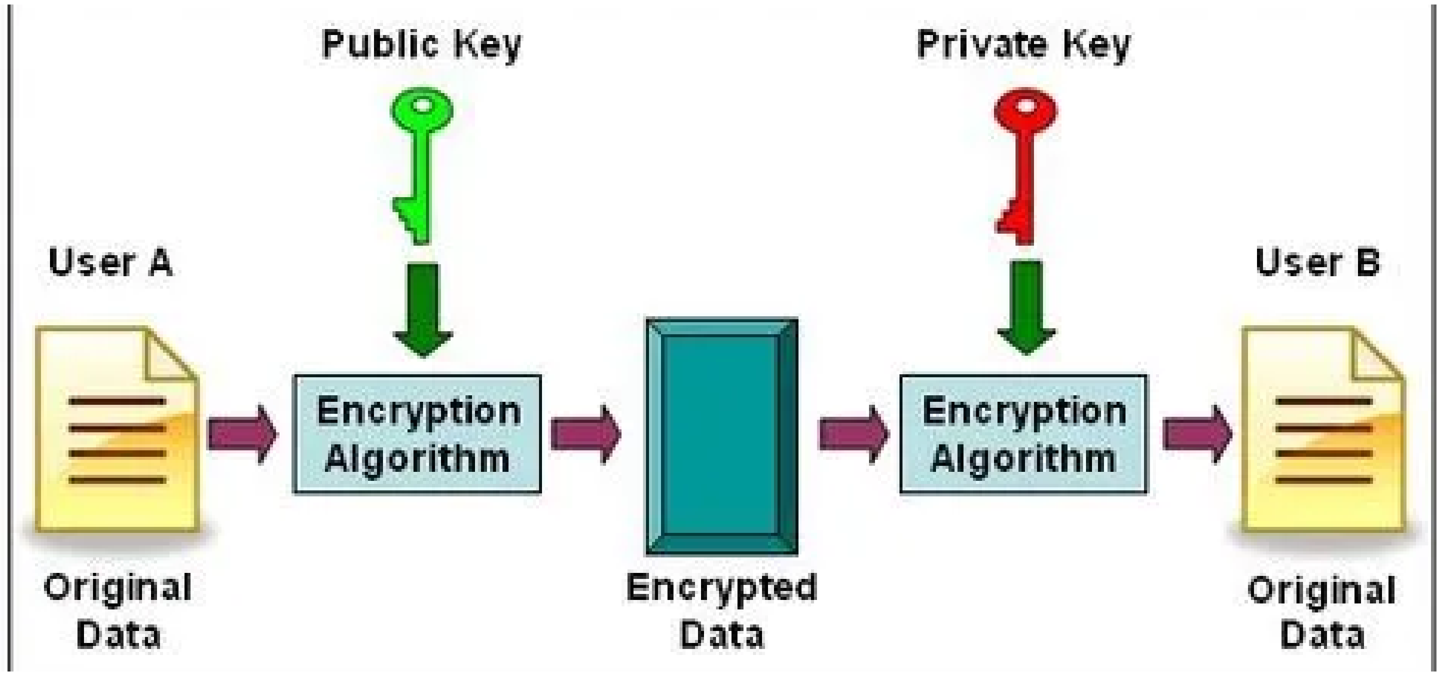
- Hides secret data directly inside images/audio.
- **Drawback:** If attacker detects and extracts hidden data, the secret is directly revealed.

2.Steganography with Symmetric Encryption (AES/DES):

- Secret message is encrypted with one secret key before hiding.
- **Drawback**: Requires key sharing. If key is leaked or intercepted, message is compromised.

3.Steganography with RSA (public-key cryptography)

- Provides better security than symmetric methods.
- **Drawback:** RSA requires large key sizes (2048–4096 bits) → heavy computation, storage, and slower performance.



Public and Private Key

Public and private keys are a matched pair used in asymmetric encryption systems for secure communication and digital signatures.

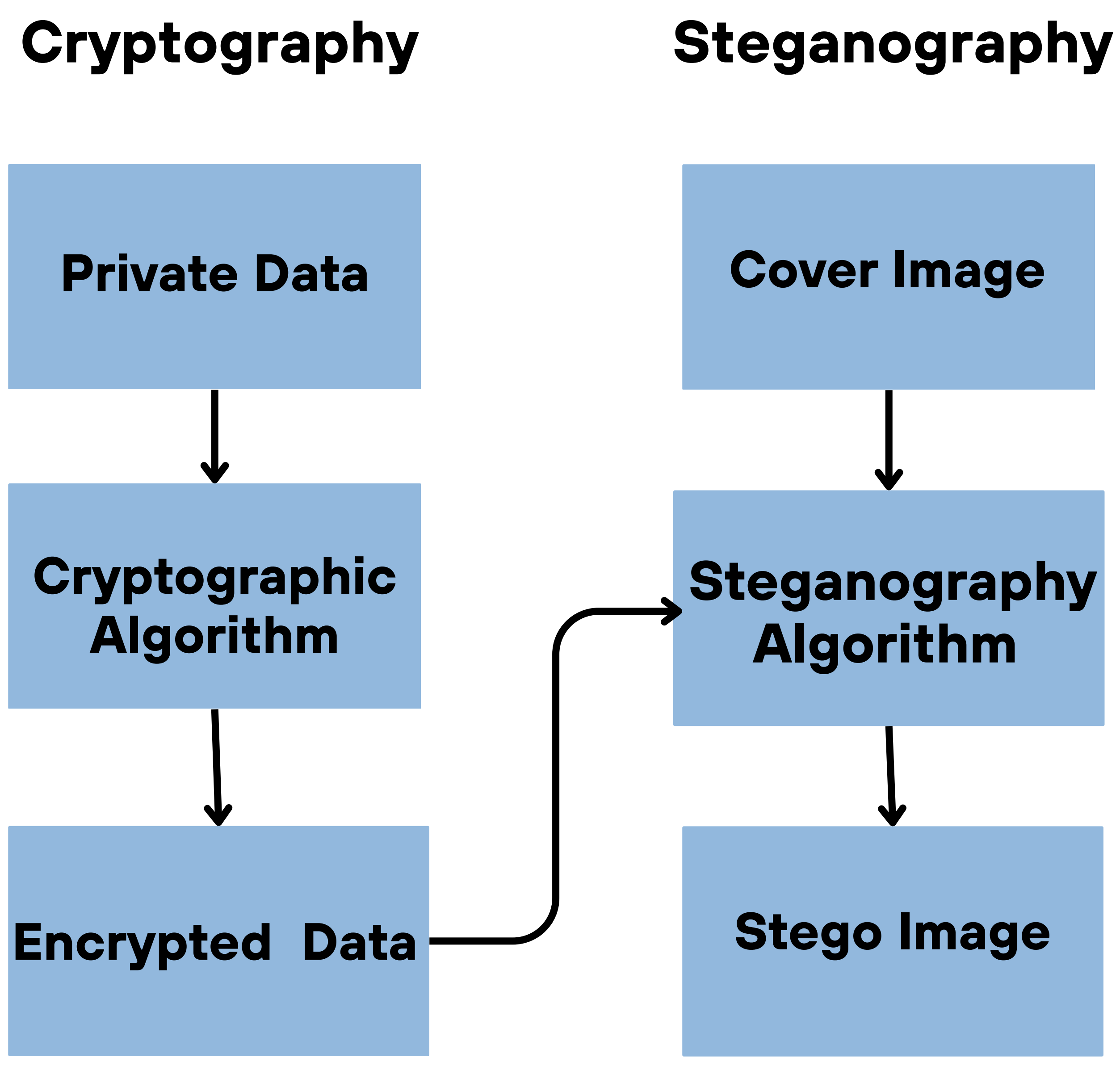
- Sender: Uses the receiver's public key to encrypt the secret.
- Receiver: Uses their private key to decrypt after extraction.

How is ECC a Better Choice?

Comparison Table

Technique	Encryption Type	Avg PSNR	Security	Computation
LSB Only	None	28 dB	Weak	Fast
AES + LSB	Symmetric	36 dB	Moderate	Moderate
RSA + LSB	Asymmetric	40 dB	Strong	Heavy
ECC + Elligator + LSB (Ours)	Public-Key	42 dB	Very Strong	Efficient

Steganography



Steganography is the technique of hiding secret data within non-secret "cover" data (like an image, audio, or text file) so that the very existence of the secret message is concealed from unauthorized observers.

Key Benefits:

- Evading detection
- Offering anonymity for users
- Ensuring resistance to tampering
- Providing digital watermarking for copyright protection

Use of ECC in Steganography:

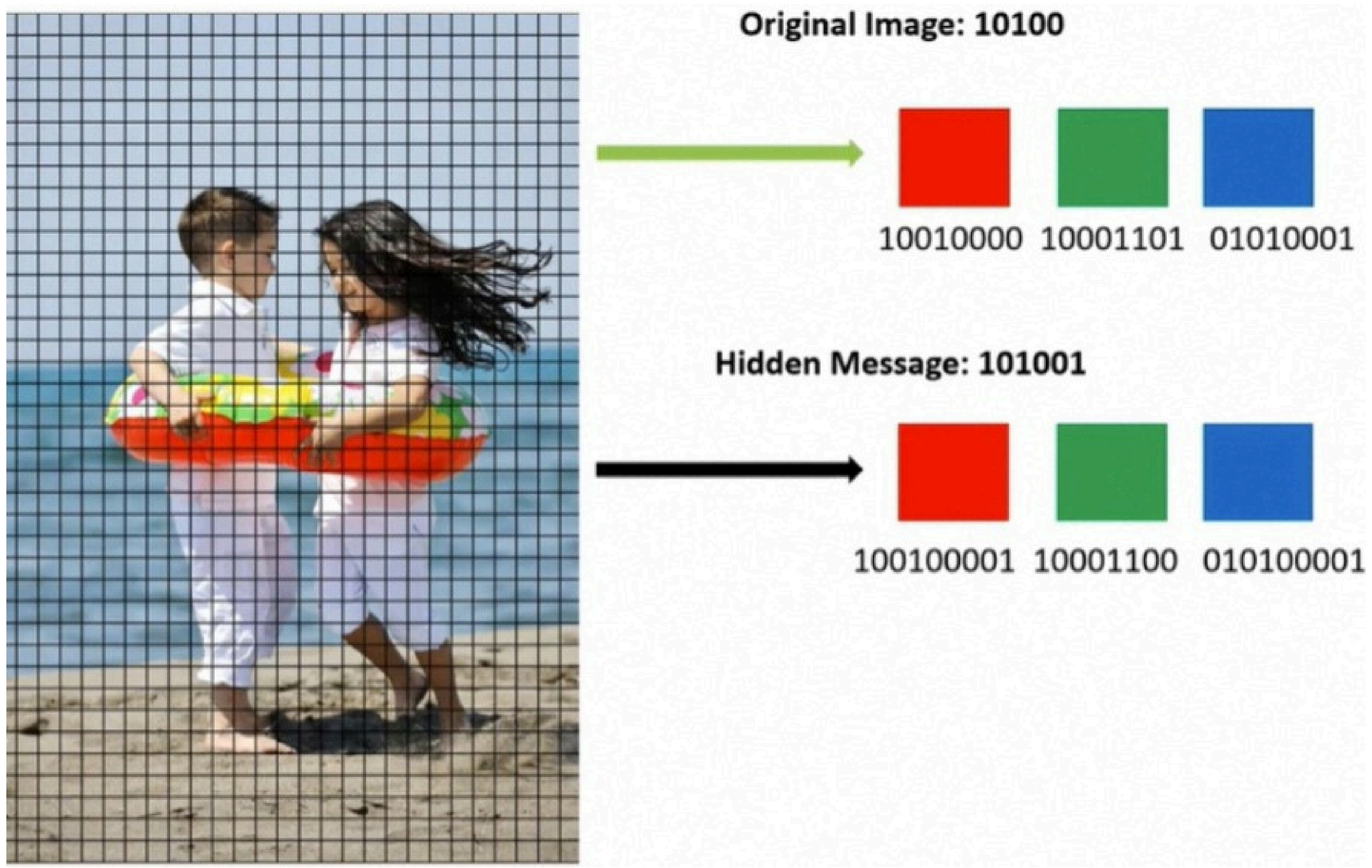
- Encryption before Hiding using ECC Public key.
- End-to-end confidentiality using Public and Private key.
- Even if steganography is broken, cryptography keeps the message safe.
- Makes system secure against both cryptanalysis and steganalysis attacks.

Image Stenography

Image steganography is the process of hiding secret information (like text, image, or file) inside an image in such a way that it looks unchanged to the human eye.

Workflow (Basic)

- Input Selection → Take a cover image and a secret message.
- Message Conversion → Convert secret message into binary form (0s and 1s).
- Embedding → Hide these bits in the Least Significant Bits (LSB) of image pixels.
- Stego Image → Save the modified image (looks same as original).
- Extraction → Read LSBs back, reconstruct binary → convert to original message.



Literature Survey

Authors & Citation	**Contribution**	**Proposed Work**	**Advantages**	**Disadvantages**
Zhang et al., 2024, Provably Secure Public-Key Steganography Based on ECC, IEEE TIFS [1]	Introduced provably secure ECC-based public-key steganography	ECC with provable security for public-key stego	Strong theoretical security; small ECC keys	High computational complexity; heavy for large media
Homam El-Taj, 2024, ECC + LSB Steganography, IJCESEN [2]	Combined ECC encryption with classical LSB steganography	ECC + LSB image stego	Simple to implement; improved confidentiality	Vulnerable to LSB-specific attacks; limited payload
Ganavi & Prabhudeva, 2022, Two-Layer Security Using ECC + DWT + LSB, MECS Press [3]	Two-layer image security combining ECC and DWT	ECC + DWT + LSB	Enhanced imperceptibility and robustness	DWT increases computational load
Dhar & Banerjee, 2019, ECC-Cryptosystem for Image Hiding, Springer [4]	ECC-based secure message hiding in images	ECC-Cryptosystem for image hiding	Strong key security; suitable for sensitive data	Limited evaluation on large images
Hemanta Kumar Mohanta, 2014, Secure Data Hiding Using ECC + Steganography, IJCA [5]	Hybrid ECC + steganography for secure data hiding	ECC + stego model	Improved data confidentiality	Older method; weaker against modern stego-analysis

Literature Survey

Authors & Citation	Contribution	Proposed Work	Advantages	Disadvantages
Ganavi et al., 2022, Efficient Image Steganography Using Bit-plane Slicing + ECC + Wavelet, MECS Press [6]	High-capacity image steganography using bit-plane slicing	ECC + Wavelet Transform + Bit-plane slicing	High payload capacity; good robustness	Complex implementation; longer processing time
Waheed Rehman, 2024, GAN-based Image Steganography, arXiv [7]	GAN-assisted stego with optional ECC integration	GAN-based image steganography	Can leverage deep learning for robustness	GAN training resource-intensive; ECC optional
Ramadhan J. Mstafa & Khaled M. Elleithy, 2023, ECC/DCT Video Steganography, River Publishers [8]	ECC in DCT domain for secure video steganography	ECC/DCT video stego	Robust against some attacks; confidentiality ensured	Computationally heavy; may affect video quality
SegNet + ECC + DCT + DL, 2023, High-Capacity Image Steganography Using ECC & DNN, DOAJ [9]	High-capacity image stego with ECC and deep neural networks	ECC + DCT	Very robust; high payload	High computation; deep learning expertise required
Ganavi et al., 2022, Bit-plane Slicing + ECC for Image Steganography, MECS Press [10]	Bit-plane slicing + ECC for improved image stego	ECC + Bit-plane slicing + LSB	Better robustness; moderate complexity	Payload limited by image size; not suitable for video

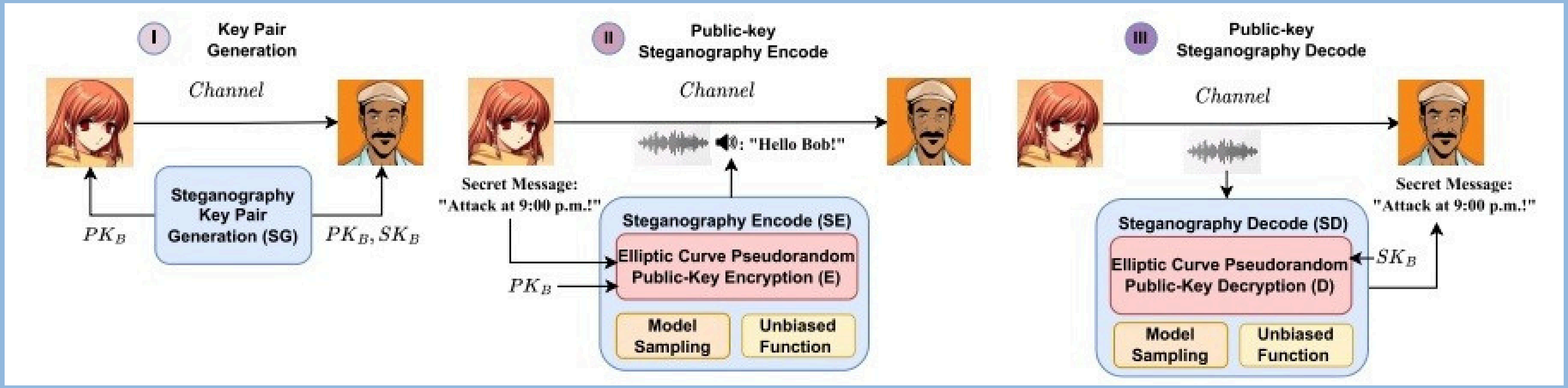
Overall Drawbacks

- **High computational complexity** – ECC operations and advanced stego techniques slow down processing.
- **Implementation difficulty** – Combining ECC with LSB, DWT, DCT, or deep learning is complex.
- **Limited payload capacity** – Some methods cannot hide large amounts of data.
- **Vulnerability to attacks** – Simple LSB or older schemes can be detected by steganalysis.
- **Resource-intensive** – Video steganography and GAN/DNN-based methods require high memory and processing power.

Project Objectives

- Provide **double-layer security** using ECC + Steganography.
- Hide encrypted data **without altering visible media quality**.
- Ensure **low computation with high security** (ECC advantage).
- Resist **cryptographic and steganographic attacks**.
- Evaluate system performance using **image quality and security metrics**.

Provably Secure Public-Key Steganography Based on Elliptic Curve Cryptography



Workflow

- Built a new public-key steganography system (hide secret messages in audio/text while looking normal).
- Used Elliptic Curve Cryptography (ECC) to make it secure.
- Designed a key exchange method using ECC that hides the exchange itself.
- Proved the system is secure using complexity theory + experiments (NIST randomness tests, steganalysis).

Technologies used

- Elliptic Curve Cryptography (ECC, Curve25519) → for encryption & security.
- Elligator2 encoding → hides ECC points as random-looking bits.
- Symmetric encryption (AES-like) → for message confidentiality.
- NIST randomness tests & steganalysis tools → to prove security.

Core Algorithms & WorkFlow

- The reference paper defines 5 key algorithms forming the ECC-based Public-Key Steganography system.

5 Key Algorithms

- Encryption / Decryption Core (A1–A2)
- Steganography Operations (A3–A5)

Pipeline Flowchart

[Key-Pair Generation (A3)]
↓
[ECC Encryption (A1)]
↓
[Stego Encode (A4)]
↓
[Transmitted Stego Cover]
↓
[Stego Decode (A5)]
↓
[ECC Decryption (A2)]
↓
[Recovered Message]

- Combines public-key security (ECC on Curve25519) with steganography for covert channels.
- Elligator2 hides ECC points → looks like random bits → resists detection.
- Achieves secure key generation, concealment of ciphertext in natural data, and reliable extraction & decoding by the receiver.

Algorithms

Algorithm 1: Elliptic Curve Pseudorandom Public-Key Encryption (E)
Input: $m \in \{0,1\}^*$, B_0, B', r, Q, $PK = x \cdot B'$
Output: c

1. **repeat**
2. $a \leftarrow U(0, rQ)$
3. $V = a \cdot B_0$
4. **until** $V \in E_r(F_p)$
5. $K = H(a \cdot PK) = H(a \cdot x \cdot B')$
6. $c_1 = \psi(V), c_2 = E_k(m), c = c_1 || c_2$
7. **return** c

Algorithm 2: Elliptic Curve Pseudorandom Public-Key Decryption (D)
Input: c, B_0, B', r, Q, $SK = x$
Output: m

1. Separate c into c_1, c_2
2. $V = \varphi(c_1)$
3. $K = H(a \cdot x \cdot B') = H(a \cdot x \cdot r \cdot B_0) = H(x \cdot r \cdot V)$
4. $m = D_k(c_2)$
5. **return m**

→ Generates random scalar r
→ Computes curve point R = r·B
→ Uses Elligator2 to map curve point to random-looking bits
→ Derives symmetric key from shared secret r·PK using hash H()
→ Encrypts plaintext message m with symmetric key K
→ Outputs ciphertext (R, c)

→ Receives ciphertext (R, c)
→ Computes shared secret S = x·R using private key x
→ Derives same symmetric key K = H(S)
→ Decrypts ciphertext c to recover message m
→ Verifies integrity and returns plaintext

Algorithm 3: Steganography Key Pair Generation (SG)
Input: $1^k \in U(|k|)$
Output: PK, SK

1. Given $E_{A,B}(x,y)$, find base point B_0 : Order(B_0) = N.
2. Compute $B' = r \cdot B_0$
3. **repeat**
4. $x \leftarrow U(0, Q)$, $V = x \cdot B'$
5. **until V** $\in E_r(F_p)$
6. **PK** = V, SK = x

→ Generates ECC private key x and public key PK = x·B′
→ Uses reversible mapping f() (Elligator2) to make public key indistinguishable from random bits
→ Produces steganographic key-pair (SK, PK′) usable in cover media
→ Ensures public key looks statistically random

Algorithm 4: Steganography Encode (SE)
Input: $m \in \{0,1\}^*$, B_0, B', r, Q, $PK = x \cdot B'$, h, f, G
Output: s

1. **repeat**
2. $a \leftarrow U(0, rQ)$, $V = a \cdot B_0$
3. until $V \in E_R(F_p)$
4. $K = H(a \cdot PK) = H(a \cdot x \cdot B')$
5. $c_1 = \psi(V)$, $c_2 = E_k(m)$, $c = c_1||c_2$, $s_0 = \{\}$
6. n = length(c), i = 0
7. **while** i < n **do**
8. $x = c[i : i + q]$
9. $C_h \leftarrow G(h)$, $C_h^x = f_h(x, C_h)$
10. $s \leftarrow C_h^x \cup C_h$
11. append s to s_0, append s to h, $i = i + q$
12. **end while**
13. **return** s_0

→ Takes input message m, base point B0, and recipient's public key PK
→ Generates random scalar a and computes V = a·B0
→ Checks if V lies in the reversible mapping set
→ Derives session key K = H(a·PK)
→ Encrypts message → embeds ciphertext into reversible mapped point
→ Outputs encoded stego-object s

Algorithm 5: Steganography Decode (SD)

Input: $s, B_0, B', r, Q, SK = x, C_h, f$

Output: m

1. $c = \{\}$
2. **for** each $x \in s_0$ do
3. $C_h \leftarrow G(h)$
4. $c = c||f_h^{-1}(x, C_h)$
5. **end for**
6. separate c into c_1, c_2
7. $V = \varphi(c_1)$
8. $K = H(a \cdot x \cdot B') = H(a \cdot x \cdot r \cdot B_0) = H(x \cdot r \cdot V)$
9. $m = D_k(c_2)$
10. **return m**

→ Receives stego-object s

→ Extracts embedded curve point and ciphertext

→ Computes shared secret using private key x

→ Derives session key $K = H(x \cdot V)$

→ Decrypts ciphertext to retrieve original message m

Tools and Platforms

- Implementing using Python(Google Colab)
- NumPy/Pillow/OpenCV for images
- Cryptography lib
- pytest for tests
- Matplotlib, Secrets
- GitHub Actions CI

Experiment Analysis
from the reference paper

Experimental Setup

- Platform: Implemented using Curve25519 for ECC.
- Cover Medium: Focused on image steganography.
- Security Evaluation:
- NIST Pseudorandomness Tests for ciphertext & embedded bits.
- Steganalysis Tools to detect presence of hidden data.
- Comparative Baselines: RSA-based and symmetric stego methods.

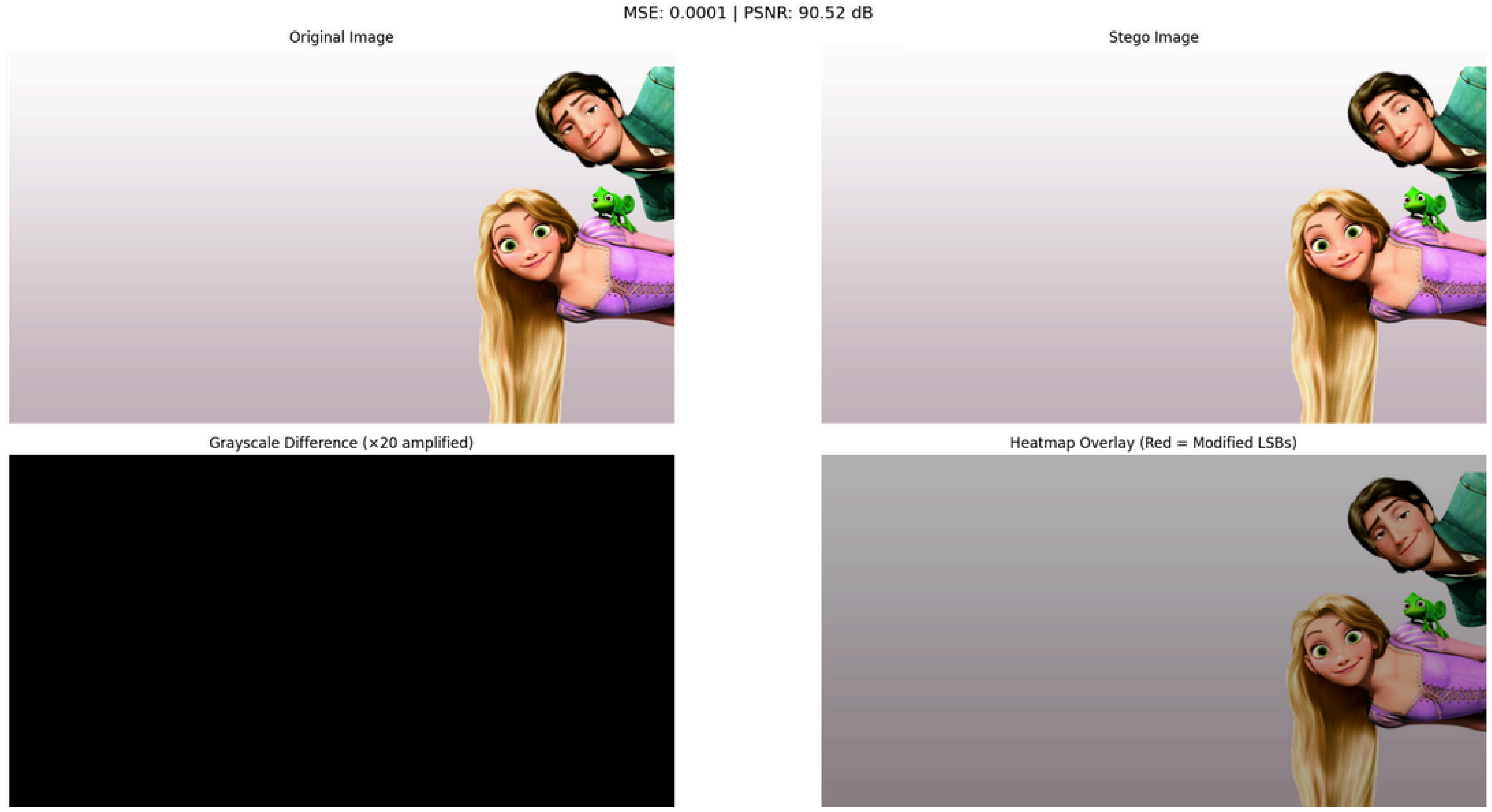
Key Findings

- Computational Efficiency:
  - ECC-based encryption is 3–4× faster than RSA/ElGamal at equivalent security.
  - Requires smaller key sizes (256-bit vs 3072-bit) → lower embedding overhead.
- Embedding Efficiency:
  - Less distortion in cover images due to smaller ciphertext size.
- Security Performance:
  - Passed NIST randomness tests → ciphertext statistically indistinguishable from random bits.
  - With Elligator2, ECC points appear random → harder to detect in image stego.
  - Resistant to Chosen-Hiddentext Attack (CHA) and steganalysis classifiers.
- Robustness in Image Stego:
  - When using reversible mapping and ECC-encrypted payloads,
  - detection accuracy by common steganalysis tools dropped close to 50% (random guess).

Implication for Our Capstone

- ECC + Elligator2 makes the embedded data smaller, harder to detect, and faster to process.
- Confirms suitability for secure image-based steganography in our project.

Result Analysis




```
Payload size: 84 bytes
Data hidden in stego_output.png
Recovered plaintext: b'THIS IS A SECRET MESSAGE'
```

```
Saved result images:
• difference_gray.png  – amplified grayscale diff
• difference_heatmap_colored.png  – color heatmap (HOT)
• difference_overlay.png  – heatmap blended with original

MSE: 0.000055 | PSNR: 90.75 dB
```

Applications

- Secure Communication – Safely transmit encrypted data hidden inside digital media.
- Defense & Intelligence – Enable covert, tamper-proof message exchange.
- Digital Watermarking – Protect copyrights and authenticate multimedia content.
- IoT & Cloud Security – Lightweight encryption for secure data transfer in connected devices.
- Medical Data Security – for transmitting sensitive patient information covertly.

Conclusion

Our project demonstrates that combining Elliptic Curve Cryptography (ECC) with steganography provides a highly secure and efficient means of concealing encrypted data within digital media. The use of pseudorandom elliptic curve point encoding ensures that hidden data remains undetectable and resistant to cryptographic or steganalysis attacks.

In the future, this work can be advanced by integrating deep learning–based adaptive embedding techniques, extending support to audio and video steganography, and implementing real-time secure communication systems using ECC-based hybrid encryption. These enhancements can make the system more robust, scalable, and practical for modern cybersecurity applications.

Reference

1. https://dl.acm.org/doi/10.1109/TIFS.2024.3361219
2. https://ijcesen.com/index.php/ijcesen/article/view/382
3. https://www.mecs-press.org/ijcnis/ijcnis-v15-n2/v15n2-3.html
4. https://link.springer.com/chapter/10.1007/978-981-13-3450-4_50
5. https://www.ijcaonline.org/archives/volume108/number3/18890-0172/
6. https://arxiv.org/abs/2412.00094
7. https://journals.riverpublishers.com/index.php/JCSANDM/article/view/5181
8. https://doaj.org/article/57f40792e5f34904a67020ed3ef53b92
9. https://ieeexplore.ieee.org/document/10418202/;jsessionid=652EE00E119D933A857A0778F6FB751A

THANK YOU