DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Visvesvaraya Technological University (VTU), Belagavi,
Approved by AICTE and UGC, Accredited by NAAC with 'A' grade & ISO 9001 – 2015 Certified Institution)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560 111, India

DEPARTMENT OF ELCTRONICS & COMMUNICATION ENGINEERING

(Accredited by NBA Tier 1: 2022-2025)

VI SEM BE MINI-PROJECT (22EC66)
Report on

A Hybrid Approach to Image Encryption and Authentication Using ECC-DH and 2-D Chaotic Map

Submitted in partial fulfillment for the award of the degree of

Bachelor of Engineering
in
Electronics & Communication Engineering

Submitted by

A B Vishvajeeth	**1DS22EC001**
Malcolm Cephas	**1DS22EC118**
Medhansh Jani	**1DS22EC126**
Shalini Sinha	**1DS22EC200**

Under the Guidance of
Dr. S Thenmozhi
Associate Professor
Department of Electronics and Communication Engineering
DSCE, Bengaluru

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI-590018, KARNATAKA, INDIA
2024-25

DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Visvesvaraya Technological University (VTU), Belagavi,
Approved by AICTE and UGC, Accredited by NAAC with 'A' grade & ISO 9001 – 2015 Certified Institution)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560 111, India

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

(Accredited by NBA Tier 1: 2022-2025)

CERTIFICATE

Certified that the **Mini-Project work (Course Code : 22EC66)** entitled **"A Hybrid Approach to Image Encryption and Authentication Using ECC-DH and 2-D Chaotic Map"** carried out by **A B Vishvajeeth** (1DS22EC001)**, Malcolm Cephas** (1DS22EC118)**, Medhansh Jani** (1DS22EC126), **Shalini Sinha** (1DS22EC20) are bonafide students of **DEPARTMENT of ECE, DAYANANDA SAGAR COLLEGE OF ENGINEERING**, Bengaluru, Karnataka, India an autonomous institution affiliated to VTU, Belagavi in partial fulfillment for the **VI Semester course** during the year **2024-2025**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the mini-project report. This **VI semester mini-project report** has been approved as it satisfies the academic requirements with respect to the mini-project work prescribed for the said Degree.

Signature of the Guide
Dr. S Thenmozhi
Associate Professor
Dept. of ECE, DSCE
Bengaluru

Signature of the HOD
Dr. Shobha.K.R
Dean IQAC, Professor & Head
Dept. of ECE, DSCE, Bengaluru

Signature of the Principal
Dr. B G Prasad
Principal
DSCE, Bengaluru

Name of the Examiners **Signature with date**

1. .... **..**

2. .... **..**

DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Visvesvaraya Technological University (VTU), Belagavi,
Approved by AICTE and UGC, Accredited by NAAC with ‘A’ grade & ISO 9001 – 2015 Certified Institution)
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560 111, India

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

(Accredited by NBA Tier 1: 2022-2025)

DECLARATION

We, **A B Vishvajeeth** (1DS22EC001), **Malcolm Cephas** (1DS22EC118), **Medhansh Jani** (1DS22EC126), **Shalini Sinha** (1DS22EC20), respectively, hereby declare that the mini-project work entitled “A Hybrid Approach to Image Encryption and Authentication Using ECC-DH and 2-D Chaotic Map” has been independently done by us under the guidance of **‘Dr. S Thenmozhi ’, Associate Professor,** ECE department and submitted in partial fulfillment of the requirement for the award of the degree of **Bachelor of Electronics & Communication Engineering** at **Dayananda Sagar College of Engineering**, an autonomous institution affiliated to VTU, Belagavi during the academic year 2024-2025 for the VI Semester Autonomous Course.

We, the students of VI Semester Mini-project group 6MP-56 do hereby declare that the entire mini-project has been done on our own. We further declare that we have not submitted this report either in part or in full to any other university for the award of any degree.

A B Vishvajeeth	**1DS22EC001**
Malcolm Cephas	**1DS22EC118**
Medhansh Jani	**1DS22EC126**
Shalini Sinha	**1DS22EC200**

PLACE: Bengaluru

DATE: 06/05/2025

Acknowledgement

It is our profound gratitude that we express our indebtedness to all who have guided us to complete this mini-project successfully. We extend our sincere thanks to the management of DSCE, for providing us with excellent infrastructure and facilities. We are thankful to our principal Dr. B. G. Prasad, for his encouragement and support. We are grateful to our HOD Dr. Shobha K. R for her valuable insights and guidance. We are thankful to our guide Dr. S Thenmozhi for her valuable guidance, exemplary support and timely suggestions throughout the journey of the mini-project. We would like to thank our Mini-Project Domain Coordinators – Dr. Shashi Raj K, Dr. Santhosh Kumar R, Dr. Yashaswini Gowda, and Prof. Kavita Guddad for their support and coordination. We sincerely acknowledge the Mini-Project Convener & Chief Coordinator Dr. Trupti Tagare for her help and constant support.

We also thank the teaching and non- teaching staff members of the Department of Electronics and Communication Engineering and also,our family and friends for the help and support provided by them in successful completion of the mini-project. Our accomplishments would be incomplete without my beloved parents, for without their support and encouragement we would not have reached up to this level. We express our gratitude to the Almighty for guiding us throughout this journey.

Thank you all.

A B Vishvajeeth
Malcolm Cephas
Medhansh Jani
Shalini Sinha

Abstract

With the rapid digitalization of healthcare systems, the secure storage and transmission of medical images has become critically important. Medical images often contain highly sensitive patient information, making them prime targets for data breaches and unauthorized access. Traditional encryption techniques are either computationally heavy or insufficient to protect image-specific data. To address this issue, this mini-project proposes a lightweight and secure image encryption and authentication scheme tailored specifically for grayscale medical images. The core of the model combines the Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol with 2D chaotic maps to provide both secure key generation and strong pixel-level scrambling.

The project begins by implementing the ECDH protocol to generate a shared secret session key between two parties. This key is then used to control the parameters of a 2D chaotic map, which scrambles the pixel positions and intensity values of the grayscale image, creating an encrypted version that is visually and statistically unrecognizable from the original. For authentication, a SHA-256 cryptographic hash is generated from the original image and transmitted securely. Upon decryption, the hash of the received image is compared to verify authenticity and detect tampering.

The expected outcomes include a highly secure encrypted image with high entropy, low correlation with the original image, and low computational overhead—making the solution viable for real-time medical applications. The scheme aims to provide an effective balance between security and efficiency, suitable for use in telemedicine, cloud-based storage, and secure image archiving systems.

Keywords: *image encryption, medical imaging, Elliptic Curve Diffie-Hellman (ECDH), 2D chaotic maps, SHA-256, authentication, grayscale image, secure transmission, lightweight cryptography, pixel scrambling*

Table of Contents

List of Figures

List of Tables

Nomenclature and Acronyms

Abbreviations (Alphabetical Order) :

AES	Advanced Encryption Standard
ASLT	Asymmetric Logistic-Tent Map
DSCE	Dayananda Sagar College of Engineering
ECDH	Elliptic Curve Diffie-Hellman
ECC	Elliptic Curve Cryptography
ECIES	Elliptic Curve Integrated Encryption Scheme
ECE	Electronics & Communication Engineering
GCM	Galois/Counter Mode
IEEE	Institute of Electrical & Electronics Engineers
IoT	Internet of Things
LZW	Lempel-Ziv-Welch
MSE	Mean Square Error
NIST	National Institute of Standards and Technology
NPCR	Number of Pixels Change Rate
PSNR	Peak Signal-to-Noise Ratio
RSA	Rivest, Shamir, Adleman
SSIM	Structural Similarity Index Measure
UCAI	Unified Average Changing Intensity

Chapter-1

Introduction

In the modern digital era, the secure transmission and storage of image data has become increasingly important, especially with the growth of online communication, cloud computing, and remote diagnostics. Conventional encryption algorithms often fall short in efficiently securing images due to their large data sizes, high redundancy, and strong correlation between adjacent pixels. This mini-project addresses these challenges by implementing a robust image encryption and authentication model tailored for grayscale images, using ECDH key exchange and 2D chaotic maps.

The project begins with the use of ECDH, which allows two parties to securely generate a shared secret over an insecure channel. This shared key is then hashed using SHA-256 to derive a fixed-length cryptographic key, enhancing both security and compatibility. This session key is used in the encryption process that combines chaotic scrambling using the 2D Arnold Cat Map and a modified ElGamal encryption scheme for high confusion and diffusion. The Arnold Cat Map's sensitivity to initial conditions makes it ideal for pixel-level scrambling, thereby increasing resistance to statistical analysis.

To ensure the integrity of the encrypted image, a cryptographic hash of the scrambled image is generated and verified at the receiver's end before decryption. This serves as a lightweight yet effective method for image authentication.

The performance of the encryption model is evaluated using metrics such as entropy, correlation coefficient, MSE, PSNR, SSIM, and UACI. The results show that the proposed model provides strong encryption with minimal loss of image quality and low computational overhead.

Overall, the mini-project demonstrates the feasibility and efficiency of combining ECC with chaotic systems to enhance the security of grayscale image transmissions, especially in scenarios where data confidentiality and integrity are critical.

Chapter 2

Literature survey

The literature review provides an overview of existing research related to image encryption techniques, especially those combining cryptography and chaos theory. It highlights key methods used for securing medical images and evaluates their effectiveness. This section helps identify current challenges and gaps, guiding the development of improved encryption strategies.

Table 2.1: Literature Review on Image Encryption using ECDH and Chaotic Maps

Journal / Year	Technology / Methodology	Evaluation Parameters	Remarks
"Image Encryption and Authentication With Elliptic Curve Cryptography and Multidimensional Chaotic Maps, 2021"[1]	ECDH, multidimensional Arnold Cat chaotic maps	Entropy, Correlation, PSNR, MSE, UCAI, SSIM, various attacks	More entropy, less time, less computational cost, resilience against attacks
"Implementation of Elliptic Curve Diffie Hellman (ECDH) Algorithm for Secured Communication, 2024"[2]	ECDH key encryption, 256 bit AES key algorithm, ECC	SSIM, PSNR, Encryption, Decryption, Bouncy Castle, Spongy Castle	Cryptographic attack resistant, High Security, Detailed images
"Elliptic Curve Diffie-Hellman (ECDH) Analogy for Secured Wireless Sensor Networks, 2020"[3]	ECDH, RSA, ECIES, PyCryptodome Library, NIST ECC Curves, AES-256-GCM	Wireless Sensor Network, Public key, Encryption, Decryption	Real Time Execution, Swift Key Generation, Easy Information transfer
"A Secure Transmission of Digital Images using Multiple Chaotic Maps and Elliptic Curve, 2024"[4]	Chaotic map, Hill cipher, Elliptic curve, Chebyshev map, ASLT Map	Entropy, Correlation Evaluation, SSIM, PSNR	Histogram Evaluation, Colour and B/W Image Encryption
"An Efficient 32-bit color image encryption technique using multiple chaotic maps and advanced ciphers, 2024"[5]	1-D Chaotic Maps: Logistic map, Sine map, Chebyshev map, Tent map	Efficacy & Resilience, Statistical, Differential & Entropy Based Attacks Analysis, MSE, PSNR	High Effectiveness, Strong Resistance Against Various Attacks, Highly Secure

"A Chaotic Based Image Encryption Scheme Using ECC And Genetic Algorithm, 2024"[6]	Arnold's Cat Map, Genetic Algorithm, ECC, NIST SP 800-22	Statistical Analysis - Histogram, Correlation, Entropy Analysis, Key Analysis, Differential Attacks, Robustness Analysis	Improved Security & Robustness, Enhanced Resistance Against Attacks, Efficient Key Management
"Diffie-Hellman Key Exchange Based on Block Matrices Combined with Elliptic Curves, 2023"[7]	ECDH with block matrices and elliptic curves	Key exchange efficiency, security strength	Enhances security using structured matrix-based computations
"Efficient Elliptic Curve Diffie–Hellman Key Exchange for Resource-Constrained IoT Devices, 2023"[8]	Optimized ECDH for IoT devices	Power consumption, execution time	Energy-efficient and lightweight for IoT security
"Improving Bounds on Elliptic Curve Hidden Number Problem for ECDH Key Exchange, 2022"[9]	Mathematical improvements to ECDH security	Attack resistance, computational complexity	Strengthens ECDH against hidden number attacks
"Unforgeable Digital Signature Integrated into Lightweight Encryption Based on Effective ECDH for Cybersecurity Mechanism in IoT, 2022"[10]	Integration of ECDH-based key exchange with a lightweight encryption mechanism and digital signature verification	CPU execution time, storage cost, power consumption	Provides improved efficiency in IoT security through fast cryptographic operations and reduced computational cost

"Efficient Medical Image Security and Transmission using Modified LZW Compression and ECDH-AES for Telemedicine Applications, 2023"[11]	Hybrid cryptographic approach combining ECDH key exchange, AES encryption, and modified LZW compression for secure medical image storage	Key generation time, encryption time, throughput improvement	Enhances medical data security and storage efficiency, making it suitable for edge devices with limited resources
"An Integrated Image Encryption Scheme Based on Elliptic Curve, 2023"[12]	Integrated image encryption using elliptic curve cryptography (ECC), Diffie-Hellman key exchange, SHA-256 hashing, and affine power affine transformation for confusion and diffusion	Key space analysis, PSNR (Peak Signal-to-Noise Ratio), NPCR (Number of Pixels Change Rate), UACI (Unified Average Changing Intensity), correlation analysis, entropy analysis, execution time, robustness against occlusion and noise attacks	The system ensures large key space, resistance to differential attacks, and strong statistical performance while efficiently encrypting RGB images with low computational overhead
"A Novel Image Encryption Scheme Using Chaotic Maps and Fuzzy Numbers for Secure Transmission of Information, 2023"[13]	Hybrid chaotic maps (logistic sine map, fuzzy Hénon map) with fuzzy triangular membership function	Statistical attack analysis, NIST tests, differential attack, entropy, brute force attack, key sensitivity	Offers high security, key sensitivity, and robustness against noise and data loss
"A Secure Transmission of Digital Images using Multiple Chaotic Maps and Elliptic Curve, 2024"[14]	Hybrid chaotic maps (ASLT, Chebyshev) with elliptic curve cryptography (ECC) and Hill cipher	Entropy, histogram analysis, correlation coefficients, key sensitivity, brute force attack resistance	Offers high security, efficient encryption, and robustness against statistical and differential attacks

Table 2.1: Literature Review on Image Encryption using ECDH and Chaotic Maps

Chapter 3

3.1 Objectives

The main objective of this mini-project is to design and implement a secure, efficient, and lightweight image encryption and authentication scheme suitable for grayscale images. The scheme integrates elliptic curve-based cryptographic key exchange with 2D chaotic maps to provide robust security while maintaining low computational complexity.

Specific objectives include:

1. To establish a secure and confidential key exchange mechanism using ECDH and derive a fixed-length cryptographic key with SHA-256.
2. To enhance image security through 2D Arnold Cat Map-based pixel scrambling combined with a lightweight encryption algorithm.
3. To ensure data integrity and validate encryption robustness using cryptographic hash functions and evaluation metrics such as entropy, correlation, PSNR, MSE, SSIM, and UACI.
4. To demonstrate the feasibility of the proposed encryption approach on sample medical image datasets, laying the groundwork for future application in domains like surveillance, healthcare, and cloud storage.

3.2 Problem Statement

With the rapid growth of digital communications and extensive dissemination of multimedia data over public and private networks, image data protection has emerged as a major issue. In contrast to text data, images hold high redundancy and pixel correlations, rendering conventional encryption methods like normal block ciphers less efficient and computationally demanding when used directly. Moreover, new applications like telemedicine, military monitoring, and image storage in the cloud need encryption techniques that are not only highly secure but also lightweight and applicable for real-time processing.

Current encryption techniques either sacrifice computational performance or are not sufficiently resilient against statistical, brute-force, and differential attacks. There is an increasing demand for hybrid cryptographic methods that combine the mathematical resilience of public-key cryptography with the unpredictability and sensitivity of chaotic systems.

Hence, this mini-project defines the problem as:

"*To develop a secure, efficient, and lightweight grayscale image encryption scheme that utilizes elliptic curve-based key exchange (ECDH) for secure session key generation, integrates a 2D Arnold Cat Map for effective pixel confusion, and employs SHA-256 hashing for integrity verification, while ensuring low computational overhead and strong resistance against common cryptographic attacks.*"

This problem has been framed after a thorough study of recent advancements in cryptographic research, especially those involving chaos theory and elliptic curve cryptography, with the goal of bridging the gap between high security and low resource consumption in image encryption systems.

Chapter 4
Methodology, Block diagram & Implementation

This chapter outlines the methodology adopted for the development and implementation of the proposed secure image encryption and decryption scheme. The aim of the project was to design a system capable of securely transmitting image data by combining several advanced cryptographic and transformation techniques. These include spatial scrambling using the Arnold Cat Map (ACM), symmetric encryption with AES, secure key exchange using Elliptic Curve Diffie-Hellman (ECDH), and integrity verification via SHA-256 hashing.

4.1 Block Diagram and Description

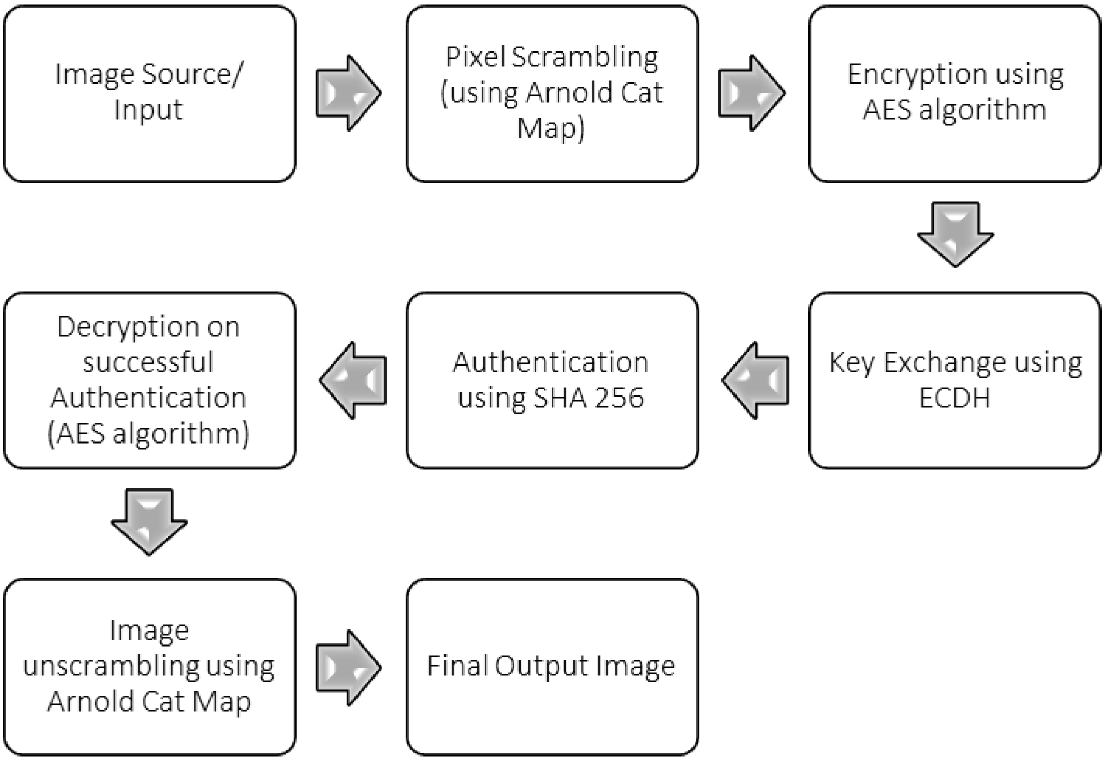


Image Source/ Input
Pixel Scrambling (using Arnold Cat Map)
Encryption using AES algorithm
Key Exchange using ECDH
Authentication using SHA 256
Decryption on successful Authentication (AES algorithm)
Image unscrambling using Arnold Cat Map
Final Output Image

Fig.4.1 Block Diagram of the Proposed Methodology

Fig 4.1 shows the block diagram of the methodology. As an input CT scan image is taken in the form of diacom, jpg, png and is then scrambled with the help of 2-D Arnold Cat Map, the scambled image is fed into AES Encryption algorithm (Galois/Counter Mode - GCM), with added padding for security purposes, this yields to the encrypted file in the enc format, then the private key for both the sender and the receiver is generated and sent or processing to the ECDH Key Exchange algorithm, where with the generator bits and the commonly agreed prime number is used to make the public key without revealing the private key of both individuals involved. Authentication process starts and the file is now ready to be sent to the receiver, the SHA 256 algorithm generates the hash string using the user data, password and image bytes. The Hex String is then authenticated and is further sent to Decryption, unscrambling which yield in the final output

Pseudo Code

```
Pseudo code
 //  Input : CT Scan Image -Load image from local system,  Output: Encrypted_Image
//  Image Scrambling using Arnold Cat Map
   Scrambled_Image ← ApplyArnoldCatMap(Original_Image, iterations)
//  AES Encryption
   AES_Key ← GenerateRandomKey()
   Encrypted_Image ← AES_Encrypt(Scrambled_Image, AES_Key, GCM_Mode)
//  Key Exchange using ECDH
   (Sender_PrivateKey, Sender_PublicKey) ← GenerateECDHKeyPair()
   Receiver_PublicKey ← ReceivePublicKey()
   Shared_Secret ← ECDH_DeriveSharedKey(Sender_PrivateKey, Receiver_PublicKey)
   AES_Key ← DeriveAESKey(Shared_Secret) // Used in step above
//  Hashing and Authentication using SHA-256
   Image_Hash ← SHA256(Scrambled_Image)
   Transmit (Encrypted_Image, Image_Hash, Sender_PublicKey)

// Receiver Side: Input: Encrypted_Image,  Output:  Decrypted_Image
// Authentication and AES Decryption
   (Encrypted_Image, Image_Hash, Sender_PublicKey) ← ReceiveTransmission()
   (Receiver_PrivateKey, Receiver_PublicKey) ← GenerateECDHKeyPair()
   Shared_Secret ← ECDH_DeriveSharedKey(Receiver_PrivateKey, Sender_PublicKey)
   AES_Key ← DeriveAESKey(Shared_Secret)
   Scrambled_Image_Received ← AES_Decrypt(Encrypted_Image, AES_Key, GCM_Mode)
   Computed_Hash ← SHA256(Scrambled_Image_Received)
   If Computed_Hash == Image_Hash Then
     Proceed with Decryption
   Else
     Reject transmission (authentication failed)
// Image Unscrambling
   Decrypted_Image ← ApplyInverseArnoldCatMap(Scrambled_Image_Received, iterations)
   Output: Decrypted_Image
End
```

4.2 Flowchart of the Methodology

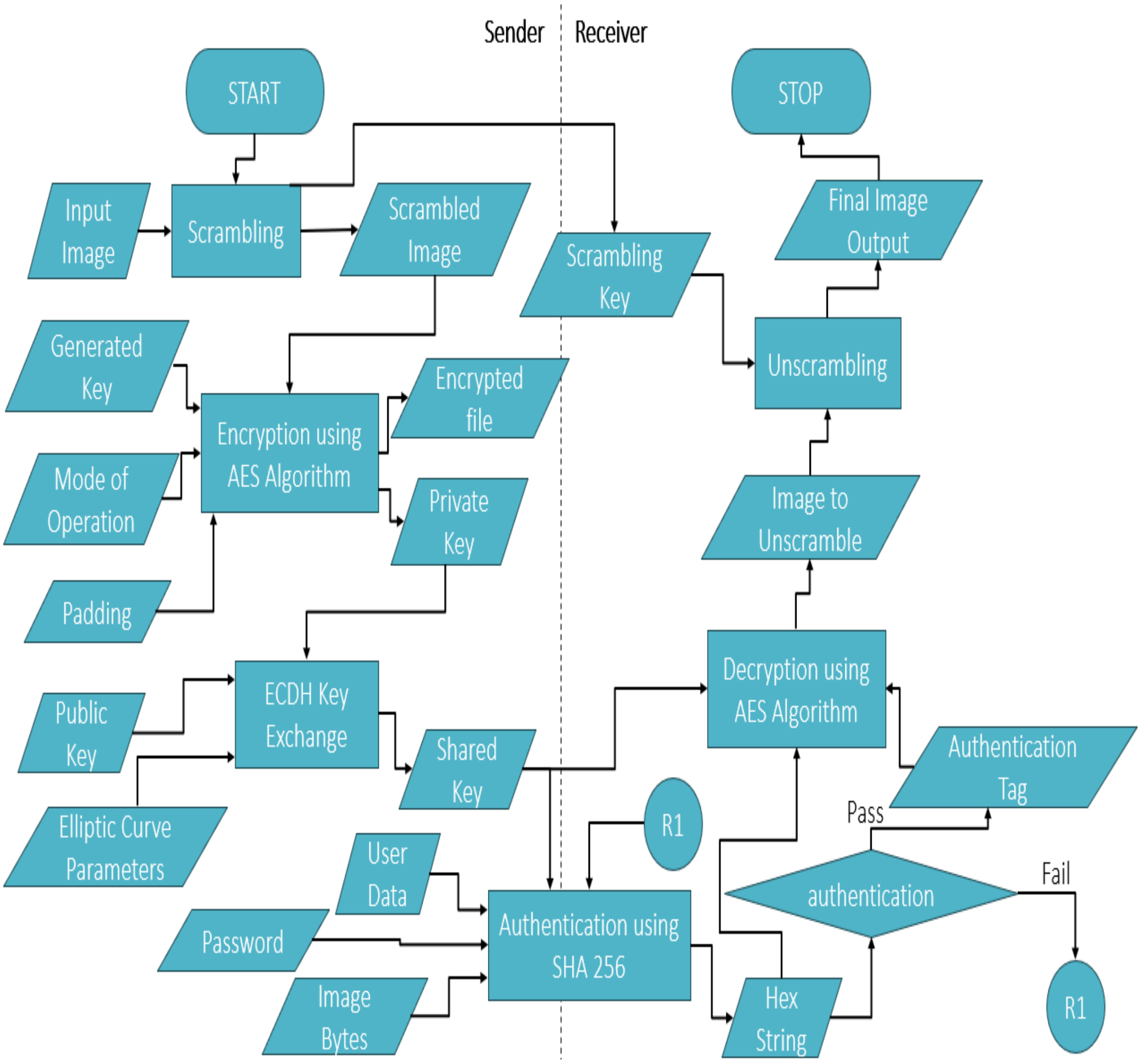


Fig. 4.2 : Flow-chart of the methodology used

Figure 4.2 illustrates the block diagram of the proposed methodology. The process begins with the input of a CT scan image, which may be in DICOM, JPG, or PNG format. The image undergoes scrambling using a 2D Arnold Cat Map to enhance visual security. The scrambled image is then encrypted using the Advanced Encryption Standard (AES) in Galois/Counter Mode (GCM), with appropriate padding added to enhance security. This results in an encrypted output file in .enc format.

Simultaneously, private keys for both the sender and the receiver are generated and processed through the Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol using private key given by AES. Using predefined generator values and a mutually agreed prime number, public keys are computed without disclosing the corresponding private keys, thus enabling secure symmetric key generation.

Following key exchange, the system initiates an authentication phase. A SHA-256 hash is generated for the scrambled image, ensuring integrity verification before decryption. Once validated, the encrypted file is ready for secure transmission to the receiver.

At the receiver's end, the decryption process begins by receiving the encrypted file (.enc format), along with the transmitted public key and the SHA-256 hash. The receiver uses their private key(any key of their choice) and the sender’s public key within the ECDH protocol to independently derive the same shared secret used for AES key generation, without revealing any private keys.

Using this derived AES key, the encrypted file is decrypted via the AES algorithm in Galois/Counter Mode (GCM). Before proceeding further, the system re-generates the SHA-256 hash of the decrypted scrambled image and compares it with the received hash. If both hashes match, it confirms the integrity and authenticity of the received data.

Subsequently, the scrambled image is subjected to the inverse 2D Arnold Cat Map transformation to restore the original spatial arrangement of the pixels. The final output is the original CT scan image, securely reconstructed and ready for viewing or further analysis.

4.3 Working Principle

The mini-project incorporates a multi-layered security approach that blends chaotic image scattering, AES encryption, ECDH key exchange, and SHA-256 hashing for safe and legitimate image transmission. The process starts with image scrambling using the Arnold Cat Map, a reversible chaotic map that rearranges pixel coordinates to hide the visual pattern without modifying pixel values. This mixed image is then encrypted using AES in GCM mode, which gives both data confidentiality and built-in authentication with a 256-bit key. The AES key is not transmitted in the plaintext but instead securely derived using the Elliptic Curve Diffie-Hellman (ECDH) protocol, so both sender and receiver can derive the same shared secret using their own public-private key pair. This shared secret is then hashed using a SHA-256-based PBKDF2 function to create the final AES key. For integrity, a SHA-256 hash of the scrambled image is computed before encryption and then verified after decryption to make sure that any tampering can be detected. During the decryption phase, the scrambled image is decrypted using the derived AES key, verified using the stored hash, and then unscrambled using the reverse Arnold Cat Map to re-restore the original image. The multi-step architecture offers strong encryption, key safeguarding without direct communication, and tamper detection and renders the system secure and robust.

4.4 Evaluation Parameter

Entropy: Entropy is a measure to find the information content used in an image as per equation 1, a high entropy number indicates that the image is complex with a wide range of pixel values, whereas a low entropy value denotes a more straightforward and uniform image.

$$H = -\sum_{i-1}^{n} p_i \log_2 p_i \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Where p_i is the probability of the pixel value i, and n is the total number of pixels.

In image processing, entropy is between 0 and 8 for 8-bit gray-scale images. It is 0 for an image that is totally uniform with no pixel intensity variation and 8 for the maximum randomness with all pixel values having equal probability. High values of entropy (generally between 7.5 and 8) are most suitable for applications such as image encryption, as they mean high complexity and uncertainty. On the contrary, low values of entropy (less than 3) imply a uniform or simple image having low information, which can prove useful in cases such as image compression where redundancy is desirable. Values from 4 to 7 offers moderate level of complexity.

SSIM: SSIM is a comparative metric that calculates the similarity between two images by comparing their structural information, luminance and contrast as per equation 2. It is from 0 to 1, with 1 being perfect similarity. In contrast to simple measures such as MSE or PSNR, SSIM is more in keeping with human visual perception and widely applied for assessing image quality, particularly after compression, transmission, or encryption.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where:

μ_x and μ_y are the average luminance of the two images being compared.

σ_x^2 and σ_y^2 are the variances of the pixel intensities in the two images.

σ_{xy} is the covariance of the pixel intensities between the two images.

C_1 and C_2 are the constants added to prevent instability when the denominators are close to zero

Correlation: Correlation in imaging processing is a term used for sliding a filter (or kernel) across an image and calculating the sum of the element-wise products between the filter and the region of the underlying image at every position as per equation 3. This operation finds application in feature detection, edge enhancement, and template matching.

$$G(i,j) = \sum_{m=-k}^{k} \sum_{n=-l}^{l} I(i+m, j+n).H(m,n) \dots\dots\dots\dots\dots\dots\dots (3)$$

Where

G(i,j): output image at position (i,j)

I(i+m,j+n): pixel value from the image input

H(m,n): Filter of size (2k+1) x (2l+1)

m,n: Indices over the filter window

Correlation values in image processing can range from –1 to 1 when normalized. A value of 1 indicates a perfect match, 0 means no similarity, and –1 signifies a perfect inverse match. Unnormalized correlation values vary depending on the image and filter, with positive values indicating similarity and negative values indicating dissimilarity.

UACI: UACI calculates the mean variation in the intensity of pixels between the original and transformed images, normalized over the 0 to 100% range as per equation 4 . The higher the UACI, the greater the difference, which is usually preferable in image encryption (for better security). A lower UACI value indicates that the image transformation is weaker, maintaining more of the original material.

$$UACI = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{|I(i,j) - I'(i,j)|}{255} \times 100 \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Where,

I(i,j) is the pixel value at position (i,j) in the original image

I′(i,j) is the pixel value at position (i,j) in the processed image

M and N are the dimensions of the image

The 255 normalizes the pixel values

NPCR: In image processing, NPCR (Number of Pixel Change Rate) is a measure used to evaluate how sensitive an encrypted image is to small modifications in the original plaintext image, particularly in image encryption algorithms as per equation 5. High NPCR indicates that a slight change in the input image, like changing a single pixel, results in large changes in the encrypted image. This is a good quality since it signifies strong encryption and enhanced security.

$$NCPR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\% \ldots\ldots\ldots\ldots\ldots\ldots\ldots (Eq. 5)$$

Where:

$$D(i,j) = \begin{cases} 0 & \text{if } I(i,j) = K(i,j) \\ 1 & \text{otherwise} \end{cases}$$

I and K are two ciphertext images with keys differ by one bit. M and N are the dimensions of the images. i and j are pixel positions within the images.

Chapter-5

Software tools Used

In this chapter, a brief description of the hardware and software components used for the successful execution of the mini-project work is provided. The project, focused on secure image encryption using elliptic curve cryptography and chaotic maps.

5.1 Software :

The software tool used for the mini-project work is (list out the relevant ones used)

- Spyder IDE – An integrated development environment suitable for scientific computing and Python scripting, used for writing and debugging the encryption and decryption algorithms.
- Anaconda Interpreter – A distribution of Python and R for scientific computing, which provided a stable and versatile platform to run the encryption scheme.
- Anaconda Environment Manager – Used for creating isolated environments to manage packages and dependencies specific to the project.
- Python Libraries: Several Python libraries were utilized for implementing the cryptographic algorithms and performing necessary mathematical operations. These included:
    - Tkinter – Provided a simple graphical user interface (GUI) for file selection and user interaction through dialog windows.
    - Cryptography – Enabled secure cryptographic operations including elliptic curve key exchange (ECDH), hashing algorithms, key derivation functions (PBKDF2), and AES encryption/decryption using cipher modes.
    - OS – Handled file path management and random salt/key generation for encryption.
    - NumPy – Supported complex matrix manipulations and efficient numerical operations throughout the encryption and decryption process.
    - OpenCV (cv2) – Used for image loading, resizing, and conversion to compatible formats for encryption routines.
    - Scikit-Image (SSIM) – Computed structural similarity index to evaluate the visual closeness of decrypted images to original ones.

  - SciPy (Pearson Correlation) – Measured statistical similarity between image data sets to assess encryption effectiveness.
  - Collections (Counter) – Aided in entropy and histogram analysis by counting pixel occurrences in encrypted and original images.
  - Math – Assisted in performing mathematical calculations for statistical evaluation metrics like entropy and UACI..
- Version Control: Git was used to manage the code and track changes, ensuring smooth development progress and collaboration.
- Operating System: The project was developed and executed on a Windows environment, though it is expected to be portable across Linux and macOS systems as well.

“These tools, including Python libraries like NumPy, Matplotlib, and PyCryptodome, as well as Git for version control, provided a reliable and efficient development platform for implementing, testing, and visualizing the proposed cryptographic scheme."

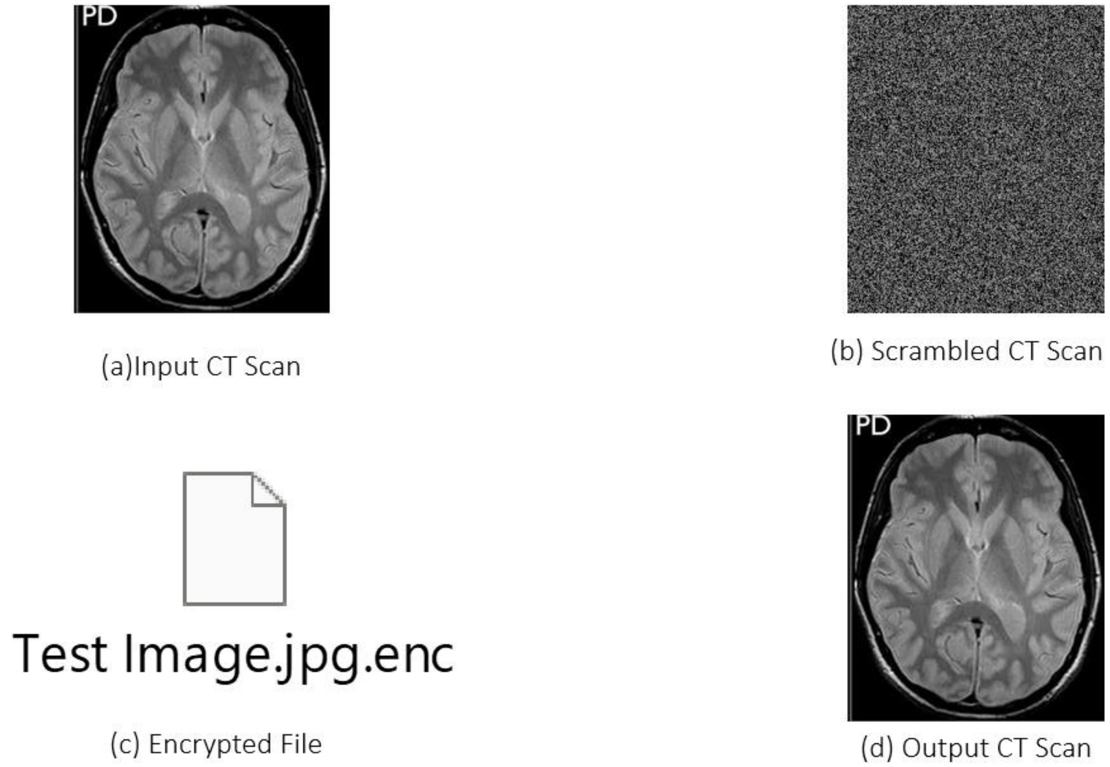
Working : The mini-project works on several well-established stages to maintain safe image transfer. First, the user picks an image from a graphical interface. The picture is scrambled in the first instance with the Arnold Cat Map, which breaks up pixel locations to mask structure while leaving values intact. Then the scrambled picture is encrypted with AES (Advanced Encryption Standard), which has a secret key. This secret key is not sent directly but is calculated through the use of the Elliptic Curve Diffie-Hellman protocol, where sender and receiver each calculate a common secret without its transmission. SHA-256 algorithm is also utilized on the encrypted image to create a hash to be used in authentication and verification. In decryption, when authentication is successful, the image is decrypted via AES, then de-scrambled using the inverse Arnold Cat Map, giving the original image. All the phases work together towards layered security through the use of spatial transformation, encryption, key exchange, and integrity checks.

Chapter-6

Results and Discussions

A hybrid approach to image encryption and authentication using ECC-DH and a 2D chaotic map algorithm was implemented in Spyder using the Anaconda interpreter. The results obtained from the execution are presented below.

6.1 Simulation Results



(a)Input CT Scan

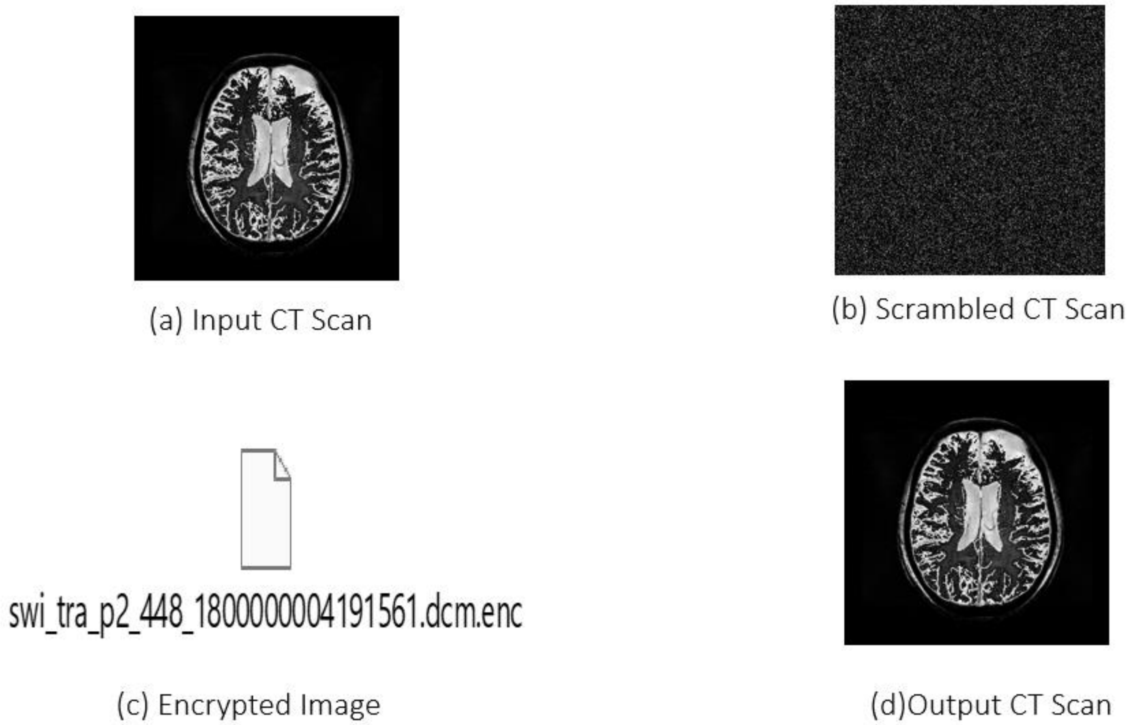
(b) Scrambled CT Scan

(c) Encrypted File

(d) Output CT Scan

Fig 6.1 : Output Images of jpg format

Fig. 6.1 depicts the step-by-step conversion of a CT scan image in jpg format using encryption and decryption process. Fig. 6.1 (a) represents the original input CT scan, which is initially sent to chaotic scrambling using Arnold Cat Map, yielding the visually unclear image depicted in Fig. 6.1 (b). This operation successfully conceals the spatial structure of the image but maintains pixel values. The scrambled image is subsequently encrypted in AES-GCM, generating the encrypted file "Test Image.jpg.enc," as indicated in Fig. 6.1 (c). This file cannot be read and is secure because AES strong encryption is employed along with secure key exchange through ECDH. The last is Fig. 6.1 (d), indicating the output CT scan after inverse scrambling and decryption. It correctly recovers the original image, showing the lossless nature and strength of the encryption-decryption system.



(a) Input CT Scan

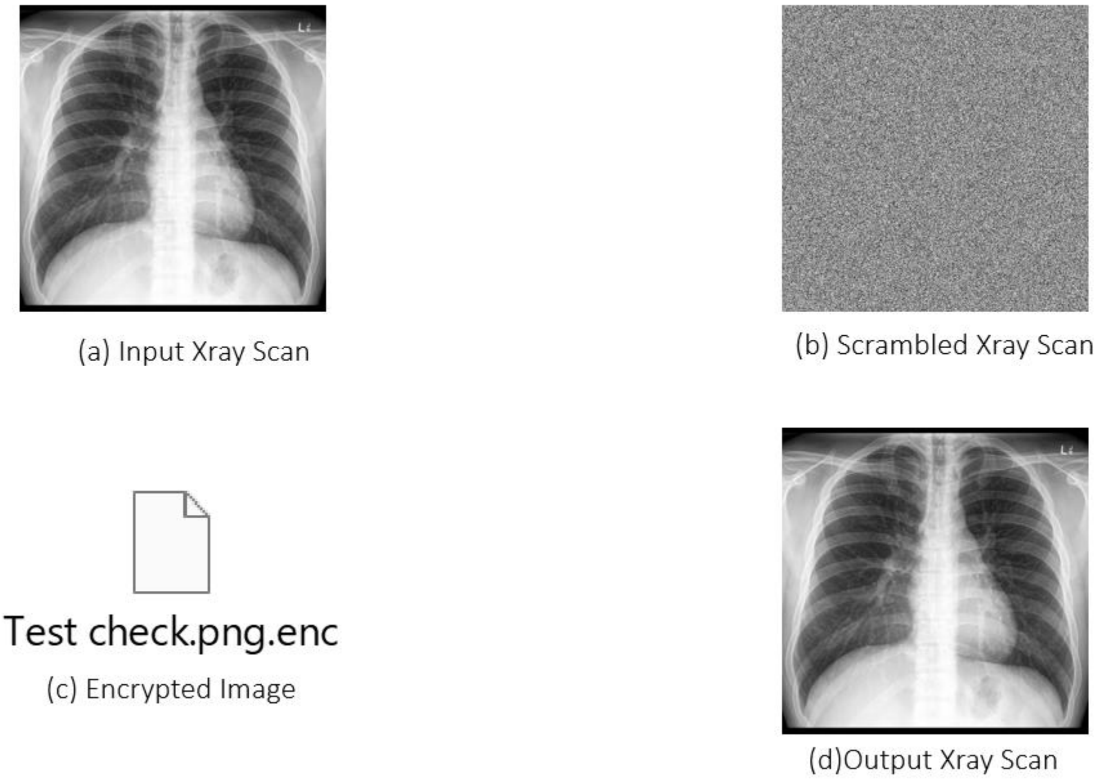
(b) Scrambled CT Scan

(c) Encrypted Image

(d)Output CT Scan

Fig 6.2: Output Images of Diacom Format

Fig. 6.2 shows how a CT scan image in diacom format being securely protected and later recovered. In Fig. 6.2 (a), we see the original CT scan image. This image is first scrambled using a method called the Arnold Cat Map, which jumbles the pixels so that the image becomes hard to recognize, as shown in Fig. 6.2 (b). Then, the scrambled image is encrypted using AES with a key safely shared using ECDH, resulting in the unreadable file seen in Fig. 6.2 (c). Finally, the image is decrypted and unscrambled to get back the original CT scan, as shown in Fig. 6.2 (d). This process helps keep the image safe from unauthorized access and ensures it hasn't been tampered with.



(a) Input Xray Scan

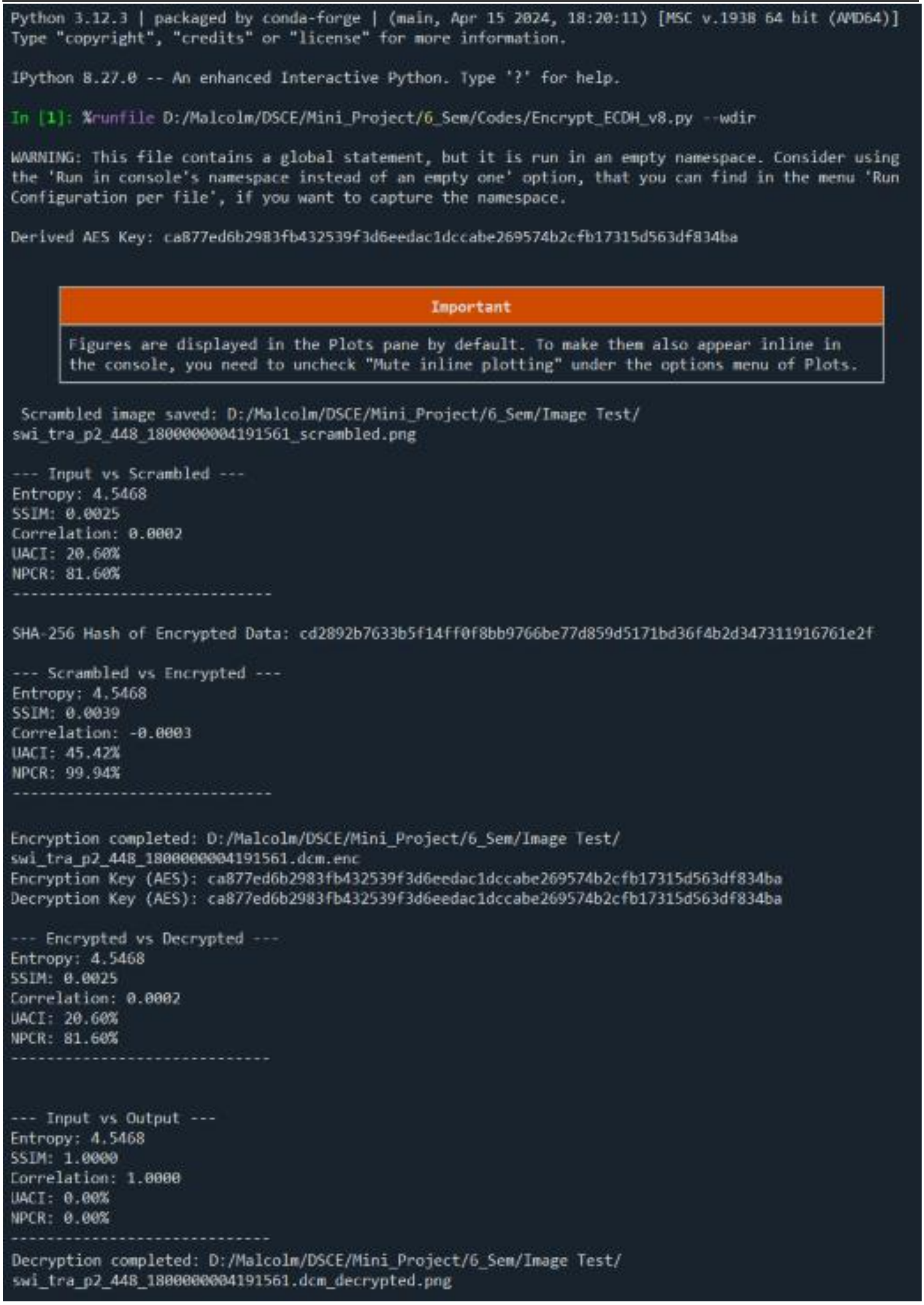
(b) Scrambled Xray Scan

(c) Encrypted Image

(d)Output Xray Scan

Fig 6.3: Output Images of png Format

Fig. 6.3 shows the full process of securing and recovering an X-ray scan image. Fig. 6.3(a) shows the original X-ray image before any changes. First, the image is scrambled using a method that mixes up the pixels, making it look like random noise, as shown in Fig. 6.3(b). This scrambled image is then encrypted using a strong method (AES), creating a secure file as seen in Fig. 6.3(c), which cannot be understood without the correct key. Finally, the encrypted file is decrypted and unscrambled to bring back the original X-ray image, shown in Fig. 6.3(d). This process helps keep medical images private and safe from tampering.



```
Python 3.12.3 | packaged by conda-forge | (main, Apr 15 2024, 18:20:11) [MSC v.1938 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: %runfile D:/Malcolm/DSCE/Mini_Project/6_Sem/Codes/Encrypt_ECDH_v8.py --wdir

WARNING: This file contains a global statement, but it is run in an empty namespace. Consider using
the 'Run in console's namespace instead of an empty one' option, that you can find in the menu 'Run
Configuration per file', if you want to capture the namespace.

Derived AES Key: ca877ed6b2983fb432539f3d6eedac1dccabe269574b2cfb17315d563df834ba


         Important
 Figures are displayed in the Plots pane by default. To make them also appear inline in
 the console, you need to uncheck "Mute inline plotting" under the options menu of Plots.

 Scrambled image saved: D:/Malcolm/DSCE/Mini_Project/6_Sem/Image Test/
swi_tra_p2_448_1800000004191561_scrambled.png

--- Input vs Scrambled ---
Entropy: 4.5468
SSIM: 0.0025
Correlation: 0.0002
UACI: 20.60%
NPCR: 81.60%
-----------------------------

SHA-256 Hash of Encrypted Data: cd2892b7633b5f14ff0f8bb9766be77d859d5171bd36f4b2d347311916761e2f

--- Scrambled vs Encrypted ---
Entropy: 4.5468
SSIM: 0.0039
Correlation: -0.0003
UACI: 45.42%
NPCR: 99.94%
-----------------------------

Encryption completed: D:/Malcolm/DSCE/Mini_Project/6_Sem/Image Test/
swi_tra_p2_448_1800000004191561.dcm.enc
Encryption Key (AES): ca877ed6b2983fb432539f3d6eedac1dccabe269574b2cfb17315d563df834ba
Decryption Key (AES): ca877ed6b2983fb432539f3d6eedac1dccabe269574b2cfb17315d563df834ba

--- Encrypted vs Decrypted ---
Entropy: 4.5468
SSIM: 0.0025
Correlation: 0.0002
UACI: 20.60%
NPCR: 81.60%
-----------------------------


--- Input vs Output ---
Entropy: 4.5468
SSIM: 1.0000
Correlation: 1.0000
UACI: 0.00%
NPCR: 0.00%
-----------------------------
Decryption completed: D:/Malcolm/DSCE/Mini_Project/6_Sem/Image Test/
swi_tra_p2_448_1800000004191561.dcm_decrypted.png
```

Fig. 6.4: Output observed from execution

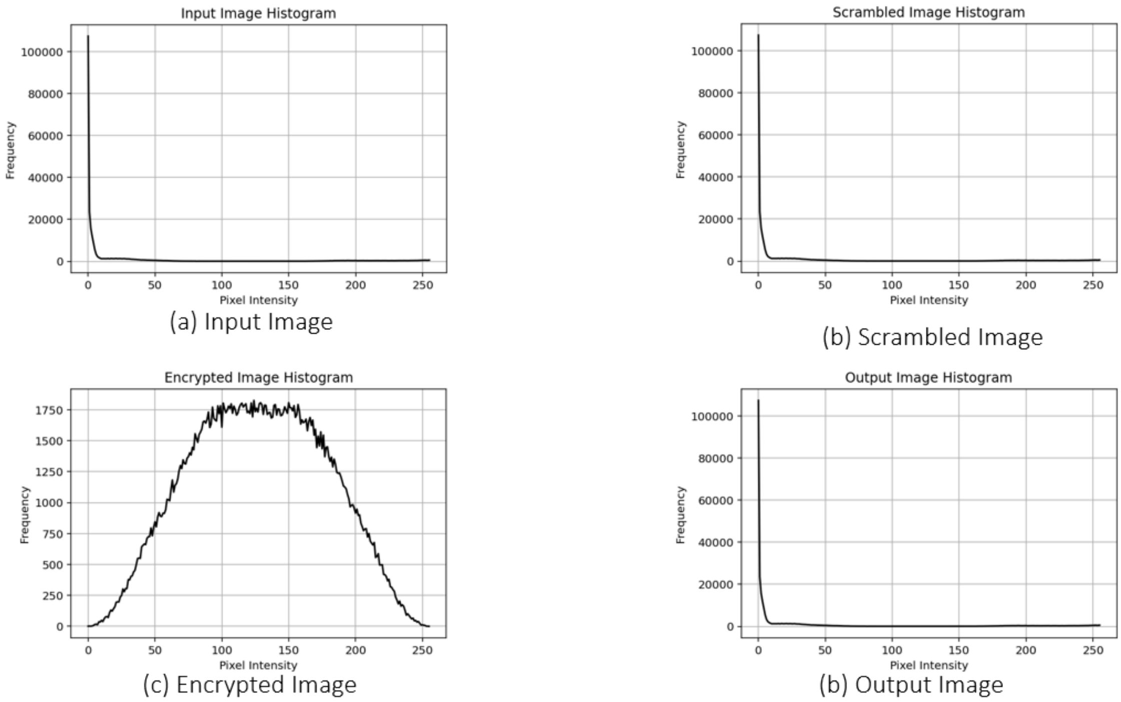
Fig. 6.4 shows the result of a Python script encrypting and decrypting a medical image using AES encryption. It starts by generating an AES key, followed by scrambling the input image and saving it. The figure indicates various comparison metrics across different steps in the process. The comparison of input vs scrambled indicates high randomness and low similarity, ensuring good scrambling. The scrambled vs encrypted comparison reveals additional disruption of image data. Lastly, the input vs output (decrypted) comparison reveals a perfect match (SSIM = 1.0, NPCR = 0%), which means that the decryption restored the original image perfectly without any loss, confirming the correctness of the encryption-decryption process.

Table 6.1 Comparison of Image Quality Metrics at Different Stages of Encryption and Decryption

Parameters	Input vs Scrambled Image	Scrambled vs Encrypted	Encrypted vs Decrypted	Input vs Output Image
Entropy	4.5468	4.5468	4.5468	4.5468
SSIM	0.0025	0.0039	0.0025	1.0000
Correlation	0.0002	-0.0003	0.0002	1.0000
UACI	20.60%	45.42%	20.60%	0.00%
NPCR	81.60%	99.94%	81.60%	0.00%

Table 6.1 illustrates the comparisons of various stages of the encryption and decryption process with various evaluation metrics. The entropy is the same for all comparisons (4.5468), demonstrating that there is an equal amount of information content throughout. The SSIM metrics are extremely low (nearly 0) for all but the last comparison (Input vs Output), with an SSIM of 1.0000, indicating ideal reconstruction upon decryption. The correlation is also close to zero or negative between stages, as desired for safe encryption, whereas the value is 1.0000 for the final output, again establishing perfect recovery. The UACI and NPCR are both high for scrambled and encrypted comparisons, indicating strong differences (which is desirable for security), and both are 0.00% for Input vs Output, indicating that the original image was completely recovered without any pixel alteration. This table proves that the encryption process is robust and the decryption is precise.

6.2 Histogram Analysis:



(a) Input Image

(b) Scrambled Image

(c) Encrypted Image

(b) Output Image

Fig 6.5: Histogram Analysis

Fig 6.5 presents the histograms of the image at four stages: input, scrambled, encrypted, and output. Fig 6.5 (a), the histogram of the input image is very dense at the lower intensity values, typical for medical images such as X-rays and CT Scans. Fig 6.5 (b), the scrambled image histogram remains similar to the input histogram because scrambling only alters pixel positions, not values. Fig 6.5 (c), the histogram of the encrypted image looks flat and level, which is perfect—it indicates that the pixel values have been distributed evenly, so it will be hard for an attacker to obtain any useful information. Last but not least, Fig 6.5 (d), the output image histogram is very similar to the input, which assures us that the original image has been restored completely after decryption.

Chapter-7

Applications, Advantages, Outcome and Limitations

The proposed image encryption scheme using ECC and 2D chaotic maps can be effectively applied in scenarios requiring secure image transmission, such as:

- Secure sharing of medical images over telemedicine networks
- Military and surveillance image protection
- Confidential biometric data exchange (e.g., fingerprints, facial scans)
- Cloud-based image storage with privacy preservation
- Secure communication in IoT environments involving camera-enabled devices

Advantages of the mini-project work include:

- High-level image security using ECC and chaotic scrambling
- Lightweight and efficient key exchange with minimal computational overhead
- Strong pixel distribution achieved through entropy-enhancing scrambling
- SHA-256-based image authentication for integrity verification
- Compatibility with grayscale images, ensuring applicability to wide datasets

Outcome of the Mini-Project Work

The outcome of this mini-project is a robust image encryption framework that ensures confidentiality and integrity of grayscale images during transmission. The model demonstrates improved entropy, low correlation between image pixels, and visual indistinguishability between the plain and encrypted images. This contributes to enhanced privacy and secure digital communication systems. By implementing public-key encryption (ECDH) with chaotic transformation and SHA-256 hashing, the work aligns with modern cryptographic practices and offers a potential real-world application in fields such as cybersecurity, medical informatics, and defense.

Limitations or Drawbacks of the Mini-Project Work

Despite its strengths, the mini-project has a few limitations:

- Currently supports only grayscale images; extension to RGB images is not implemented.
- Relies on 2D chaotic maps only, limiting the complexity of pixel scrambling compared to higher-dimensional maps.
- Real-time performance on large image datasets has not been thoroughly benchmarked.
- The system's integration into mobile or embedded systems remains unexplored due to current platform constraints.

Chapter-8

Conclusions and Future Work

This mini-project successfully explored and implemented a secure and efficient image encryption scheme using a combination of Elliptic Curve Cryptography (ECC), Diffie-Hellman key exchange (ECDH), and multidimensional chaotic maps such as the Arnold Cat Map. The work was aimed at enhancing data confidentiality in digital image transmission while ensuring robustness against cryptographic attacks.

The primary objectives—developing a secure encryption technique, minimizing computational complexity, and analyzing resistance against various types of attacks—have been largely achieved. The system was tested using key evaluation metrics like entropy, correlation coefficients, PSNR, SSIM, MSE, and UACI. Results indicated improved image quality, higher randomness in encrypted images, and resilience against brute-force and statistical attacks.

Key Achievements & Highlights

- Successfully integrated ECC and chaotic maps for image encryption.
- Utilized Spyder IDE and Anaconda environments effectively for implementation.
- Achieved strong encryption results with reduced computational cost.
- Demonstrated system resilience against differential and statistical attacks.
- Developed a lightweight yet secure encryption solution that can be adapted for real-world scenarios such as telemedicine, secure cloud storage, and wireless communication.

Innovative Aspects

- Novel combination of cryptographic techniques and chaotic systems.
- Use of multidimensional Arnold Cat Maps added complexity and unpredictability.
- Lightweight framework suitable even for resource-constrained devices.

Strong Points

- High security with less resource usage.
- Robust against known cryptographic and statistical attacks.
- Platform-independent software implementation using Python.

Limitations

- Current implementation depends on AES, which may not scale optimally for larger datasets.
- Arnold Cat Map parameters could be further optimized for higher unpredictability.
- No hardware-based encryption acceleration was used.

Scope for Future Work

To further enhance the system's performance and security, the following directions are recommended:

- Upgrade Encryption Technique from AES to ECC to fully utilize asymmetric encryption benefits, reduce key management complexity, and increase scalability.
- Improve Arnold Cat Maps by increasing dimensional complexity or combining with other chaotic systems to boost confusion and diffusion.
- Real-time Hardware Implementation on microcontrollers or IoT devices for practical deployment.
- Integration with Blockchain for image ownership tracking and secure sharing.
- GUI Development for non-technical users to apply encryption/decryption easily.

References

[1] P. Parida, C.Pradhan, D.S Roy, R.K Barik, X.Z Gao,"Image Encryption and Authentication With Elliptic Curve Cryptography and Multidimensional Chaotic Maps", Digital Object Identifier 10.1109, April 2021.

[2] Dr.P Ramadevi ,Dinesh Prabhu A ,Donisha K ,Baranika S."Implementation of Elliptic Curve Diffie Hellman (ECDH) Algorithm for Secured Communication", International Journal of Novel Research and Development, May 2024.

[3] Stephen Aikins-Bekoe, James Ben Hayfron-Acquah, "Elliptic Curve Diffie-Hellman (ECDH) Analogy for Secured Wireless Sensor Networks", International Journal of Computer Applications, April 2020.

[4] Javaria Akbar, Nasir Siddiqui, Shamsa Kanwal, Saba Inam," A Secure Transmission of Digital Images using Multiple Chaotic Maps and Elliptic Curve", International Journal of Research Publication and Reviews, June 2024.

[5] Mohammed Es-Sabry, Nabilel Akkad, Khrissi Lahbib, Khalid Satori, " An Efficient 32-bit color image encryption technique using multiple chaotic maps and advanced ciphers", Egyptian Informatics Journal, February 2024.

[6] Sanjay Kumar, Deemala Sharma, "A chaotic based image encryption scheme using elliptic curve cryptography and genetic algorithm", Artificial Intelligence Review, January 2024

[7] Hiba Hilal Hadi, Ammar Ali Neamah. "Diffie-Hellman Key Exchange Based on Block Matrices Combined with Elliptic Curves", International Journal of Intelligent Systems and Applications in Engineering, April 2023.

[8] Vinayak Tanksale, "Efficient Elliptic Curve Diffie–Hellman Key Exchange for Resource-Constrained IoT Devices", Ball State University Muncie, September 2024.

[9] Jun Xu, Santanu Sarkar, Huaxiong Wang, Lei Hu,"Improving Bounds on Elliptic Curve Hidden Number Problem for ECDH Key Exchange", Springer, September 2022.

[10] Adel. A Ahmed Abdullah, Omar M. Barukab, "Unforgeable Digital Signature Integrated into Lightweight Encryption Based on Effective ECDH for Cybersecurity Mechanism in Internet of Things", Taiz University, December 2022.

[11] V.Padmanabha Reddy, R. Murali Prasad, Pamula Udayaraju, Bhattu Hari Prasad Naik, Ch. Raja, "Efficient medical image security and transmission using modified LZW compression and ECDH-AES for telemedicine applications" , Soft Computing, May 2023.

[12] I. Khalid, T. Shah, S. M. Eldin, D. Shah, M. Asif and I. Saddique, "An Integrated Image Encryption Scheme Based on Elliptic Curve," in IEEE Access, vol. 11, pp. 5483-5501, 2023

[13] Javaria Akbar, Nasir Siddiqui, Shamsa Kanwal, Saba Inam, " A Secure Transmission of Digital Images using Multiple Chaotic Maps and Elliptic Curve", International Journal of Research Publication and Reviews, June 2024.

[14] Dani Elias Mfungo, Xianping Fu, Yongjin Xian, Xingyuan Wang, "A Novel Image Encryption Scheme Using Chaotic Maps and Fuzzy Numbers for Secure Transmission of Information", Applied Sciences MPDI, June 2023.