Kwaghe International
Journal of Sciences and Technology

ISSN : 0000-0000
Index: Harvard, Boston, Sydney University, Dimensions, Lens, ResearchGet Scilit, Semantic, Google Scholar, Base etc

https://doi.org/10.58578/KIJST.v1i2.3859

Enhancing Medical Image Security through Dual Cryptographic Paradigms: AES Symmetric Encryption and ECC Asymmetric Key Cryptography

Hambali Moshood A[1], Olasupo Yunusa A[2], Umar Shehu[3]
Federal University Wukari, Nigeria
hambali@fuwukari.edu.ng

Article Info:

Submitted:	Revised:	Accepted:	Published:
Sep 1, 2024	Sep 10, 2024	Sep 18, 2024	Sep 21, 2024

Abstract

The rapid advancements in medical imaging technologies have highlighted the urgent need for secure transmission methods to protect patient confidentiality and ensure data integrity. This study presents a hybrid encryption approach that integrates Elliptic Curve Cryptography (ECC) with the Advanced Encryption Standard (AES) to effectively encrypt and decrypt medical images. The research methodology includes dataset collection, the development of AES and ECC algorithms using the Tkinter GUI, and performance assessments. AES utilizes a 128-bit key length, allowing for quick encryption and decryption, while ECC enhances security through the use of a public-private key pair. The performance evaluation focuses on throughput in relation to image size and the time taken for encryption and decryption. This research work enhances data security in healthcare by providing a reliable and efficient model for the encryption and decryption of medical images.

Keyword: Cryptography, Cryptography Key, Decryption, Encryptions, ECC, Encoding, AES

INTRODUCTION

Recent advancements in medical imaging technologies, such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT), X-rays, and ultrasound, have significantly improved the diagnosis of various medical conditions (Magdy et al., 2022). However, the transmission of these medical images over networks has raised serious security concerns, particularly in telemedicine applications like telesurgery and teleconsultation, where secure sharing of images between patients, doctors, and diagnostic centers is essential. The secure transmission and safeguarding of medical images are vital in the context of disease diagnosis, given that these images frequently contain sensitive information. It is crucial to maintain the confidentiality and integrity of medical data during transmission, as unauthorized access or tampering can have serious repercussions for patient care (Qadir & Varol, 2019; Kamal et al., 2021; Gbolagade et al., 2022). Consequently, there is an urgent requirement for effective and robust security measures to protect medical images from potential breaches and unauthorized access. To protect patient privacy and ensure data integrity, robust security measures must be implemented. This study focuses on enhancing the security of medical images through cryptographic techniques, specifically by employing a hybrid approach that combines the Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC). This dual strategy aims to safeguard sensitive medical data from unauthorized access and tampering, thereby ensuring confidentiality and integrity during transmission.

Review of Related works

Cryptography, as defined by Mathur et al. (2023), involves creating written communications with the primary goal of preserving secrecy. Its origins can be traced back to ancient civilizations like Rome and Egypt, where hieroglyphic writing was used (Dewangan, 2020). Cryptography involves two fundamental operations: encryption and decryption. Encryption techniques are used to protect messages or data from unauthorized access within a network. Encryption is the process of obfuscating or hiding sensitive data, while decryption involves revealing or deciphering concealed or previously encoded information (Mallouli et al., 2019). Essentially, decryption converts encrypted data back into its original, readable form. Cryptography can be approached from various perspectives, and it includes three main types of algorithms: symmetric cryptography, asymmetric cryptography, and hash encryption.

Al-Haj et al. (2015) developed two cryptographic algorithms for encrypting DICOM images, utilizing robust functions like hash codes and symmetric keys. They implemented the Whirlpool hash function and Advanced Encryption Standard-Galois counter mode to ensure confidentiality and authenticity. However, these algorithms are computationally inefficient, taking approximately 811 and 484 seconds to encrypt.

Chen and Hu (2017) proposed an adaptive encryption algorithm for medical images, using an upgraded chaotic mapping technique with sine chaos and a hyper-chaotic system. They employed the Logistic-sine chaos mapping for initial spatial arrangement perturbation, followed by an adaptive encryption technique for enhanced diffusion. Their algorithm demonstrated commendable encryption performance, user-friendly operation, substantial key space, robust security, and resilience against attacks.

Laiphrakpam and Khumanthem (2017) introduced a novel encryption algorithm for medical images based on an advanced ElGamal algorithm. They eliminated the step of encoding images into elliptic curve coordinates, resulting in a stronger cipher image and significantly reduced execution time.

Cao et al. (2017) introduced an encryption algorithm using edge maps derived from the original image. The process involves bit-plane decomposition, random sequence generation, and permutation. Their cryptosystem is adaptable to different image types, bit-plane decomposition methods, and permutation algorithms, with system keys generated from the plain image, edge detector, and scrambling algorithm parameters. This approach enhances resistance to brute-force attacks.

Hua et al. (2018) proposed a method for manipulating pixel arrangements in medical images. It begins with introducing random noise to the image, followed by two iterations of pixel scrambling for diffusion. This process rearranges adjacent pixels and spreads the noise across the image. Key operations include XOR and modulo arithmetic, improving both security and encryption speed.

Nematzadeh et al. (2018) developed a hybrid encryption technique combining Genetic Algorithms with connected map lattices. The method generates a population of encrypted images, using a genetic algorithm to select optimal ciphers based on a fitness function for minimal loss and computational efficiency. This hybrid system aims to make cipher images resistant to conventional attacks.

Several studies have focused on optimizing AES for better performance in medical imaging contexts. For instance, Taha et al. (2019) proposed an optimized AES encryption method to enhance the speed and security of medical image encryption. While AES is secure, its computational intensity can be a drawback for real-time applications.

Li et al. (2020) introduced a novel chaos-based encryption algorithm that uses a combination of chaotic maps for enhanced security. The method showed improved resistance against cryptographic attacks. The complexity of chaos-based systems can lead to implementation challenges and potential performance issues.

Ahmad et al. (2021) developed a hybrid encryption scheme that integrates AES with chaos-based encryption, achieving a balance between security and performance. This approach demonstrated superior resilience against common attacks. Hybrid methods can be complex to implement and may require more computational resources.

Zhang et al. (2022) explored the feasibility of quantum encryption for medical images, showing promising results in terms of security. However, practical implementation remains in the early stages. Quantum cryptography is still in the experimental phase and faces significant technical and logistical challenges before widespread adoption.

El-Latif et al. (2023) proposed an image-specific encryption algorithm that takes into account the high redundancy and correlation in medical images, achieving efficient and secure encryption. Such tailored approaches may lack generalizability across different types of medical images and applications.

Key findings indicate that symmetric encryption, such as the Advanced Encryption Standard (AES), offers high-speed processing and efficiency, while asymmetric encryption, like Elliptic Curve Cryptography (ECC), provides enhanced security through better key management. The proposed hybrid approach, which combines these two paradigms, aims to leverage their respective strengths to address the unique challenges associated with healthcare data transmission and storage (Qadir & Varol, 2019; Kamal et al., 2021).

There is a notable deficiency in research on hybrid cryptographic methods that integrate symmetric (AES) and asymmetric (ECC) encryption techniques for safeguarding medical data. While the individual advantages of these algorithms are well-documented, their combined use in healthcare settings remains underexplored. Furthermore, existing studies do not sufficiently address the challenges of securely transmitting and storing medical images in the face of evolving cybersecurity threats. This research gap highlights the need

for developing integrated cryptographic solutions to enhance the confidentiality, integrity, and accessibility of sensitive medical information. The study aims to provide insights into effectively implementing such hybrid systems, ultimately strengthening the security framework for medical data management.

METHODS

The following outlines the research methodology for encrypting and decrypting information transmitted over the cloud, including descriptions of the cryptographic algorithms used, their pseudocode, and the tools for implementation.

Dataset Collection: Medical datasets are procured from the Kaggle repository and integrated into the program file.

Cryptography Model Implementation: The ECC public key was used to encrypt the AES secret key before sending it to the receiver on the cloud. While the ECC private key serves as the encryption key for the AES secret key sent to the receiver. The encrypted image is now decrypted with the AES secret key. Figure 1 provides a visual summary of the implementation of the research methodology.

Performance Evaluation: This phase involves assessing encryption and decryption times, and evaluating throughput based on image size and processing durations to determine the system's efficiency and effectiveness.

Dataset Description

This study uses the Chest X-ray images (Pneumonia) dataset from Kaggle (https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia). It includes three classes: normal (clear lungs), bacterial pneumonia (focal lobar consolidation, often in the right upper lobe), and viral pneumonia (diffuse interstitial pattern affecting both lungs).

The dataset comprises 5,863 JPEG-format chest X-ray images, categorized into Pneumonia and Normal, organized into train, test, and validation folders. The images, sourced from pediatric patients aged one to five years at Guangzhou Women and Children's Medical Center, were part of routine clinical care. Quality control screening removed low-quality scans, and diagnoses were graded by two expert physicians, with a third expert reviewing the evaluation set to ensure accuracy.

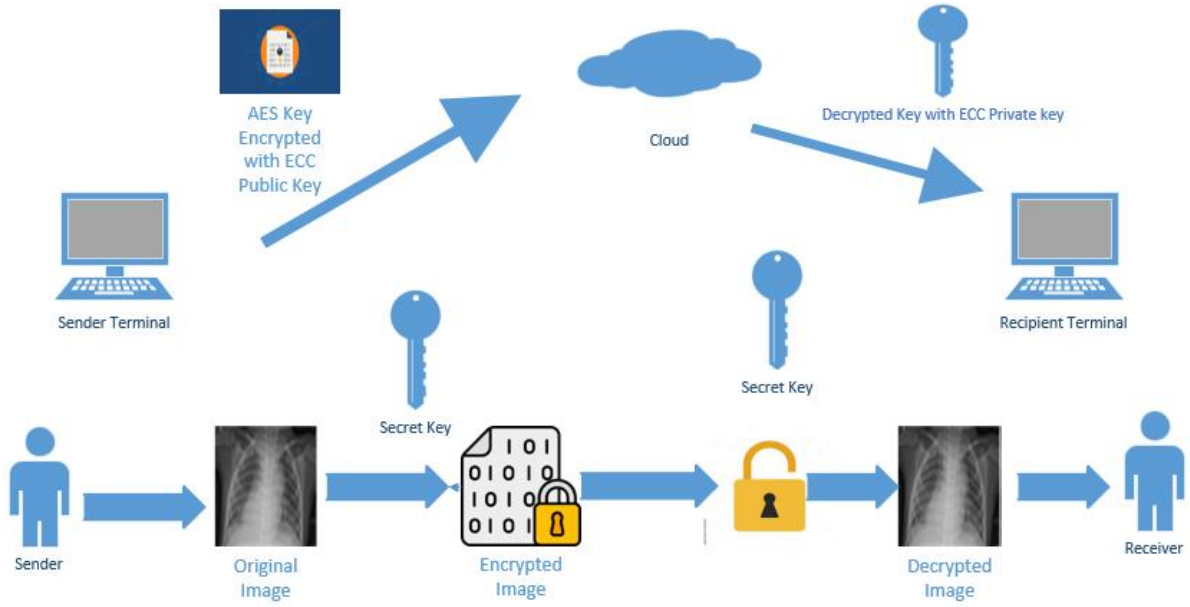


Figure 1**:** Proposed Framework

Analysis of the Proposed Cryptographic Method

This section presents details of the approaches used for the implementation of encryption and decryption of medical images. The system works by encrypting the input data using a two-layer approach involving ECC and AES cryptography. This encrypted information is then transmitted over the cloud to the intended recipient. At the receiver's end, the platform reverses the encryption process to decrypt the information and make it accessible to the user.

Advanced Encryption Standard (AES)

AES is a block cipher with a block size of 128 bits and key lengths of 128, 192, or 256 bits. In this study, AES with a 128-bit key was used. The encryption process involves 10 rounds of processing for 128-bit keys. Each round, except the final one, follows the same sequence of operations. The 16-byte key is expanded into a schedule of 44 4-byte words, and the 128-bit input block is arranged as a 4x4 matrix of bytes, known as the state array.

Before encryption begins, the input state is combined with the first four key schedule words using XOR. Each encryption round includes four steps:

i. **Sub Bytes:** This step performs non-linear byte substitution using the S-box lookup table. Each byte in the state is replaced based on its position in the S-box, which is constructed using Galois Field arithmetic and bitwise operations.

This substitution adds crucial non-linearity to the encryption, enhancing its resistance to cryptanalysis.

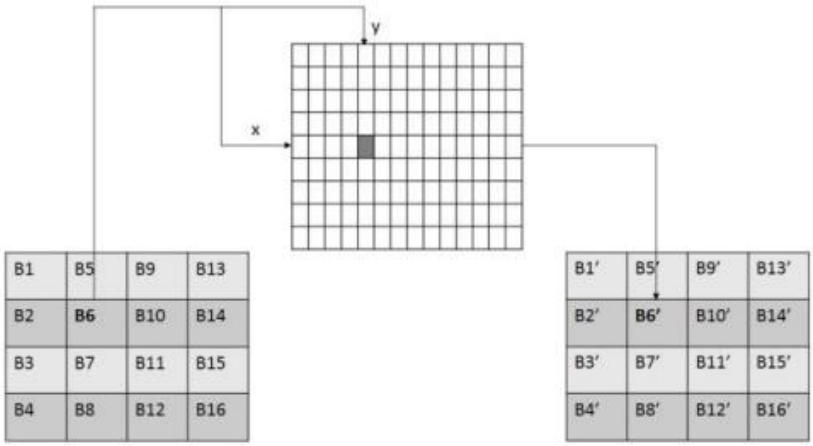


Figure 2: SubByte Transformation. Source: (Jakobsen and Knudsen, 1997)

ii. Shift Rows: This step cyclically shifts the bytes in each row of the state matrix. Each row is shifted a different number of bytes to disperse information across the state, enhancing encryption. For instance, row 0 remains unchanged, row 1 shifts left by one byte, row 2 shifts left by two bytes, and row 3 shifts left by three bytes. This process ensures systematic transformation and improved security.

$$\begin{bmatrix} B1 & B5 & B9 & B13 \\ B2 & B6 & B10 & B14 \\ B3 & B7 & B11 & B15 \\ B4 & B8 & B12 & B16 \end{bmatrix} \qquad \begin{bmatrix} B1 & B5 & B9 & B13 \\ B6 & B10 & B14 & B2 \\ B11 & B15 & B3 & B7 \\ B16 & B4 & B18 & B12 \end{bmatrix}$$

iii. **Mix Columns**: This step mixes the bytes within each column of the state matrix using predefined matrix multiplication. Each byte is multiplied by values from a standard matrix, and the results are XORed to enhance confusion and diffusion. For example, each byte in a column is multiplied by specific matrix values and then XORed to produce new byte values i.e. the first byte denoted as B1", is multiplied by 02, 03, 01, and 01, respectively, and then XORed to generate the new value, B1'", within the resulting matrix.. This operation increases encryption security by thoroughly mixing the column data.

$$\begin{bmatrix} 01 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} B1 & B5 & B9 & B13 \\ B6 & B10 & B14 & B2 \\ B11 & B15 & B3 & B7 \\ B16 & B4 & B18 & B12 \end{bmatrix} \begin{bmatrix} B1 & B5 & B9 & B13 \\ B6 & B10 & B14 & B2 \\ B11 & B15 & B3 & B7 \\ B16 & B4 & B18 & B12 \end{bmatrix}$$

iv. **Add Round Key**: In this step, each byte of the state is XORed with the corresponding byte of the round key, derived from the original cipher key through key scheduling. This ensures continuous alteration of the state and strengthens security against unauthorized decryption. Algorithm 1 depicts the process of how the system implements the AES algorithm and Figure 3 depicts AES architecture.

Algorithm 1: AES

```
function AESencrypt(plaintext, key)
blocks := divideIntoBlocks(plaintext)
roundKeys := getRoundKeys(key)
for block in blocks:
state := block
addRoundKey(state, roundKeys[0])
for round := 1 to Nr - 1:
substituteBytes(state)
shiftRows(state)
mixColumns(state)
addRoundKey(state, roundKeys[round])
substituteBytes(state)
shiftRows(state)
addRoundKey(state, roundKeys[Nr])
return combineBlocks(state)
```

Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) is an asymmetric encryption technique gaining popularity as an alternative to RSA. ECC uses elliptic curve theory to generate smaller, faster, and more efficient keys. Breaking ECC requires solving the elliptic curve discrete logarithm problem, which is much harder than factoring. Consequently, ECC can provide comparable security to RSA with smaller key sizes, reduced computing power, and lower battery usage. The elliptic curve is defined by a Weierstrass equation:

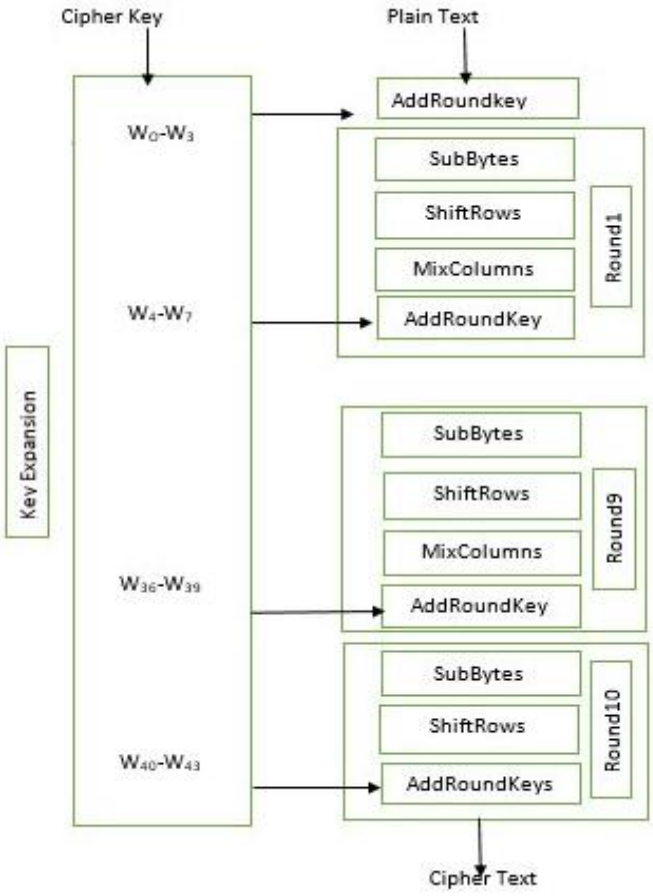


Figure 3: Advanced Encryption Standard (AES) (source: Ghosh et al., 2019)

$$Y^2 = X^2 + ax + b(\text{mob } p) \qquad 1$$

Here, p is a prime number p ≠2 and 3, and a and b are the constants concerning the curve satisfying $4a^3 + 27b^2 \neg\equiv 0(\text{mob } p)$. The $(\text{mob } p)$ indicates that the algorithm will be dealing with the curve in a finite field. The curve also consists of the point at infinity O. The curve also has a point G known as the curve generator whose point multiplication can generate all points of the field F. In summation, the curve can be written as $E(F_p) = \{a, b, p, G\}$. The properties of the Elliptic Curve include point addition, doubling, and multiplication.

Point Addition: assuming, there exist 2 points say $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ where P_1 is not equal to P_2. Adding these 2 points result to $R := (x_3, y_3)$ which relies on the same curve where:

$$S = \frac{y_2 - y_1}{x_2 - x_1} (\text{mod } p) \qquad 2$$

$$x_3 = s^2 - (x_2 + x_1) (\text{mod } p) \qquad 3$$

$$y_3 = s(x_1 - x_3) - y_1 (\text{mod } p) \qquad 4$$

Point Doubling: When $P_1 = P_2$, the condition is known as Point Doubling. Hence, R can be defined as

$$S = \frac{3x_1^2 + a}{2y_1} (\text{mod } p) \qquad 5$$

The formula for x and y coordinates remains the same as the point addition.

Point Multiplication: The point multiplication (performing k times P) is defined by the repetitive addition of point P with itself for k times. The algorithm computes the point multiplication in log (k) time complexity, efficiently allowing computation for large values of k.

$$R = kP = P + P + P + \cdots + P \text{ (k times)} \qquad 6$$

An effective algorithm to solve point multiplication can be shown as an example:

$$R = 15P = 2(2(2P + P) + P) + P \qquad 7$$

Algorithm 2 depicts the process of how the system implements the ECC algorithm.

Performance Measures

The performance of the encryption system is evaluated based on the following measures:

i. **Time:** the duration taken to encrypt and decrypt some information.
ii. **File-Size:** the size of the file utilized to encrypt and decrypt using the two cryptographic algorithms.
iii. **Throughput:** is a measure of the file's sizes per unit execution time of the algorithms for both encryption and decryption.

Algorithm 2: ECC

```
function PointAdd(P, Q, a, p)
    if P.x == Q.x and P.y == -Q.y:
    return O
     elif P.x == Q.x:
     m = ((3 * P.x^2 + a) * modInv(2 * P.y, p)) % p
      else:
       m = ((Q.y - P.y) * modInv(Q.x - P.x, p)) % p
       Rx = (m^2 - P.x - Q.x) % p
       Ry = (m * (P.x - Rx) - P.y) % p
       return (Rx, Ry)
```

Experimental Setup

The image cryptography system was implemented in a Python studio environment with Tkinter standard GUI toolkit. Combining Tkinter and Python provides a practical approach to integrating cryptographic algorithms with a graphical interface, ensuring a user-friendly experience while maintaining the security and confidentiality of the data. This synergy makes it an effective choice for researchers and developers in image cryptography. Table 1 shows details of the system configuration used for the experiments.

Table 1: Specifications Definition

Utilities	**Values/Size**
Operating System (OS)	Window 10
Programming Language	Python
Processor	Intel Core i7
Processor speed	2.7GHz
Disk	Hard Disk Drive (HDD)
Programming environment	Visual Studio Code

Parameter Settings

The module parameter settings define the variables for the hybrid cryptographic system, with parameters for AES and ECC algorithms detailed in Table 2.

- **Nonce:** A random or pseudo-random number used to secure and authenticate communication protocols.

- **Secret Key:** A randomized stream of characters used for encrypting plaintext and decrypting encrypted data.

For handling incomplete data blocks, PKCS7 padding is used. PKCS7 pads block with a consistent number of bytes to ensure full blocks and remove padding during decryption.

ECC Private Key Generation: Utilizes the "SECP256R1" elliptic curve, which is a 256-bit curve recommended by SECG (Standards for Efficient Cryptography Group) for generating private key characters.

Table 2: parameter setting

Parameters	Value
Nonce	32 bytes string
Secret key	16 bytes string
Pad	PKCS7-128-bit block
Un-Pad	PKCS7-128-bit block
Private-Key	SPECP256R1

RESULTS AND DISCUSSION

The hybrid cryptographic system architecture combines cryptographic algorithms to enhance security, as described in the methodology section. This architecture operates in layers:

- **Image Encryption:** The AES algorithm encrypts the image data.
- **Key Encryption:** The ECC algorithm encrypts the AES secret key.

In this model, the sender first encrypts the image using AES and then encrypts the AES key with ECC. The sender transmits both the encrypted image and the encrypted AES key to the recipient.

At the recipient's end, the process is reversed:

- The recipient uses ECC's public key to decrypt the AES key.
- The decrypted AES key is then used to decrypt the image.

This layered approach ensures that the AES key remains confidential and secure throughout the process. Figures 4 to 5 illustrate the encryption and embedding procedures applied to medical images sourced from the medical dataset. Specifically, Figure 4 depicts the control menu, which enables navigation to either the key generation using the ECC, the

Encryption via the encrypt button from Figure 4, and the decryption via the decrypt button.

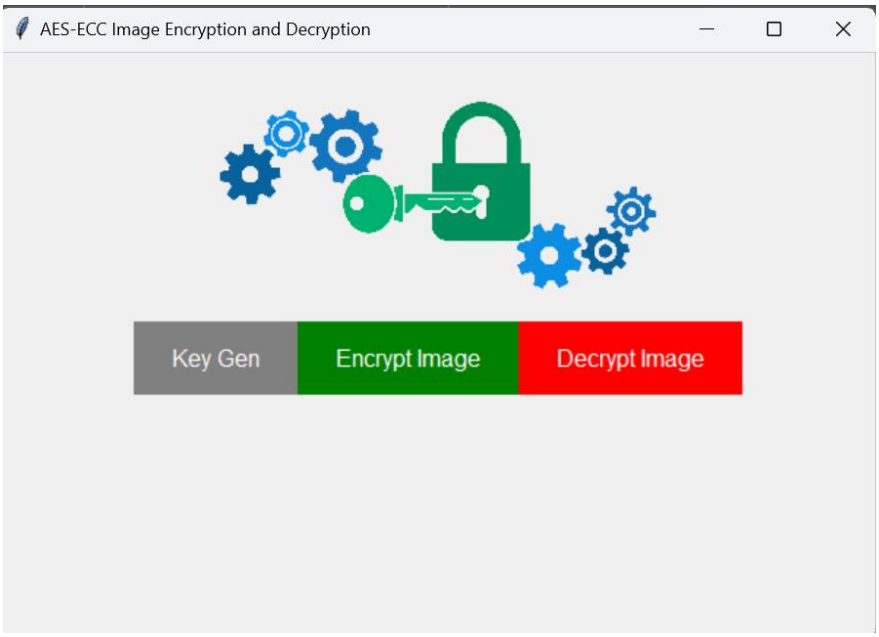


Figure 4: Control Menu

In the context of key generation from Figure 5, the interface incorporates input fields for user interaction, wherein the key is entered, and subsequently, the "generate key" button is activated. Upon this action, the system initiates the generation process, resulting in the production of essential cryptographic components: the public key, private key, and the cypher nonce associated with the key. These components play pivotal roles in securing communication channels and safeguarding sensitive information in cryptographic protocols.

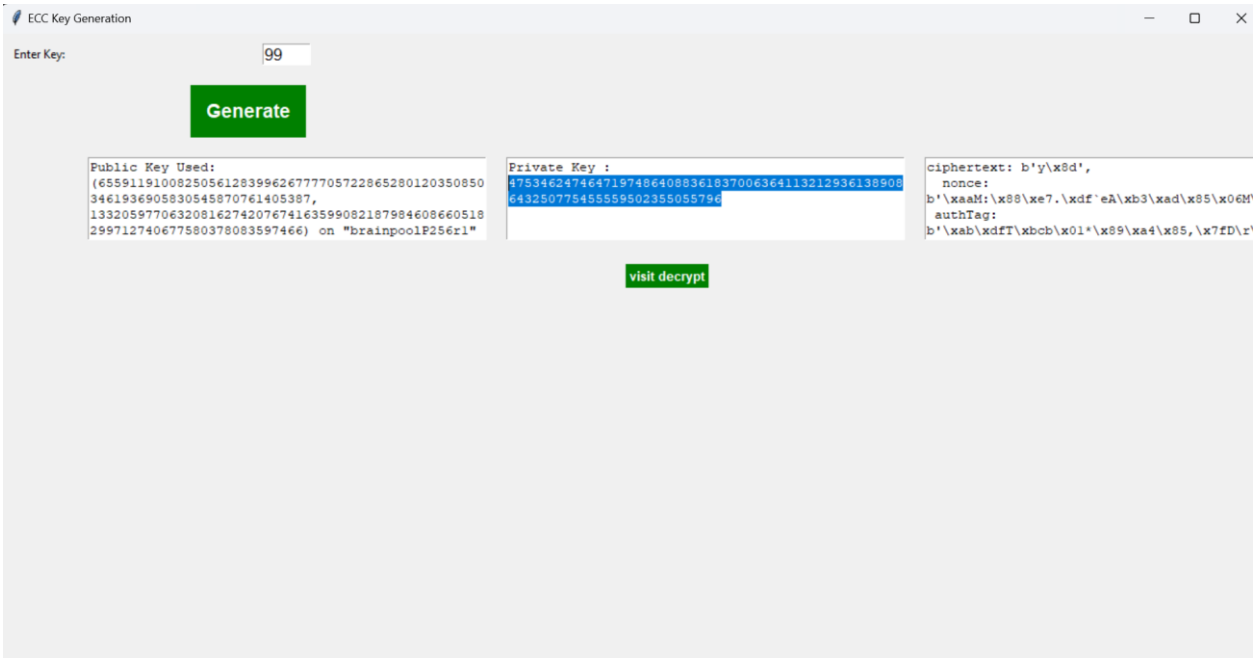


Figure 5: Key Generation Panel

To initiate the encryption process after key generation, the encryption interface, as depicted in Figure 6, offers input fields facilitating the insertion of the private key and browsing of the images. Following completion of the requisite fields and activation of the encrypt button, the interface facilitates the display of both the encrypted image and its original counterpart.

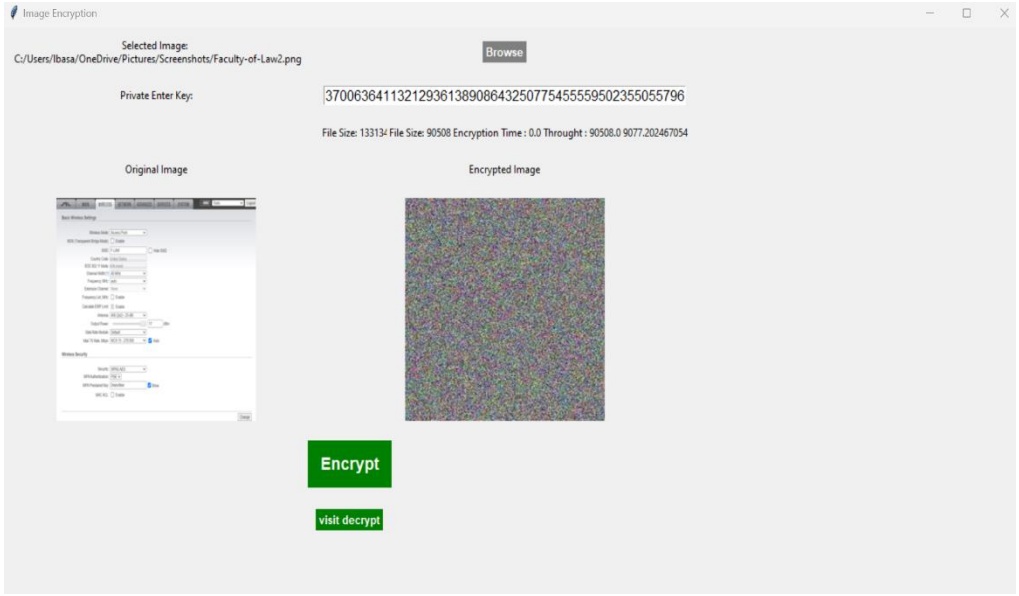


Figure 6: Image Encryption Panel

The decryption process, as depicted in Figure 7, facilitates the provision of fields for browsing the encrypted image and inputting the decryption key. Subsequently, upon activation of the submit button, the decrypted image is generated and displayed alongside the encrypted image for comparative analysis. This method enables a comprehensive examination of the encryption and decryption processes, facilitating a deeper understanding of the algorithm's effectiveness in preserving data integrity and confidentiality.

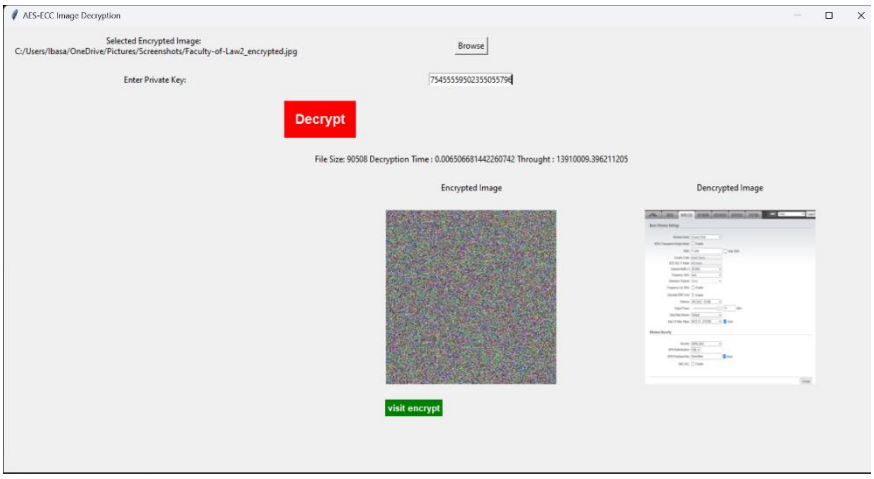
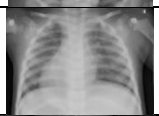


Figure 7: Decryption Panel

Table 3: Experimental Results

S/N	IMAGES	FILE SIZE	ENCRYPTION TIME	ENCRYPTIONTHROUPUT	DECRYPTION TIME	DECRYPTION THROUGHPUT
1		25452	0.0080	3,181,500.00	0.0156	1,631,538.46
2		81811	0.0159	5,145,345.91	**0.0001**	818,110,000
3		132725	0.0130	10,209,615.39	0.0156	8,508,012.82
4		139080	0.0119	11,687,394.96	0.0156	8,915,384.62
5		10867	0.0128	848,984.38	**0.0001**	108,670,000

6		74804	0.0133	5,624,360.90	**0.0001**	748,040,000
7		183331	0.0127	14,435,511.81	0.0156	11,751,987.18
8		168188	0.0132	12,741,515.15	0.0156	10,781,282.05
9		48517	**0.0054**	8,984,629.63	**0.0001**	485,170,000
10		127764	0.0142	8,997,464.79	0.0625	2,044,224
11		55124	0.0112	4,921,785.71	**0.0001**	551,240,000
12		51460	0.0151	3,407,947.02	0.0156	3,298,717.95
13		51460	0.0143	3,598,601.40	**0.0001**	514,600,000
14		84306	0.0141	5,979,148.94	0.0156	5,404,230.76
15		32705	0.0112	2,920,098.29	**0.0001**	327,050,000

Table 3 presents the results of AES-ECC medical image encryption and decryption processes across multiple image files, detailing file size, encryption time, encryption throughput, decryption time, and decryption throughput. Analyzing the data reveals several noteworthy observations and implications.

Encryption and Decryption Performance

Encryption Time: Encryption times vary from 0.005 to 0.0159 seconds across different image files, influenced by image complexity and size. Despite these variations, the AES-ECC algorithm maintains efficient encryption performance for all images.

Encryption Throughput: Throughput ranges from millions to hundreds of million bytes per second. Smaller images typically show higher throughput due to reduced computational

demands, while larger images have lower throughput. Overall, throughput remains high, demonstrating the algorithm's efficiency in processing medical image data.

Decryption Time: Decryption times range from 0.0001 to around 0.063 seconds, with variability due to image complexity and size. Despite this, decryption times are generally low, highlighting the effectiveness of the AES-ECC algorithm in recovering medical image data.

Decryption Throughput: Similar to encryption throughput, decryption throughput varies from millions to hundreds of million bytes per second. The algorithm consistently achieves high throughput, ensuring rapid data recovery while preserving data integrity.

The AES-ECC algorithm demonstrates robust efficiency and reliability in encrypting and decrypting medical image data, with high performance across various metrics.

CONCLUSION

The growing reliance on digital systems and data transmission across networks underscores the need for robust security protocols. Cryptography plays a crucial role in protecting data integrity, confidentiality, and authenticity. This study examines cryptography's role in enhancing system and file protection, focusing on the medical sector's cybersecurity needs. Using Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC), the research aims to secure medical scan images. A three-phase approach—data collection, cryptographic system development, and performance evaluation—was employed. The study utilized Python and Tkinter for system development and evaluation, revealing cryptography's effectiveness in protecting against unauthorized data access and manipulation. The findings highlight cryptography's essential role in safeguarding sensitive medical data. The AES-ECC hybrid cryptographic system has proven effective in securing medical image data. The performance metrics such as encryption time, decryption time, file size, and throughput demonstrate the system's robustness and scalability. The developed GUI enhances user experience by facilitating seamless encryption and decryption. The future research direction will focus on optimizing the AES-ECC system for better performance and scalability.

REFERENCES

Al-Haj, A., Abandah, G., and Hussein, N. (2015). Crypto-based algorithms for secured medical image transmission. *IET Inf. Secur*, *9*, 365–373.

Cao, W., Zhou, Y., Chen, C.P., and Xia, L. (2017). Medical image encryption using edge maps. *Signal Process*, *132*, 96–109

Chen, X., and Hu, C.J. (2017). Adaptive medical image encryption algorithm based on multiple chaotic mapping. *Saudi J. Biol. Sci.*, *24*, 1821–1827.

Dewangan, P. (2020). A review paper on network security and cryptography. *International Journal of Science and Research (IJSR)*, *9*(1).

Gbolagade Morufat D., Moshood A. Hambali, Oluwadamilare H. Abdulganiyu, E. Lawrence (2022). Enhance Facial Biometric Template Security using Advance Encryption Standard with Least Significant Bit, Journal of Computer Science and Engineering (JCSE), 3(2), 60-70.

Ghosh, P., Hasan, M. Z., Atik, S. T., & Jabiullah, M. I. (2019, December). A variable length key based cryptographic approach on cloud data. In *2019 International Conference on Information Technology (ICIT)* (pp. 285-290). IEEE.

Hua, Z., Yi, S.; and Zhou, Y. (2018). Medical image encryption using high-speed scrambling and pixel adaptive diffusion. *Signal Process*, *144*, 134–144.

Kamal, S. T., Hosny, K. M., Elgindy, T. M., Darwish, M. M., & Fouda, M. M. (2021). A New Image Encryption Algorithm for Grey and Color Medical Images. *IEEE Access,* 9, 37855- 37865

Laiphrakpam, D.S.; Khumanthem, M.S. (2017). Medical image encryption based on improved ElGamal encryption technique. *Optik*, *147*, 88–102.

Magdy, M., Hosny, K. M., Ghali, N. I., & Ghoniemy, S. (2022). Security of medical images for telemedicine: a systematic review. *Multimedia Tools and Applications*, *81*(18), 25101-25145.

Mallouli, F., Hellal, A., Saeed, N. S., & Alzahrani, F. A. (2019). A survey on cryptography: comparative study between RSA vs ECC algorithms, and RSA vs El-Gamal algorithms. *IEEE,* 173-176.

Mathur, P., Saxena, S., & Mishra, Y. (2023). The Evolution and Impact of Cryptography in Ensuring Data Privacy. *International Journal of Scientific Research & Engineering Trends,* 9(3), 709-711

Nematzadeh, H., Enayatifar, R., Motameni, H., Guimarães, F.G., and Coelho, V.N. (2018). Medical image encryption using a hybrid model of modified genetic algorithm and coupled map lattices. *Opt. Lasers Eng*, *110*, 24–32

Qadir, A. M., & Varol, N. (2019). A review paper on cryptography. IEEE, 1-6.