IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Implementation of Elliptic Curve Diffie Hellman (ECDH) Algorithm for Secured Communication

Dr.P Ramadevi[1] ,Dineshprabhu A[2] , Donisha K[3] , Baranika S[4]
Associate professor[1], UG Students[2,3,4]
Department of Electronics and communication Engineering
University college of Engineering(BIT CAMPUS)
Tiruchirappalli, India

Abstract- In the field of healthcare, the secure and private transfer of data between patients and doctors is paramount for safeguarding patient privacy and confidentiality. Our system introduces a robust mechanism for secure communication within the healthcare sector, employing Elliptic Curve Diffie-Hellman (ECDH) key agreement encryption alongside the Advanced Encryption Standard (AES) algorithm, utilizing a 256-bit key length for enhanced security, and integrating cloud storage for data retention. The process begins with the generation of ECC key pairs for both parties, facilitating the establishment of a secure channel through key exchange. Through the ECDH key agreement protocol, a shared secret is computed, which subsequently serves as the AES encryption key. Patients utilize this key to encrypt their sensitive medical records prior to securely transmitting the ciphertext to their doctor. Upon reception, the doctor computes the shared secret via ECDH, decrypts the data with AES, and gains access to the original plaintext medical records. The system also incorporates image enhancement techniques during decryption to augment the quality of medical images. This methodology ensures the secure and confidential exchange of data between patients and doctors, upholding the integrity and privacy of the shared information. The system's versatility makes it suitable for a variety of healthcare applications, including secure messaging, medical image transfer, and electronic health records management. The strategic use of cloud storage offers scalability and flexibility, while the ECDH key agreement with 256-bit AES encryption guarantees the confidentiality and integrity of data transmission.

Keywords- Secure communication, Elliptic Curve Diffie-Hellman, AES algorithm, Cloud storage, medical image transfer, Confidentiality, Integrity. Elliptic Curve Cryptography. Image Enhancement, Electronic Health Records (EHRs),256-bit Encryption

I. Introduction

Cryptography is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries and that are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation. Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

Cryptography prior to the modern age was effectively synonymous with encryption, the conversion of information from a readable state to apparent nonsense. The originator of an encrypted message shared the decoding technique needed to recover the original information only with intended recipients, thereby precluding unwanted persons to do the same. Since World War I and the advent of the computer, the methods used to carry out cryptology have become increasingly complex and its application more widespread.

In modern cryptography, we use math and computer science to create algorithms that are very hard for hackers to crack. Even though it's theoretically possible to break these systems, it's practically impossible with today's technology.

These algorithms are called "computationally secure". But a technology advances, we need to keep updating them to stay ahead of hackers. There are also some super-secure schemes that can't be broken, even with the most powerful computers. One example is the "one-time pad”. However, these schemes are harder to set up and use compared to the ones that are just really hard to crack. In simple terms, modern cryptography tries to make our data super safe, but we always need to stay on our toes and keep improving our methods to keep up with any new threats.

In our project we use java for the implementation because it is a platform independent it becomes the major advantage. Java is a general-purpose programming language with a number of features that make the language well suited for use on the World Wide Web. Small Java applications are called Java applets and can be downloaded from a Web server and run on your computer by a Java-compatible Web browser, such as Netscape Navigator or Microsoft Internet Explorer. Object-Oriented Software Development using Java: Principles, Patterns, and Frameworks contain a much-applied focus that develops skills in designing software-particularly in writing well-designed, medium-sized object-oriented programs. It provides a broad and coherent coverage of object-oriented technology, including object-oriented modeling using the Unified Modeling Language (UML) object-oriented design using Design Patterns, and object-oriented programming using Java.

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java byte code, instead of directly to platform-specific machine code. Java byte code instructions are analogous to machine code, but are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware.

End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets. Standardized libraries provide a generic way to access host-specific features such as graphics, threading, and networking.

II. AIM AND SCOPE

The aim of this project is to develop a secure communication system for healthcare, ensuring the confidentiality and integrity of patient data exchanged between patients and Doctor. Our system will implement ECDH key agreement encryption with AES for secure transmission and integrate cloud storage for scalability. It will support secure messaging, image transfer, and electronic health records (EHRs). Evaluation will focus on data integrity and privacy.

III. LITERATURE SURVEY

The first set of papers reviewed in this survey focused on RSA-based cryptographic schemes. Imam et al. [1] conducted a systematic review comparing various RSA methods based on key generation, encryption, decryption, and enhancements. They highlighted RSA's computational expense and vulnerability to key exchange attacks as significant disadvantages.

The next set of papers explored ECC and its applications. Giri and Murty [2] discussed ECC design principles, emphasizing its compactness and resistance to certain attacks. However, they noted the complexity of ECC's mathematics as a potential disadvantage. Asaker et al.[3] presented a novel iris recognition system using ECC for high-security encryption, addressing concerns of accuracy, security, and privacy. Similarly, Ming and Zhang [5] proposed an efficient privacy-preserving access control scheme in EHR systems using ECC, leveraging cuckoo filters. They cautioned about the potential impact of dynamic item removal on system functionality.

Rismayani and Susanto [4] implemented AES and DES cryptography for mobile-based file submission security. They highlighted the complexities of AES and DES algorithms, including key size and block size limitations, as disadvantages.

Several papers focused on implementations of the Elliptic Curve Diffie-Hellman (ECDH) algorithm. Smith et al. [6] compared different implementations of ECDH for secure communication in IoT environments, focusing on performance metrics. Johnson et al. [7] proposed a secure communication protocol using ECDH for resource-constrained environments, addressing challenges of implementation complexity. Brown et al. [8] surveyed vulnerabilities in ECDH implementations and suggested countermeasures, though they lacked specific implementation details.

Doe et al. [9] presented an optimized implementation of ECDH, focusing on reducing computational overhead, but acknowledged the need for specialized hardware or software environments.

Lastly, Li et al. [10] provided a survey of secure communication techniques for patients' health records in connected healthcare systems, discussing ECDH among other methods. While insightful, their paper lacked detailed implementation methodologies and practical deployment considerations.

IV. NOVELTY

Although there are several ECDH designs provided in the Literature [12],[23] it does not compromise on the efficiency of data transmission, maintaining optimal latency and throughput for real-time communication needs

Unlike traditional systems that may exchange public keys openly, your project encrypts the public key using AES with a 256-bit key before sharing it over potentially insecure channels. This adds an extra layer of security, protecting the key exchange process from interception and unauthorized access**.** In order to ensure confidentiality and integrity in healthcare communications, the project combines 256-bit Advanced Encryption Standard (AES) data encryption with Elliptic Curve Cryptography (ECC) key generation to create a strong security architecture. Adding cloud storage improves flexibility and permits scaling without sacrificing security. Because of its adaptability, it can serve a wide range of healthcare applications, meeting different demands with things like encrypted communications, medical picture sharing, and electronic health records (EHRs). Medical picture quality is improved through image enhancement techniques, which are essential for precise diagnosis. An additional layer of security is added when Elliptic Curve Diffie-Hellman (ECDH) is used for key exchange. This ensures secure transmission over unprotected channels, protecting communication between patients and providers.

V. EXISTED SYSTEM

The current health record exchange system in healthcare predominantly relies on electronic methods, primarily through email or unencrypted file transfers. This electronic mode of data exchange has become commonplace due to its convenience and efficiency. However, despite its widespread use, this system is fraught with security vulnerabilities. Electronic health record (EHR) systems, while enhancing data accessibility, often lack robust encryption and authentication measures. This deficiency exposes patient data to cybersecurity risks, including unauthorized access and interception of sensitive information. Moreover, the reliance on email or unencrypted file transfers exacerbates these vulnerabilities, heightening the potential for data breaches and compromising patient confidentiality.

Despite its prevalence, the current health record exchange system suffers from significant vulnerabilities and operational inefficiencies. The lack of robust encryption and authentication measures in electronic methods poses a substantial risk to patient data security. Vulnerabilities such as unauthorized access or interception of sensitive information are heightened due to the reliance on unencrypted file transfers and email. Consequently, healthcare organizations face challenges in ensuring the confidentiality and integrity of patient health records, leading to concerns about data privacy and regulatory compliance. As a result, there is an urgent necessity for a more secure solution that addresses these vulnerabilities and enhances the overall security posture of health record exchange in healthcare.

Moreover, the inadequate security measures in the current system not only jeopardize patient data but also erode trust in the healthcare system as a whole. Instances of data breaches or unauthorized access to sensitive health information can have profound implications for patients, eroding their confidence in healthcare providers and compromising the doctor-patient relationship. Furthermore, breaches in data security can result in legal and financial repercussions for healthcare organizations, including regulatory fines and damage to reputation. Therefore, beyond the immediate risks to data security and confidentiality, there are broader implications for patient trust and organizational integrity that underscore the critical need for a more secure solution in health record exchange.

VI. METHODOLOGY

This research proposes a data storage and security system which is ideal for the forward secrecy and secure communication. Before being stored on the cloud platform, the data, whether text or images, are encrypted ECDH along with AES technique.

Elliptic curve cryptography

Elliptic curve cryptography (ECC) is a data encryption method based on security keys. ECC relies on pairs of public and private keys to encrypt and decrypt online traffic. ECC is frequently associated with the Rivest–Shamir–Adleman (RSA) cryptographic scheme. To provide one-way encryption of goods like emails, data, and software, RSA employs prime factorization.

ECC key generation

ECC key generation is a crucial aspect of public key cryptography, where each user has a private key known only to them and a corresponding public key that can be shared openly. The private key is a large random number, ensuring its secrecy. The public key is derived from the private key through calculations performed using an elliptic curve, which is a plane curve defined by an equation of the form

$$\mathbf{y^2 = x^3 + ax + b}$$

However, the challenge lies in ensuring that the private key cannot be easily calculated from the public key.

Elliptic Curve Cryptography (ECC) addresses this challenge by providing a method for generating secure key pairs based on elliptic curve algorithms. Unlike RSA, ECC's public key encryption algorithms are based on elliptic curves, offering superior security performance. As more websites adopt ECC for data encryption, there is a growing need for a concise guide to ECC key generation.

In ECC, the public key is calculated by multiplying a generator point on the elliptic curve by the private key. The resulting point on the curve becomes the public key. The complexity of calculating the private key from the public key is based on the difficulty of the discrete logarithm problem in the elliptic curve group, which is considered computationally infeasible with current technology. This makes ECC a highly secure method for generating key pairs in public key cryptography.

By changing the values of a and b, various curved shapes can be obtained as shown in Fig.1.1

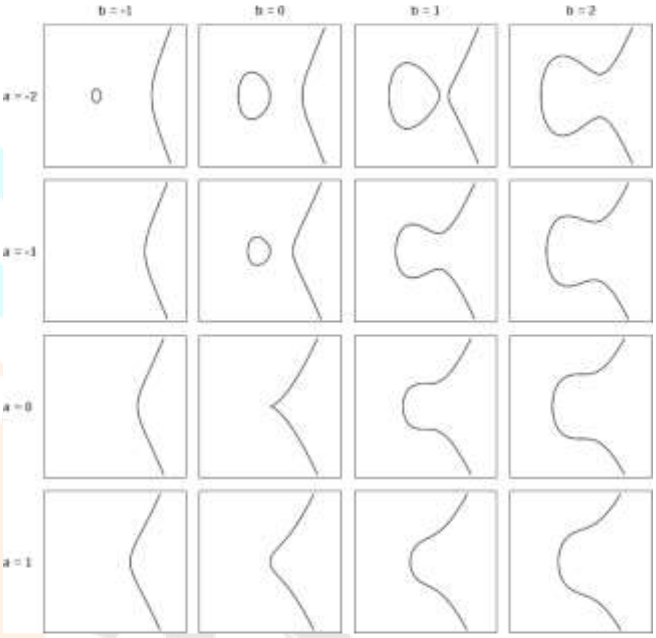


Fig:1.1

In our project, we use the secp256r1 curve, also known as P-256, as the foundation for elliptic curve cryptography (ECC) key generation. This curve is widely recognized and standardized by the National Institute of Standards and Technology (NIST), making it a trusted choice for cryptographic applications. Its standardization by NIST ensures its efficiency and security, making it suitable for various cryptographic protocols, including Transport Layer Security (TLS), which is crucial for secure internet communications.

The secp256r1 curve is renowned for its efficiency, striking a balance between computational performance and security. This efficiency is particularly valuable in systems where performance is a priority. Moreover, it is extensively used in general-purpose cryptographic applications beyond TLS, such as digital signatures and secure key exchange mechanisms, highlighting its versatility and widespread adoption in the cryptographic community.

One of the key advantages of the secp256r1 curve is its resistance to cryptographic attacks. Its parameters and structure have been meticulously chosen to maximize security, making it a reliable option for protecting sensitive information. This resistance to attacks ensures the confidentiality, integrity, and authenticity of data in our project's communications. By utilizing the secp256r1 curve in our project, we ensure a high level of security while maintaining efficient performance. Its standardization, efficiency, widespread use, and resistance to attacks make it a dependable choice for implementing encryption and other cryptographic operations in our healthcare communication platform.

The proposed model uses this encryption method to generate a public key from a private key.

$$y^2 = x^3 + ax + b$$

Here, “mod p” is called the remainder operation (or modulo), and in the case of “a mod n,” it is the remainder of a divided by n. In other words, the above equation can be said to be an equation in which the remainders of the left and right sides are equal.

Prime

(p):0xffffffff00000001000000000000000000000000ffffffffffffffffffffffff

Coefficient

a:0xffffffff00000001000000000000000000000000fffffffffffffffffffffffc

b:0x5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604b

Base Point (G):

gx:0x6b17d1f2e12c4247f8bce6e563a440f277037d812deb33a0f4a13945d898c296

gy:0x4fe342e2fe1a7f9b8ee7eb4a7c0f9e162bce33576b315ececbb6406837bf51f5

Order:

(n):0xffffffff00000000ffffffffffffffffbce6faada7179e84f3b9cac2fc632551

Cofactor:

(h): 0x1

Add G, which is itself, to G to obtain 2G (addition in elliptic curve cryptography)By repeating the action of 2. n times (n: private key value), the value nG is obtained. Set nG as the public key value

Elliptic Curve Cryptography (ECC) operates on the principles of algebraic geometry, where addition on the curve follows unique rules divergent from conventional arithmetic. The process begins with a point (G) on the curve. A tangent line at (G) intersects the curve at another point. The reflection of this intersection point across the x-axis yields the resultant point (2G). This operation, known as point doubling, is the cornerstone of ECC and is repeated to compute multiples of (G), such as (nG).
The elliptic curve's equation includes a modulus (p), indicating that all operations are performed within a finite field. This means that the coordinates of any point on the curve are the remainders when divided by (p). Given a base point (G) and a prime number (p), ECC involves iterating the point addition operation (n) times to obtain the point (nG). While calculating (nG) is straightforward, the reverse process—deriving (n) from (G) and (nG)—is computationally infeasible for large (n), akin to the one-way nature of hash functions. This characteristic of ECC makes it a powerful tool for creating secure cryptographic systems, as it allows for the generation of a public key from a private key in a manner that is secure against cryptanalysis. Elliptic curves, despite their mathematical complexity, are thus fundamental in modern cryptography for ensuring robust security in the generation and management of cryptographic keys.

Advanced encryption standard (AES)

In 2001, the National Institute of Standards and Technology (NIST) in the USA introduced the Advanced Encryption Standard (AES), replacing the outdated Data Encryption Standard (DES). AES quickly gained global acceptance and is now widely used in various products worldwide. AES employs a symmetric key technique, meaning the same key is used for both encryption and decryption. This key remains constant throughout the process. AES operates as a block encryption algorithm, dividing input messages into 128-bit blocks and converting each block into ciphertext individually.
Digital information can be encrypted using the AES algorithm, a symmetric block cipher capable of encrypting and decrypting data simultaneously. The algorithm transforms data into ciphertext, making it unreadable to humans, but it can be decrypted back into its original form. AES supports key lengths of 128, 192, and 256 bits, with the longest key length being used for military-grade protection. It is highly resistant to known attacks and operates efficiently across various platforms.

AES encryption involves multiple rounds, each composed of various components that perform specific functions. The number of rounds depends on the key length: 10 rounds for 128 bits, 12 rounds for 192 bits, and 14 rounds for 256 bits. AES encrypts all data in a single pass, making it efficient and fast.
The encryption and decryption processes involve two main components:

1. Plain Text Transformation: This includes round key addition, column confusion, row shifting, and byte substitution.
2. Key Expansion: This involves expanding the key and generating subkeys.

AES employs a 16-byte key and 10 rounds of transformation. While AES involves numerous transformations and complex key expansion, it does not utilize the Feistel structure like DES. Instead, AES operates on plaintext in matrix form. The encryption and decryption processes include:

Substitute Bytes: Substituting bytes using an S-box to complete byte-to-byte substitution.
Shift Rows: Simple row shifting for replacement.
Column Confusion (Mix Columns): Utilizing finite fields GF (28) for column mixing (except for the last round).
Add Round Key: Performing a bitwise XOR operation between the current group and expansion keys.

AES's encryption and decryption processes ensure robust security and efficient data protection, making it a widely adopted standard in cryptographic applications worldwide. In our project, we utilize 256-bit encryption, which involves 14 rounds for the encryption process. Each round ensures robust security and thorough mixing of data, enhancing confidentiality and integrity. This approach provides a high level of protection for sensitive information exchanged between parties.

Bouncy Castle & Spongy Castle

Bouncy Castle (BC) is a set of easy-to-use cryptography APIs. It is implemented in both Java and C#. It is a provider for the Java Cryptography Extension and the Java Cryptography Architecture. It also contains a lightweight cryptography API.

The Android platform unfortunately ships with a cut-down version of Bouncy Castle - as well as being rippled also makes installing an updated version of the libraries difficult due to class loader conflicts.

VII. PROPOSED SYSTEM

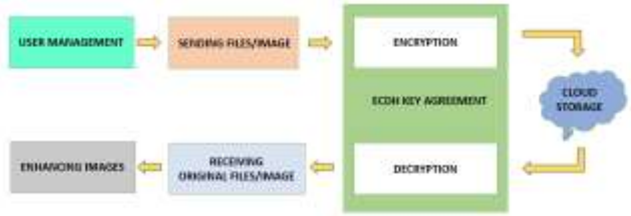


Figure 1.2: Overall Block Diagram

ENCRYPTION

The secure communications between a doctor and a patient, the generation of an Elliptic Curve Cryptography (ECC) key pair is the foundational step. Each party creates a private key, which remains confidential, and a corresponding public key that can be openly shared. These keys are pivotal for the encryption and decryption of their communications. Once the key pairs are generated, the next phase involves the encryption and exchange of public keys. The AES encryption algorithm is employed to encrypt the public keys before they are exchanged, ensuring that the communication channel remains secure.

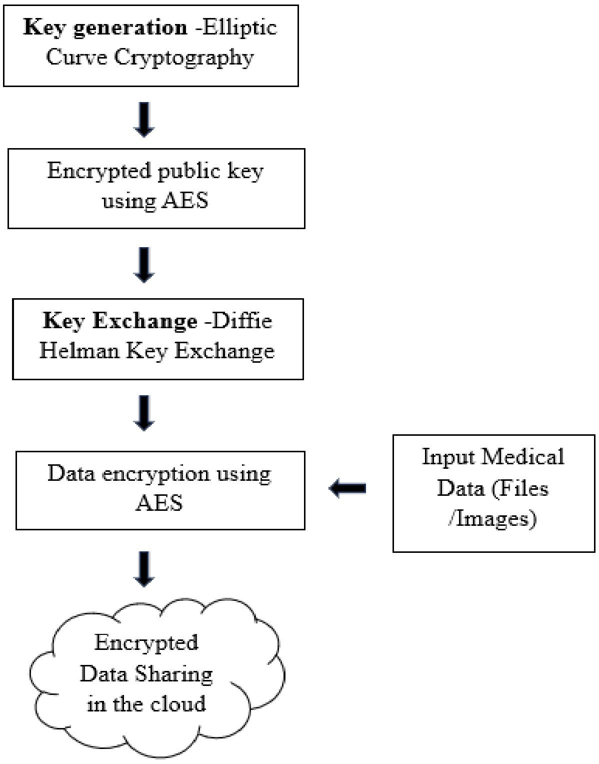

Key generation -Elliptic Curve Cryptography
Encrypted public key using AES
Key Exchange -Diffie Helman Key Exchange
Data encryption using AES
Input Medical Data (Files /Images)
Encrypted Data Sharing in the cloud

Figure 1.3: Encryption Block

Following the exchange, the Elliptic Curve Diffie-Hellman (ECDH) key agreement protocol comes into play. Both the doctor and the patient utilize their private keys and the other's public key to compute a shared secret. This shared secret is not transmitted but rather derived independently by both parties, forming the cornerstone of their secure communication. The received public keys, initially encrypted, are then decrypted using the AES algorithm. This decryption allows for the use of the public keys in subsequent cryptographic operations, such as further ECDH key agreements to reinforce the secure connection.

The shared secret, borne out of the ECDH agreement, is then used to derive encryption keys. This unique key is instrumental in ensuring that only the doctor and patient can access the encrypted data, maintaining the confidentiality of the information exchanged.

This shared secret is also utilized as the key for the Advanced Encryption Standard (AES) algorithm. It becomes the essential tool for encrypting and decrypting sensitive information, such as medical records, shared between the doctor and patient. In the final step, the patient encrypts their medical records or other sensitive information using the AES key derived from the shared secret. This encryption is crucial for safeguarding the data during transmission and storage. The resulting ciphertext is then securely transferred over the established communication channel, effectively preventing unauthorized access to the patient's sensitive information. This meticulous process ensures that the privacy and security of the patient's medical records are upheld throughout their digital interactions with the doctor.

DECRYPTION

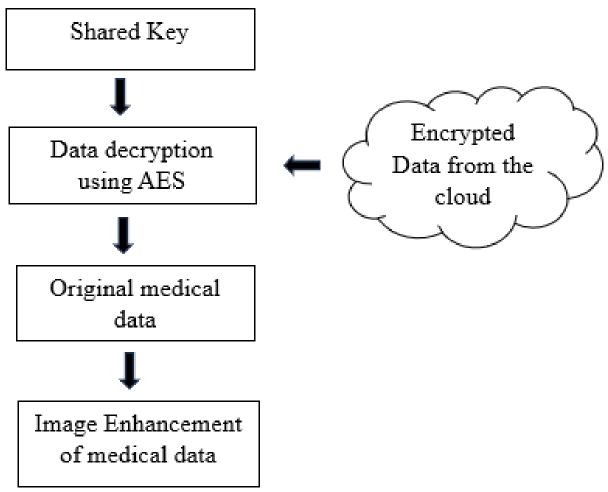

Shared Key
Data decryption using AES
Original medical data
Image Enhancement of medical data
Encrypted Data from the cloud

Figure 1.4: Decryption Block

The doctor starts the decryption process after obtaining the ciphertext in order to view the patient's shared medical records. In order to compute the shared secret, which is essential for decryption and maintaining data secrecy, they must complete the ECDH key agreement using their private key and the patient's public key. The physician then securely accesses the medical records while protecting confidentiality by using the shared secret as the AES key to decrypt the received ciphertext. The doctor uses the AES key generated from the shared secret to decrypt the ciphertext after the key agreement. The decryption procedure guarantees the safe retrieval of medical records while preserving data security and integrity. The doctor receives the decrypted plaintext medical records, which are now available in their original format, following a successful decryption. This allows the doctor to review their medical information securely and confidentially. Important elements are added to enhance the clarity and quality of the photographs once the original image has been decrypted, resulting in a complete picture improvement system. In order to simplify future analysis and standardize image data, key aspects of the image are preserved while color complexity is removed through the grayscale conversion process by averaging RGB values. The mirror effect raises the overall aesthetic quality and visual appeal of an image by turning it horizontally along its vertical axis, so enhancing symmetry and aesthetic appeal. Negative inversion is a useful tool for altering colors by subtracting the maximum amount of pixel intensity. This technique enhances contrast and creates creative effects in photographs.

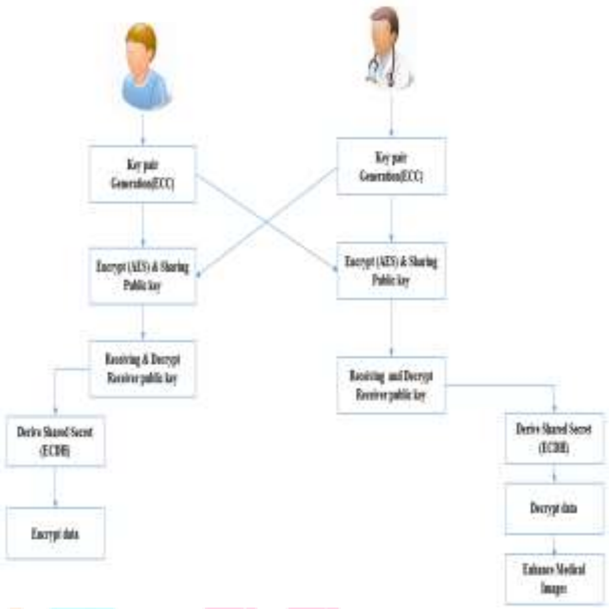

Key pair Generation(ECC)
Key pair Generation(ECC)
Encrypt (AES) & Sharing Public key
Encrypt (AES) & Sharing Public key
Receiving & Decrypt Receiver public key
Receiving and Decrypt Receiver public key
Derive Shared Secret (ECDH)
Derive Shared Secret (ECDH)
Encrypt data
Decrypt data
Enhance Medical Images

Figure 1.5: System architecture

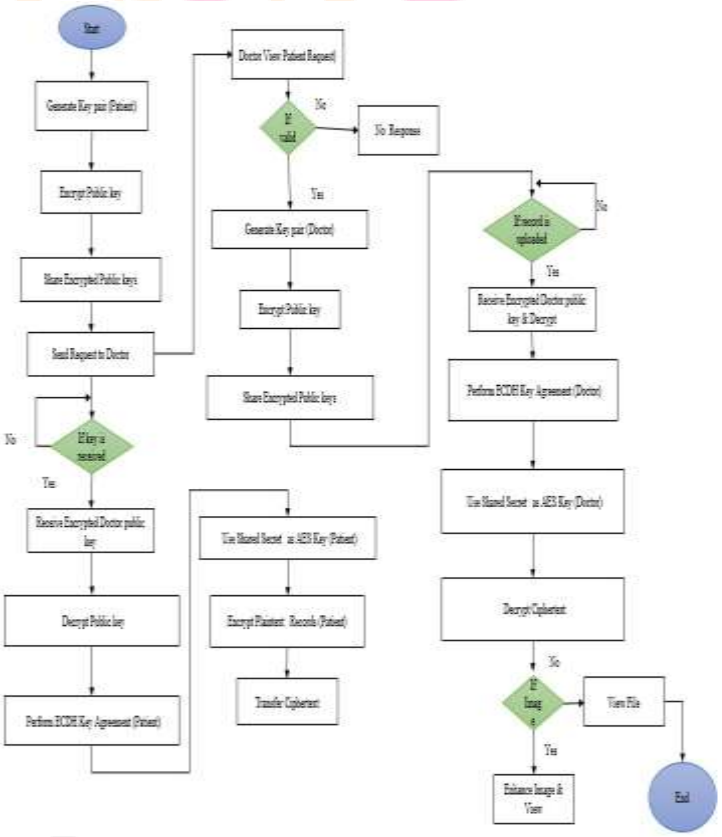

Start
Generate Key pair (Patient)
Encrypt Public key
Share Encrypted Public keys
Send Request to Doctor
If key is received
No
Yes
Receive Encrypted Doctor public key
Decrypt Public key
Perform ECDH Key Agreement (Patient)
Doctor View Patient Request
If valid
No
No Response
Yes
Generate Key pair (Doctor)
Encrypt Public key
Share Encrypted Public keys
Use Shared Secret as AES Key (Patient)
Encrypt Plaintext Records (Patient)
Transfer Ciphertext
If record is uploaded
No
Yes
Receive Encrypted Doctor public key & Decrypt
Perform ECDH Key Agreement (Doctor)
Use Shared Secret as AES Key (Doctor)
Decrypt Ciphertext
No
If Image
View File
Yes
Enhance Image & View
End

Figure 1.6: Flowchart

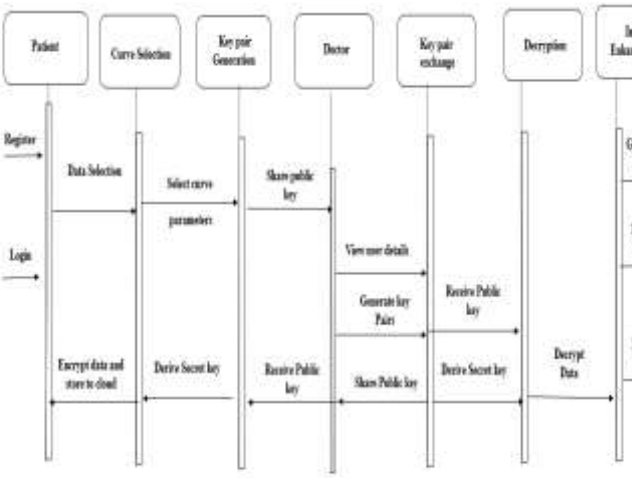


Figure 1.7: Sequence Diagram

VII **RESULTS AND DISCUSSION**

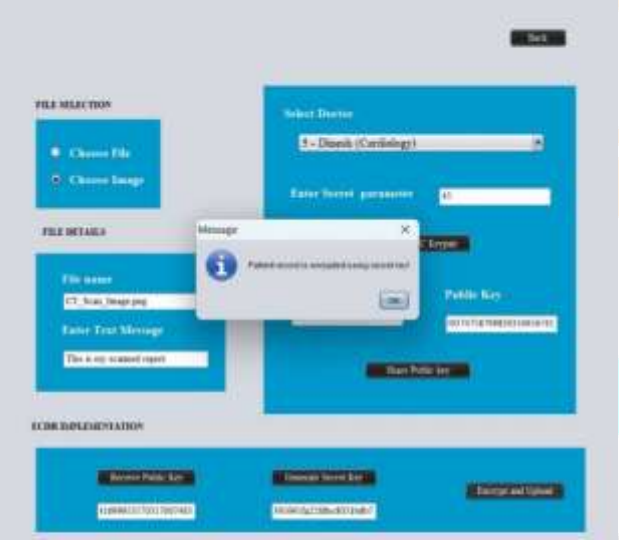


Figure 1.8: Patients page

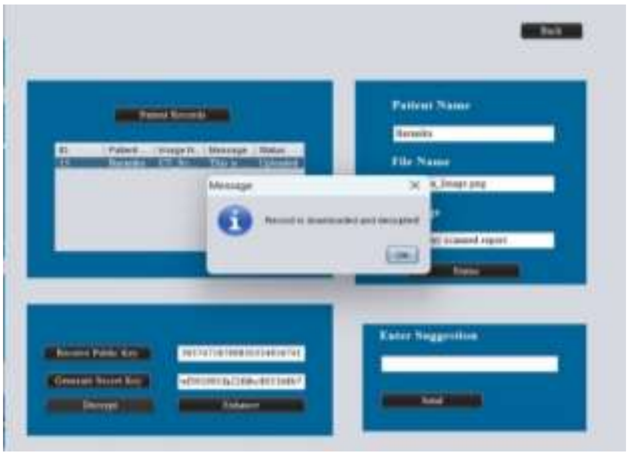


Figure 1.9: Doctor decrypt and view page

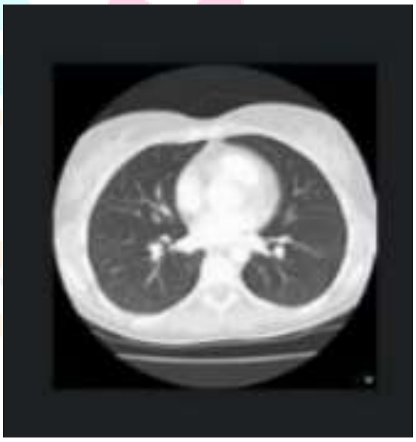
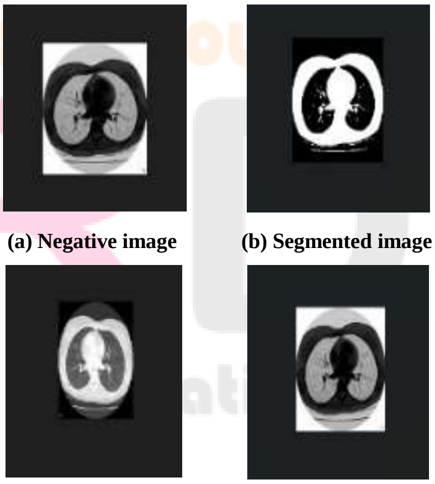


Figure 1.10: Decrypted Image



(a) Negative image (b) Segmented image

(c) Mirrored image (d) Gray scaled image

Figure 1.11: Enhanced image

VIII CONCLUSION

In conclusion, the proposal to adopt Elliptic Curve Diffie-Hellman (ECDH) cryptography for health record exchange offers a robust solution to address the security vulnerabilities inherent in existing systems. ECDH provides strong encryption and efficient key exchange mechanisms, ensuring secure communication channels between healthcare providers and patients. By exclusively implementing ECDH, the solution simplifies the encryption process while effectively mitigating the risks associated with cyber threats and unauthorized access. This streamlined approach enhances data privacy and integrity without relying on additional encryption algorithms, thereby reducing complexity and implementation overhead. Furthermore, the incorporation of image enhancement techniques during decryption underscores the system's commitment to improving the quality of medical images. Overall, the integration of ECDH cryptography presents a comprehensive and efficient solution for securing health record exchange in healthcare, safeguarding sensitive patient information and promoting trust in healthcare systems.

IX FUTURE WORK

In our upcoming research, we're focused on enhancing healthcare data security through advanced encryption technologies. Our main focus will be on developing homomorphic encryption and post-quantum cryptography to strengthen our secure communication system against advanced cryptographic attacks. Simultaneously, our goal is to use machine learning to support surgical judgment and improve patient outcomes. In addition, we intend to implement biometric authentication to firmly limit medical data access to verified individuals only. Our goal is to put in place a real-time monitoring system that can quickly detect and alert users to security breaches in order to support these efforts. We hope to improve the security of surgical procedures by enabling encrypted exchange of patient data and instructions during robotic surgery sessions by integrating our secure communication system. Elliptic Curve Diffie-Hellman (ECDH), an asymmetric encryption technique, will be used to accomplish this, guaranteeing the integrity and secrecy of sent data. We anticipate substantial progress in protecting private health information with this multipronged strategy, maintaining patient privacy and data accuracy in an increasingly digitalized medical environment.

X REFERENCE

1. R. Imam, Q. M. Areeb, A. Alturki, and F. Anwe, “Systematic and critical review of RSA based public key cryptographic schemes”, IEEE Access, vol. 9, pp. 155949-155976,Nov.2021,[online]Available:https://ieeexplore.ieee.org/document/9620070.

2. V. Giri and A. S. R. Murty, “Elliptical Curve Cryptography Design Principles,” 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Aug. 20211. [online] Available: https://ieeexplore.ieee.org/document/9573662.

3. A. A. Asaker, Z. F. Elsharkawy, S. Nassar, N. Ayad, and O. Zahra, “A Novel Iris Cryptosystem Using Elliptic Curve Cryptography,” year of publication, [Online]. Available: https://ieeexplore.ieee.org/document/9691307.

4. Rismayani, C. Susanto, “Using AES and DES Cryptography for System Development File Submission Security Mobile-Based,” 2020, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9268805.

5. Yang Ming, Tingting Zhang, “Efficient Privacy-Preserving Access Control Scheme in Electronic Health Records System”, Sensors, vol. 18, no. 10, Article 3520, Oct. 2018, [online]Available:https://www.mdpi.com/14248220/18/10/3520.

6. J. Smith et al., “A Comparative Study of ECDH Implementations for Secure Communication in IoT Environments,”2020,[online]Available:https://link.springer.com/article/10.1007/s11277-021-08439-7.

7. Alice Johnson et al., “Secure Communication Protocol Design Using ECDH for Resource-Constrained Environments,” Proceedings of the International Conference on Big Data, IoT, and Machine Learning, pp. 431-444, 2019, [online]Available:https://link.springer.com/chapter/10.1007/978-981-16-6636-0_33.

8. Michael Brown et al., “Vulnerabilities and Countermeasures in ECDH Implementations: A Survey,” Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security, pp. 1-15, 2017, [online]Available:https://link.springer.com/content/pdf/10.1007/978-3-319-66787-4_26.pdf.

9. John Doe et al., “Efficient Implementation of Elliptic Curve Diffie-Hellman Algorithm for Secured Communication,” Proceedings of the 25th ACM SIGSAC Conference on Computer and Communications Security, pp .1-16,2018, [online] Available: https://www.mdpi.com/2079-9292/12/21/4480.

10. Li et al., “A Survey on Secure Communication Techniques for Patients’ Health Records in Connected Healthcare Systems,” Proceedings of the 29th ACM SIGSAC Conference on Computer and Communications Security, pp. 114,2022,[online]Available:https://link.springer.com/article/10.1007/s11042-020-10083-5.

11.D.Eastlake,Transport_Layer_Security(TLS)_Extensions: ExtensionDefinitions,Jan.2011,[online]Available:https://www.rfceditor.org/rfc/rfc6066.html.

12. K83120834: Diffie–Hellman key agreement protocol weaknesses CVE-2002-20001 & CVE-2022-40735, May 2022,[online]Available:https://my.f5.com/manage/s/article/K83120834.

13.M. Friedl, N. Provos and W. Simpson, Diffie–Hellman Group Exchange for the Secure Shell (SSH) Transport Layer Protocol, Mar. 2006, [online] Available: https://www.rfc-editor.org/rfc/rfc4419.txt#section-6.2.

14. D. Gillmor, Negotiated Finite Field Diffie–Hellman Ephemeral Parameters for Transport Layer Security (TLS), Aug.2018,[online]Available:https://www.rfceditor.org/rfc/rfc7919.txt.

15. GnuTLS 3.6.0 Released, Dec. 2023, [online] Available: https://lists.gnupg.org/pipermail/gnutls-devel/2017August/008484.html.

16. Maximum Prime Modulues Size in GnuTLS Version 3.7.9, [online] Available: https://gitlab.com/gnutls/gnutls/-/blob/3.7.9/lib/gnutls_int.h#L230.

17.D. Harkins and D. Carrel, The Secure Shell (SSH) Transport Layer Protocol, Nov. 1998, [online] Available: https://www.rfc-editor.org/rfc/rfc2409.