

A
Project Report

on

ENHANCED MEDICAL DATA SECURITY FRAMEWORK

Submitted in partial fulfillment of the requirements for the award of the degree of
Bachelor of Technology

by

Mounika Bandaru

(20EG105404)

Surya Teja Kancha

(20EG105418)

Manvitha Peddapurapu

(20EG105434)



Under the guidance of

Dr.K.Madhuri

Associate Professor

Department of CSE

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ANURAG UNIVERSITY
VENKATAPUR (V), GHATKESAR (M), MEDCHAL (D), T.S - 500088
TELANGANA
(2023-2024)**

DECLARATION

We hereby declare that the report entitled “**Enhanced Medical Data Security Framework**” submitted to the **Anurag University** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology (B.Tech)** in **Computer Science and Engineering** is a record of original work done by us under the guidance of **Dr.K.Madhuri, Associate Professor** and this report has not been submitted to any other university for the award of any other degree or diploma.

Place: Anurag University, Hyderabad

Mounika Bandaru
(20EG105404)

Date:

Surya Teja Kancha
(20EG105418)

Manvitha Peddapurapu
(20EG105434)



CERTIFICATE

This is to certify that the project report entitled “**Enhanced Medical Data Security Framework**” being submitted by **Mounika Bandaru, Surya Teja Kancha, Manvitha Peddapurapu** bearing the Hall Ticket numbers **20EG105404, 20EG105418, 20EG105434** respectively in partial fulfillment of the requirements for the award of the degree of the **Bachelor of Technology in Computer Science and Engineering** to **Anurag University** is a record of bonafide work carried out by them under my guidance and supervision from 2023 to 2024.

The results presented in this report have been verified and found to be satisfactory. The results embodied in this report have not been submitted to any other University for the award of any other degree or diploma.

Dr.K.Madhuri
Supervisor

Dr. G. Vishnu Murthy
Dean, Department of CSE

External Examiner 1

ACKNOWLEDGMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Dr.K.MADHURI** , Associate Professor, Department of Computer Science and Engineering, Anurag University for her constant encouragement and inspiring guidance without which this project could not have been completed. Her critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work. Her patience, guidance and encouragement made this project possible.

We would like to acknowledge our sincere gratitude for the support extended by **Dr.G.VISHNU MURTHY**, Dean, Department of Computer Science and Engineering, Anurag University. We also express our deep sense of gratitude to **Dr. V. V. S. S. S. BALARAM**, Academic coordinator. **Dr. PALLAM RAVI**, Project Coordinator and project review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stages of our project work.

We would like to express our special thanks to **Dr. V. VIJAYA KUMAR**, Dean School of Engineering, Anurag University, for his encouragement and timely support in our B. Tech program.

Mounika Bandaru
(20EG105404)

Surya Teja Kancha
(20EG105418)

Manvitha Peddapurapu
(20EG105434)

ABSTRACT

Medical field deals with a lot of physical actions, reactions and responses. Most of the communication between doctor and the patient is not digital. But, it would be easier, if it was. This framework allows the patient and the doctor to have contact over a network. It also follows the security parameters, confidentiality, authentication and integrity, by encryption, access control policy and, key management. Only the authorized doctor will be able to view the patient file. This helps both the parties involved, as the doctor does not need to view every patient file and the patient need not worry about the security.

Keywords- Confidentiality, security, Encryption, Decryption, Key management.

TABLE OF CONTENT

| S. No. | CONTENT | Page No. |
|---------------|--|-----------------|
| 1. | Introduction | 1 |
| | 1.1. Data Attributes | 1 |
| | 1.2. Security Parameters | 6 |
| | 1.2.1 Confidentiality | 6 |
| | 1.2.2 Authentication | 7 |
| | 1.2.3 Integrity | 8 |
| | 1.3. Problem Definition | 9 |
| | 1.4. Problem Illustration | 9 |
| | 1.5. Objective of the Project | 11 |
| 2. | Literature Survey | 12 |
| 3. | Enhanced Medical Data Security Framework | 24 |
| | 3.1. Registration and Login | 24 |
| | 3.2. File Upload and Key Generation | 25 |
| | 3.2.1. AES Algorithm | 26 |
| | 3.3. Key Requests and Receiving Files | 30 |
| | 3.4. Prescriptions | 31 |
| | 3.5. Overview | 32 |
| 4. | Implementation | 34 |
| | 4.1. Functionalities | 34 |
| | 4.1.1. Encryption and Decryption | 34 |
| | 4.1.2. Key Management | 34 |
| | 4.1.3. Access Control | 35 |
| | 4.1.4. Prescription Upload | 36 |
| | 4.1.5. Encryption Time Calculation | 36 |
| | 4.1.6. Secure Communication | 37 |
| | 4.1.7. Logging and Auditing | 37 |
| | 4.1.8. User Interface | 38 |

| | |
|----------------------------------|----|
| 4.1.9. Secure Storage | 39 |
| 4.2. Attributes | 40 |
| 4.3. Experimental Screenshots | 42 |
| 5. Experimental Setup | 47 |
| 5.1. NetBeans | 47 |
| 5.2. Backend Development | 47 |
| 5.3. Frontend Development | 48 |
| 5.4. Database setup | 49 |
| 5.5 Integration and Testing | 50 |
| 5.6 Deployment | 51 |
| 5.7 Monitoring and Maintenance | 52 |
| 5.8. Libraries used | 53 |
| 5.8.1. For JSP codes | 53 |
| 5.8.2. Encryption and Decryption | 56 |
| 5.8.3. Database Connection | 57 |
| 5.9. Parameters | 58 |
| 6. Discussion of Results | 60 |
| 7. Conclusion | 62 |
| 8. Future Enhancements | 63 |
| 9. References | 64 |

List of Figures

| Figure No. | Figure Name | Page No. |
|-------------------|---------------------------|-----------------|
| 1.1.1 | Concept Tree | 1 |
| 1.4.1 | Existing Algorithm | 9 |
| 1.4.2 | Existing Algorithm graph | 10 |
| 1.5.1 | An example | 11 |
| 3.2.1 | Successful Key Generation | 25 |
| 3.2.1.1 | AES Algorithm | 27 |
| 3.3.1 | Key not matched | 30 |
| 3.4.1 | Database examples | 31 |
| 3.5.1 | Concept Overview | 32 |
| 3.5.2 | Illustration | 33 |
| 4.3.1 | Doctor Registration page | 42 |
| 4.3.2 | Doctor Login page | 42 |
| 4.3.3 | Patient Registration page | 43 |
| 4.3.4 | Patient Login page | 43 |
| 4.3.5 | Patient File Upload | 44 |
| 4.3.6 | Patient side- View data | 44 |
| 4.3.7 | Patient side - View Keys | 44 |
| 4.3.8 | Doctor side- send request | 45 |
| 4.3.9 | Patient side- send key | 45 |
| 4.3.10 | Mail | 45 |
| 4.3.11 | File Download | 46 |

| | | |
|--------|-----------------------|----|
| 4.3.12 | Prescription | 46 |
| 4.3.13 | Prescription view | 46 |
| 6.1 | Encryption time graph | 61 |
| 6.2 | Decryption Time Graph | 62 |

List of Tables

| Table No. | Table Name | Page No. |
|------------------|--------------------------------|-----------------|
| 1.4.1 | Encryption and Decryption Time | 10 |
| 6.1 | Encryption Time | 60 |
| 6.2 | Decryption Time | 61 |

List of Abbreviations

| Abbreviations | Full Form |
|----------------------|------------------------------|
| EHR | Electronic health records |
| RBAC | Role-based access control |
| MFA | Multi-factor authentication |
| AES | Advanced Encryption Standard |
| ABE | Attribute-based encryption |

| | |
|------|------------------------------------|
| IDE | Integrated Development Environment |
| JCA | Java Cryptography Architecture |
| JDBC | Java Database Connectivity |
| CRUD | Create, Read, Update, Delete |
| UI | User Interface |

1. Introduction

Medical data stands out as a highly confidential and crucial subset, encompassing a patient's health records, medical history, diagnosis, and prescriptions. The sensitivity of this data necessitates stringent privacy measures, restricting access solely to the patient and their designated healthcare provider. Data, a versatile information resource, is particularly critical in the context of medical information, encompassing patient health records and other sensitive details. The rising importance of preserving the confidentiality of medical data has led to the development of many strategies. These strategies include secure storage, encryption, access control policies, and regulated data sharing, ensuring that only authorized individuals can access and interact with the information.

1.1. Data Attributes

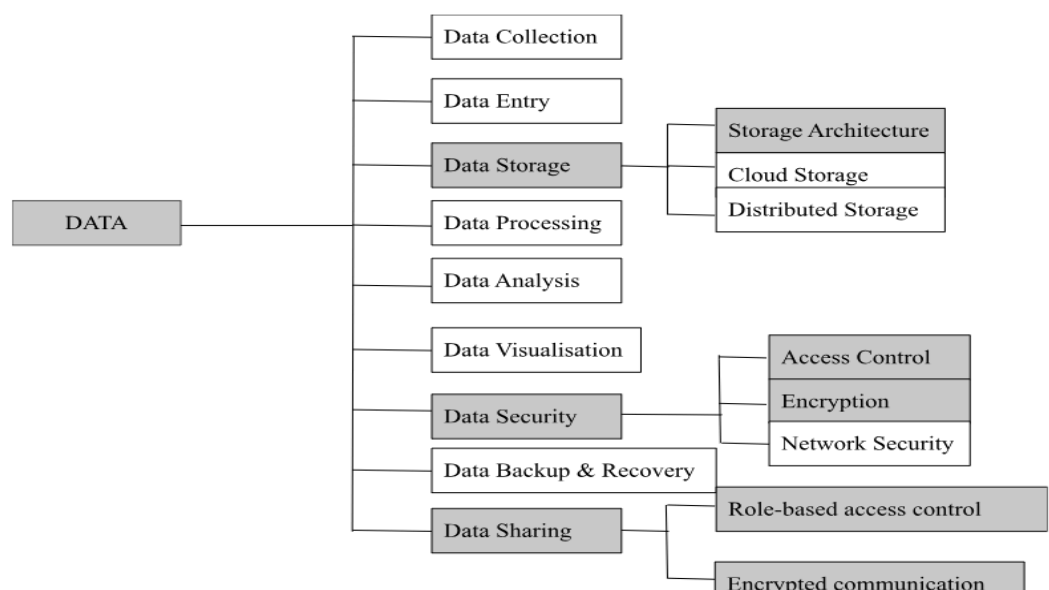


FIGURE 1.1.1 CONCEPT TREE

Data can be operated and managed through different functions. These functions are collection, entry, storage, security, sharing and many more. Medical data will be mainly associated with storage, security and sharing. These three functions along with other functions are important to run communication across the internet.

Data Storage

Data storage in medical healthcare is crucial for managing various types of patient information, including electronic health records (EHRs), diagnostic images, lab results, and treatment plans. It involves securely storing and managing sensitive data to ensure accessibility, interoperability, compliance with regulations such as HIPAA, and protection against unauthorized access and cyber threats. Cloud storage solutions offer scalability and cost-effectiveness, while data backup and disaster recovery plans are essential for ensuring data availability and integrity. Interoperability enables seamless data sharing among healthcare providers, and data analytics techniques can be applied to derive insights for improving patient care and outcomes.

Efficient data storage facilitates access to accurate patient information, enabling healthcare providers to make informed decisions and deliver timely and personalized care. It supports health information exchange and interoperability, promoting better coordination and continuity of care among different healthcare providers involved in a patient's treatment. Digital data storage solutions reduce paperwork, and lower operational costs for healthcare organizations. If healthcare data is stored properly it can be used for research purposes, supporting medical discoveries, treatment advancements, and public health initiatives. Robust data storage and backup strategies ensure data availability and integrity in the event of hardware failures, natural disasters, or cyberattacks, minimizing disruptions to patient care. Analyzing stored healthcare data can provide valuable insights for improving clinical outcomes, optimizing healthcare delivery, and addressing population health challenges.

Data Security

Data security is paramount in medical healthcare due to the sensitive nature of patient information. Access control and encryption are critical components of ensuring the confidentiality, integrity, and availability of healthcare data. Access control mechanisms regulate who can access patient data, ensuring that only authorized personnel can view or modify sensitive information. Encryption involves encoding data in a way that can only be decoded by authorized parties, thus protecting it from unauthorized access or interception. Data security measures, such as access control and encryption, safeguard patient confidentiality by limiting access to authorized individuals and preventing unauthorized disclosure of sensitive medical information.

Access control prevents unauthorized users from accessing patient data, reducing the risk of data breaches and the associated consequences, such as identity theft, fraud, and reputational damage. Those mechanisms help mitigate insider threats by limiting employees' access to only the information necessary to perform their job duties. Encryption ensures that even if data is accessed by unauthorized insiders, it remains protected from unauthorized disclosure. If patient data is encrypted it remains secure during transmission over networks, such as the internet or internal networks, protecting it from interception by malicious actors. Strong data security measures, like access control and encryption, build trust among patients, healthcare providers, and other stakeholders by demonstrating a commitment to protecting patient privacy and confidentiality.

Data Sharing

Data sharing in medical healthcare involves the exchange of patient information, clinical data, research findings, and other relevant healthcare data among healthcare providers, researchers, and other stakeholders. It facilitates coordinated patient care, enables clinical research and innovation, supports public health efforts, and improves healthcare delivery and outcomes. Data sharing enables healthcare providers to access relevant patient information, such as medical history, diagnoses, and treatment plans, facilitating more informed and coordinated patient care across different healthcare settings. Sharing healthcare data supports clinical research and innovation by providing researchers with access to large and diverse datasets for analysis, leading to advancements in medical knowledge, treatments, and healthcare practices.

Data sharing plays a crucial role in public health surveillance efforts by enabling the timely detection and monitoring of disease outbreaks, trends, and other health threats, allowing for effective public health interventions and responses. Sharing clinical and outcome data among healthcare providers and institutions allows for benchmarking, performance measurement, and quality improvement initiatives, leading to better healthcare delivery and patient outcomes. It can empower patients by giving them greater access to their own health information, enabling them to make more informed decisions about their care, participate in research, and engage with healthcare providers as active partners in their health. Interoperable data sharing systems enable seamless exchange of patient information among different healthcare systems and

providers, improving care coordination, reducing duplication of services, and enhancing the overall patient experience. It can also lead to increased efficiency in healthcare delivery processes, reduced administrative burdens, and cost savings by eliminating redundancies and streamlining workflows. The development of personalized medicine approaches is supported by providing insights into individual patient characteristics, genetic profiles, and treatment responses, allowing for more tailored and effective healthcare interventions.

Role-based Access Control

Role-based access control or RBAC is a method of restricting system access to authorized users based on their roles within an organization. Each user is assigned one or more roles, and access rights are granted to these roles rather than to individual users. It helps organizations manage and enforce access controls efficiently, improve security, simplify administration, and ensure compliance with regulatory requirements. Risk of unauthorized access can be reduced by ensuring that users only have access to the resources and data necessary for their roles. This helps prevent data breaches, insider threats, and other security incidents. RBAC simplifies the management of access controls by allowing administrators to assign permissions to roles rather than individual users. This streamlines user provisioning, access changes, and revocation processes, reducing administrative overhead and errors. It allows organizations to define fine-grained access controls based on users' roles and responsibilities. This enables more precise control over who can access specific resources, perform certain actions, or view sensitive information.

RBAC provides flexibility in managing access permissions as organizational roles evolve or change. Administrators can easily update role assignments and access rights to accommodate changes in users' responsibilities or business requirements. It enhances auditability and accountability by providing a clear record of who has access to which resources and what actions they are authorized to perform. This supports compliance audits, incident investigations, and accountability mechanisms. RBAC is scalable and can accommodate organizations of all sizes, from small businesses to large enterprises. It can easily scale to manage access controls for hundreds or thousands of users and resources without sacrificing security or performance. It also ensures that users have access to the resources they need to perform their roles

effectively, without unnecessary restrictions or delays. This enhances productivity and user satisfaction while maintaining security and compliance.

Encrypted communication

Encrypted communication plays a crucial role in medical healthcare by ensuring the confidentiality, integrity, and security of sensitive patient information exchanged between healthcare professionals, patients, and other stakeholders. It involves encoding data transmitted over networks using encryption algorithms, making it unreadable to unauthorized parties. It helps protect patient privacy, prevent unauthorized access to medical data, comply with regulatory requirements, and mitigate the risk of data breaches and cyberattacks. Encrypted communication safeguards patient privacy by preventing unauthorized access to sensitive medical information transmitted over networks. It ensures that only authorized recipients can access and interpret the data, enhancing patient confidentiality and trust.

Encrypted communication helps protect medical data from interception, eavesdropping, and tampering during transmission over insecure networks, such as the internet or wireless networks. It prevents malicious actors from accessing or manipulating patient information, reducing the risk of data breaches and cyberattacks. With the increasing adoption of telemedicine and remote healthcare services, this type of communication ensures the secure transmission of patient data between healthcare providers and patients, regardless of their location. It enables remote consultations, telemonitoring, and electronic health record access without compromising data security. Encrypted communication demonstrates a commitment to protecting patient data and maintaining confidentiality, which enhances trust and confidence among patients, healthcare professionals, and other stakeholders. It helps build a positive reputation for healthcare organizations and fosters strong patient-provider relationships. Encrypted communication enables secure data exchange and interoperability among different healthcare systems, platforms, and devices. It facilitates the seamless sharing of patient information across healthcare settings, improving care coordination, clinical decision-making, and patient outcomes. In addition to confidentiality, encrypted communication ensures the integrity of medical data by detecting and preventing unauthorized modifications or alterations during

transmission. It helps maintain the accuracy and reliability of patient information, reducing the risk of errors and misinformation in healthcare practices.

1.2 Security Parameters

This approach concentrates on the three security parameters, confidentiality, authentication and integrity. These parameters are the basic units to establish any security framework. Considering them, the framework has been designed. It also encompasses a secure infrastructure of the storage of patient information to ensure everything is only visible to the authorized. It employs encryption protocols to safeguard the information when it is transmitted. Access control policies are implemented to regulate user access.

Security parameters refer to the various aspects, measures, or configurations that contribute to the security of a system, network, or application. These parameters are typically set, monitored, and managed to protect against unauthorized access, data breaches, cyber threats, and other security risks.

1.2.1 Confidentiality

Confidentiality is a critical security parameter in medical healthcare, primarily focused on protecting sensitive patient information from unauthorized access, disclosure, or tampering. It ensures that patients have control over who can access their medical information, including diagnoses, treatment plans, test results, and personal demographics. Respecting patient privacy is a fundamental ethical principle in healthcare. Patients are more likely to share sensitive information with healthcare providers if they trust that their information will remain confidential. Maintaining confidentiality fosters trust between patients and healthcare professionals, facilitating open communication and better patient outcomes. Confidentiality measures, such as access controls, encryption, and secure authentication, prevent unauthorized individuals from accessing patient information. This protects against data breaches, identity theft, fraud, and other security incidents that can compromise patient confidentiality.

Medical information is highly sensitive and can be exploited for various malicious purposes, such as insurance fraud, blackmail, or discrimination. Confidentiality measures help safeguard this information, reducing the risk of harm to patients and

maintaining their dignity and autonomy. Healthcare professionals have a duty to maintain the confidentiality of patient information as part of their professional ethics. Breaching patient confidentiality not only violates trust but also undermines the integrity and reputation of healthcare providers and organizations. Confidentiality measures enable secure data sharing and collaboration among healthcare professionals, researchers, and other stakeholders while ensuring that patient privacy is protected. This facilitates interdisciplinary care, medical research, and public health initiatives without compromising confidentiality. Confidentiality breaches can have severe consequences for patients, healthcare providers, and organizations, including financial losses, legal consequences, damage to reputation, and loss of trust. Implementing robust confidentiality measures helps mitigate these risks and protects the interests of all stakeholders.

1.2.2 Authentication

Authentication is a vital security parameter in medical healthcare, serving to confirm the identity of users accessing sensitive patient information, medical records, and healthcare systems. Authentication ensures that only authorized healthcare professionals can access patient records, protecting patient confidentiality and privacy. It prevents unauthorized individuals from viewing or tampering with sensitive medical information, such as diagnoses, treatments, and test results. Authentication mechanisms, such as passwords, biometrics, and multi-factor authentication (MFA), prevent unauthorized access to healthcare systems and patient records. Strong authentication measures mitigate the risk of data breaches, identity theft, fraud, and other security incidents. Authentication verifies the identity of users accessing healthcare systems or patient information, ensuring that they are who they claim to be. This helps prevent impersonation and unauthorized access by malicious actors attempting to exploit vulnerabilities in the system.

Authentication systems provide an audit trail of user activities, allowing organizations to track and monitor who accessed patient records, when, and for what purpose. This promotes accountability and helps identify and investigate security breaches or policy violations. With the rise of telemedicine and remote healthcare services, authentication enables secure access to patient records and healthcare systems from remote locations. Healthcare professionals can authenticate themselves using secure methods, such as VPNs or secure authentication tokens, to ensure the confidentiality

and integrity of patient data during remote consultations or telehealth sessions. Authentication facilitates user management processes, such as user provisioning, access control, and account deactivation. Healthcare organizations can grant or revoke access privileges based on user roles, responsibilities, and authorization levels, ensuring that only authorized personnel have access to sensitive information. Authentication systems can be integrated with identity management systems to centralize user authentication, streamline user provisioning and deprovisioning, and enforce security policies consistently across healthcare systems and applications.

1.2.3 Integrity

Integrity is a crucial security parameter in medical healthcare, ensuring that patient data remains accurate, complete, and unaltered throughout its lifecycle. Maintaining data integrity is critical for patient safety, as inaccuracies or alterations in medical records could lead to incorrect diagnoses, treatments, or medications. Ensuring the integrity of patient data helps healthcare providers make informed decisions and deliver appropriate care. Patients and healthcare providers rely on the accuracy and reliability of medical records to make informed decisions about treatment plans, medications, and interventions. Data integrity builds trust in the healthcare system and fosters confidence in the quality of care provided. Ensuring data integrity helps prevent data corruption or loss due to accidental errors, software glitches, hardware failures, or cyberattacks. By maintaining data accuracy and consistency, healthcare organizations can minimize the risk of data loss and ensure data availability when needed.

Integrity mechanisms provide an audit trail of data modifications, allowing organizations to track and monitor changes to patient records over time. This promotes accountability and helps identify unauthorized alterations, policy violations, or suspicious activities that may compromise data integrity. Integrity measures are essential for securing medical devices, systems, and software applications used in healthcare settings. Ensuring the integrity of software updates, patches, and configurations helps prevent unauthorized modifications that could introduce vulnerabilities or compromise patient safety. Integrity mechanisms validate the accuracy and consistency of patient data through validation checks, checksums, digital signatures, and cryptographic hashes. These techniques verify that data has not been altered or corrupted during transmission, storage, or processing.

1.3 Problem Definition

Many healthcare organizations struggle to effectively encrypt sensitive medical data stored in text files. Without robust encryption measures, patient information is vulnerable to unauthorized access, data breaches. The absence of access control policies exacerbates the risk of unauthorized access to sensitive medical data. Without proper access controls, there is no mechanism in place to restrict access to authorized individuals or organizations, leaving patient information vulnerable to unauthorized viewing, modification, or disclosure. Patients and healthcare providers require a secure communication interface to exchange sensitive medical information, discuss treatment plans, and address healthcare concerns. However, the current lack of such interfaces leaves communication channels vulnerable to interception, eavesdropping, or tampering by unauthorized parties. In the current healthcare landscape, sensitive medical data must be securely managed and communicated to ensure patient privacy and regulatory compliance. However, existing methods fall short in providing adequate encryption for text files, implementing access control policies, and facilitating secure communication between patients and healthcare providers. This poses a significant challenge in establishing comprehensive data security and communication channels in medical healthcare settings.

1.4 Problem Illustration

The system is associated with providing security to image and audio files from implantable devices in the human body via the internet. Algorithms like RC6, ECDSA, and SHA-256 are combined as a hybrid algorithm. RC6 algorithm is used for generating a key value, and ECDSA encrypts the first generated key value then the encrypted key value is sent to a secure hash algorithm.

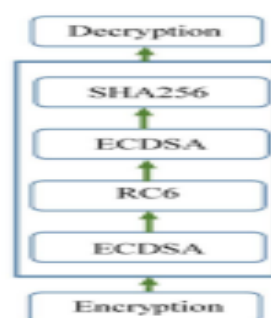


FIGURE 1.4.1 EXISTING ALGORITHM

| Input File Type | Encryption(MS) | Decryption(MS) |
|-----------------|----------------|----------------|
| 1 | 50 | 90 |
| 2 | 120 | 160 |
| 3 | 207 | 210 |
| 4 | 250 | 285 |
| 5 | 303 | 315 |
| 6 | 320 | 397 |
| 7 | 325 | 403 |
| 8 | 420 | 545 |
| 9 | 510 | 629 |
| 10 | 621 | 717 |

TABLE 1.4.1 ENCRYPTION AND DECRYPTION TIME

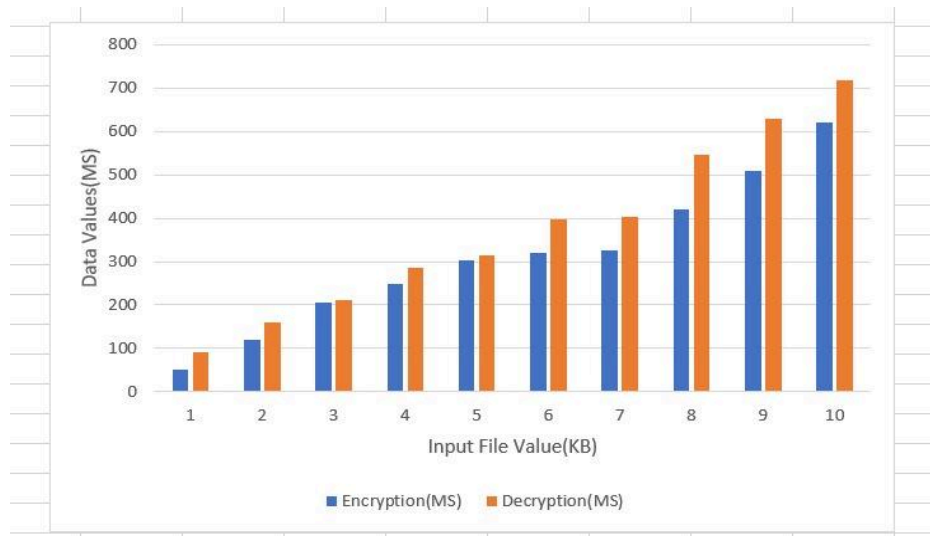


FIGURE 1.4.2 EXISTING ALGORITHM GRAPH

The whole process of using three algorithms for encryption and decryption is complex and time-consuming. The major limitation of the existing system is there is no interface or framework to connect the sender and receiver of the files.

1.5. Objective Of The Project

The objective of the project is to provide an interface in which the patient and the doctor can communicate in a secure and comfortable way. The project will contribute to the exchange of information through text files which has not been explored in existing systems. It is to find a better algorithm that is less complex to replace the time consuming algorithm.

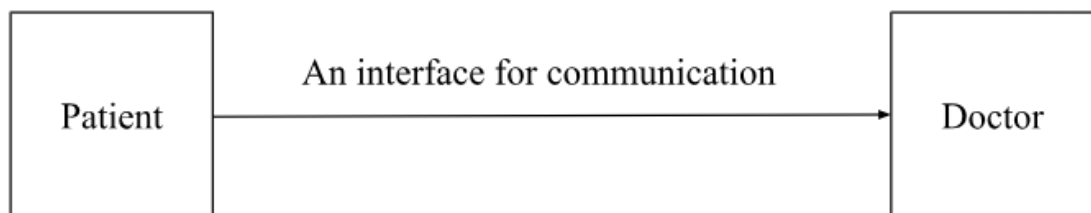


FIGURE 1.5.1 AN EXAMPLE

To establish communication for users,i.e, for patients and doctors, an interface is required for them. The project will try to achieve this along with security approaches. The time analysis using various algorithms have been looked into and in this project the chosen algorithm must take less time to encrypt and decrypt.

2. Literature Survey

Senthil Murugan Nagarajan's approach to securing data transmission in the Internet of Medical Things (IoMT)[1]with the utilization of the RES-256 algorithm demonstrates a commitment to ensuring the confidentiality and integrity of medical data as it traverses the internet. The RES-256 algorithm, likely a cryptographic method, employs a robust 256-bit key length to provide a high level of security, which is essential for safeguarding sensitive medical information such as patient health records and diagnostic data. By implementing RES-256, the IoMT ecosystem aims to prevent unauthorized access to medical data and protect against tampering or manipulation of information during transmission. This is particularly crucial in healthcare settings where maintaining the privacy and security of patient data is paramount to comply with regulations and uphold ethical standards. However, it's important to recognize that effective implementation of cryptographic algorithms like RES-256 requires more than just selecting a strong encryption method. Proper key management practices are essential to ensure the security and integrity of encrypted data. Without secure key management, the cryptographic strength provided by RES-256 could be compromised, leaving medical data vulnerable to unauthorized access or manipulation. Moreover, in resource-constrained environments, such as certain IoT devices commonly found in IoMT deployments, the computational demands of RES-256 may present challenges. The processing power required to perform encryption and decryption operations could potentially impact device performance, leading to latency issues or reduced responsiveness. As such, it's crucial to carefully consider the computational requirements and performance implications when deploying RES-256 in resource-constrained environments within the IoMT ecosystem. Additionally, while RES-256 addresses the confidentiality and integrity of medical data during transmission, it's important to note that text data, such as clinical notes, reports, and patient communications, also represent critical components of medical information. Ensuring the security of text data within the IoMT ecosystem requires comprehensive encryption and access control measures tailored to the specific characteristics and requirements of textual information.

The paper "Emerging security mechanisms for medical cyber-physical systems" [2] authored by O. Kocabas, T. Soyata, and M. K. Aktas, delves into the realm of security

advancements tailored for medical cyber-physical systems (MCPS). These systems amalgamate computing, communication, and control facets to augment healthcare services, promising transformative impacts in patient care. The paper accentuates the imperative of mitigating vulnerabilities and fortifying the integrity, confidentiality, and reliability of data within MCPS. Within the domain of MCPS, where interconnected devices and systems underpin critical healthcare operations, safeguarding against unauthorized access, manipulation, or disruptions is paramount. By leveraging robust security measures, MCPS endeavors to fortify patient safety by erecting formidable barriers against potential threats. Central to these security measures is the preservation of the confidentiality of sensitive patient data. This entails deploying encryption protocols, access controls, and data segregation techniques to assuage privacy concerns and adhere to stringent regulatory mandates such as HIPAA and GDPR. Moreover, the proactive stance adopted by MCPS in addressing emerging cyber threats underscores its resilience in the face of evolving risks within the healthcare landscape. By continuously evolving security mechanisms, MCPS remains steadfast in its commitment to shield against novel threats, ensuring the robustness and reliability of healthcare services. However, the implementation of these advanced security solutions may present formidable challenges. The complexity of deploying and managing sophisticated security mechanisms necessitates specialized skills and resources, which may be beyond the purview of smaller healthcare facilities with limited expertise and financial constraints. Indeed, the adoption of advanced security measures entails substantial investments in technology, training, and maintenance, potentially posing a financial burden for healthcare organizations with constrained budgets. Despite the transformative potential of these security mechanisms, their adoption may be hindered by cost considerations, necessitating strategic planning and resource allocation to surmount these barriers effectively.

The paper titled "A secure IoT-based modern healthcare system with fault-tolerant decision-making process," [3] co-authored by P. Gope, Y. Gheraibia, S. Kabir, and B. Sikdar, offers insights into the development of a healthcare system that harnesses the Internet of Things (IoT) for connectivity and data exchange, while also integrating fault-tolerant decision-making processes. In contemporary healthcare landscapes, the integration of IoT technologies holds significant promise for revolutionizing patient

care delivery, facilitating remote monitoring, and enhancing treatment outcomes. By leveraging IoT-enabled devices and sensors, healthcare systems can collect real-time patient data, enabling healthcare providers to make informed decisions and deliver personalized care interventions. Central to the design of the proposed healthcare system is the emphasis on security to safeguard sensitive healthcare information against unauthorized access, data breaches, or malicious attacks. Robust security measures, including encryption, access controls, and intrusion detection systems, are implemented to ensure the confidentiality, integrity, and availability of patient data throughout its lifecycle. Moreover, the incorporation of fault-tolerant decision-making processes is pivotal for enhancing the system's resilience to errors or failures. By employing redundancy, error detection, and error correction mechanisms, the system can mitigate the impact of faults or failures, ensuring that critical decisions can still be made accurately and promptly even in adverse conditions. Seamless communication and data exchange among various components of the healthcare system facilitate improved coordination among healthcare providers, enabling timely access to pertinent patient information and fostering collaborative care delivery. This promotes a patient-centric approach to healthcare, where interventions are tailored to individual patient needs based on real-time data insights. Furthermore, a secure IoT-based healthcare system has the potential to revolutionize healthcare delivery by enabling remote monitoring, proactive disease management, and early intervention strategies. Patients can benefit from continuous monitoring of vital signs and health metrics, leading to improved health outcomes, reduced hospital admissions, and enhanced quality of life. However, the design, deployment, and maintenance of a secure IoT-based healthcare system may necessitate advanced technical expertise, specialized skills, and resources. Healthcare organizations must invest in training programs and collaborate with experts in cybersecurity and IoT technologies to ensure the successful implementation and operation of such systems.

The research paper titled "Secure health data sharing for medical cyber-physical systems for the healthcare 4.0,"[4] conducted by H. Qiu, M. Qiu, M. Liu, and G. Memmi, delves into the intricacies of designing and implementing secure health data sharing mechanisms within the framework of medical cyber-physical systems (MCPS), with a particular emphasis on the emerging paradigm of Healthcare 4.0. Healthcare 4.0 represents a transformative shift in healthcare delivery, characterized

by the integration of advanced technologies such as the Internet of Things (IoT), artificial intelligence (AI), and data analytics to enhance patient care, improve clinical outcomes, and optimize healthcare processes. In this context, the paper explores novel approaches to securely share health data within the interconnected and technologically advanced ecosystem of Healthcare 4.0. The proposed method adopts a user-centric design philosophy, wherein the security and privacy of health data are prioritized, and end-users are empowered to control access to their data. This is achieved by leveraging trusted devices, such as the end-users' smartphones, as secure enclaves for storing and managing sensitive health information. By entrusting users with the authority to manage data access permissions, the proposed approach not only enhances data privacy but also fosters a sense of ownership and control over personal health information. This user-centric model aligns with evolving regulatory frameworks, such as GDPR and HIPAA, which emphasize the importance of individual consent and data sovereignty in healthcare data sharing practices. Furthermore, the research evaluates the performance of the proposed algorithm on smartphone platforms to demonstrate its efficiency and suitability for real-world deployment. By conducting performance evaluations in practical settings, the paper aims to validate the scalability, reliability, and computational efficiency of the proposed solution, thereby laying the groundwork for its widespread adoption in Healthcare 4.0 environments.

The paper titled "Secure Medical Data Sharing For Healthcare System"[5] authored by Zina Chkirbene, Ridha Hamila, and Aiman Erbad introduces a mechanism designed to address the challenges associated with storing, sharing, and securing large volumes of data in healthcare systems. In today's healthcare landscape, the adoption of advanced information technologies has revolutionized the delivery of healthcare services, necessitating efficient mechanisms for managing and protecting sensitive medical data. The proposed mechanism employs a specialized approach to data splitting and encryption, ensuring that the distributed data can only be accessed and reconstructed with the approval of a predetermined number of trusted nodes. This distributed storage and access control model enhances data security by preventing unauthorized access or tampering, even if individual nodes within the network are compromised. A key feature of the proposed mechanism is its robustness to failures and network constraints. To optimize the selection of nodes participating in the data

storage process, an iterative algorithm is proposed. This algorithm balances the need for fault tolerance and network efficiency, minimizing transmission delays while ensuring data availability and integrity. Simulation results validate the effectiveness and efficiency of the proposed approaches in securely storing and sharing data among legitimate nodes within the network. By demonstrating the feasibility and performance of the proposed mechanism through simulations, the paper provides valuable insights into its practical applicability in real-world healthcare systems. The paper presents a comprehensive solution to the challenges of secure distributed data storage and sharing in healthcare systems. By leveraging innovative data splitting, encryption, and access control techniques, coupled with optimized node selection algorithms, the proposed mechanism offers a robust and efficient framework for managing and protecting sensitive medical data. This contributes to the advancement of secure and efficient healthcare information management systems, ultimately improving the quality and accessibility of healthcare services for patients and providers alike.

The EPI Framework [6], as developed by J. Kassem, C. De Laat, A. Taal, and P. Grosso, has garnered significant attention in recent research on dynamic data sharing frameworks for healthcare applications. This framework serves as a model for enabling the secure and efficient exchange of healthcare data among various stakeholders. Researchers have extensively explored different approaches and technological advancements to address the challenges of interoperability, privacy, and scalability inherent in healthcare data sharing. Interoperability remains a key concern in healthcare data sharing, as diverse systems and formats often hinder seamless exchange. The EPI Framework and similar approaches aim to overcome these challenges by facilitating interoperability through standardized protocols, data formats, and communication interfaces. Privacy is another critical aspect addressed by dynamic data sharing frameworks. With sensitive patient information at stake, ensuring privacy and confidentiality is paramount. Techniques such as encryption, access controls, and anonymization are integrated into these frameworks to safeguard patient data from unauthorized access or disclosure. Scalability is also a key consideration, especially with the increasing volume and complexity of healthcare data. The EPI Framework and related solutions are designed to scale effectively to accommodate growing data volumes and diverse user requirements, ensuring efficient

data sharing across large healthcare networks. Various strategies have been explored to enhance the security and adaptability of healthcare data sharing frameworks. Federated learning, blockchain technology, and hybrid cloud architectures are among the innovative approaches investigated to create secure and resilient environments for data exchange. Moreover, researchers emphasize the importance of adopting patient-centric methodologies and complying with legal standards such as the Health Insurance Portability and Accountability Act (HIPAA) when designing and implementing these frameworks. Prioritizing patient privacy, consent, and control over their data is essential for building trust and ensuring ethical data sharing practices.

The research conducted by N. Tsafack et al. sheds light on the growing interest in leveraging chaotic maps for encryption applications in the realm of the Internet of Health Things (IoHT) [7]. Chaotic maps, known for their complex and unpredictable behavior, have been studied extensively to enhance encryption methods aimed at safeguarding data transferred within IoT networks, particularly in healthcare contexts. In response to the increasing security risks posed by unauthorized access and data breaches in IoHT systems, researchers have explored various chaotic maps and their dynamic characteristics. By leveraging the inherent randomness and complexity of chaotic maps, scholars aim to develop robust encryption algorithms capable of thwarting potential security threats and ensuring the confidentiality and integrity of sensitive health data. The integration of chaotic maps into encryption systems represents a promising approach to bolstering the durability and effectiveness of current security mechanisms employed in IoHT environments. By incorporating chaotic maps into encryption algorithms, researchers seek to enhance the resilience of healthcare data protection measures, thereby mitigating the risk of unauthorized access or manipulation. The review of the literature underscores the critical role of encryption in safeguarding private health data within IoHT systems. As the healthcare industry continues to embrace IoT technologies for remote monitoring, telemedicine, and personalized healthcare, the need for robust encryption solutions becomes increasingly imperative to preserve patient privacy, comply with regulatory requirements, and maintain trust in healthcare services.

The survey conducted by S. Wijethilaka and M. Liyanage delves into the burgeoning field of network slicing for Internet of Things (IoT) implementation within 5G networks [8]. Network slicing has emerged as a pivotal technology in the context of 5G, enabling the efficient allocation of resources to cater to the diverse requirements of IoT applications. Researchers have shown a keen interest in exploring various aspects of network slicing to effectively support a wide array of IoT use cases. The survey encompasses investigations into the design and implementation of network slicing frameworks tailored specifically for IoT applications. These frameworks aim to address critical challenges such as resource allocation, scalability, and quality of service (QoS) provisioning, which are essential for ensuring seamless and reliable connectivity for IoT devices. A key focus of research efforts is on enhancing IoT service delivery and performance through the integration of network slicing with cutting-edge technologies such as edge computing and artificial intelligence (AI). By leveraging edge computing capabilities, network slicing can optimize data processing and analysis closer to the point of data generation, reducing latency and enhancing responsiveness for IoT applications. Additionally, AI-driven approaches can further enhance network slicing mechanisms by enabling dynamic resource allocation, predictive analytics, and autonomous decision-making to meet the evolving demands of IoT environments. The survey underscores the importance of network slicing in enabling the realization of diverse IoT applications within 5G networks. By providing a comprehensive overview of research advancements and emerging trends in this domain, the survey serves as a valuable resource for researchers, practitioners, and industry stakeholders seeking to leverage network slicing for IoT deployment in 5G networks.

The study conducted by R. Saha, G. Geetha, G. Kumar, T.-H. Kim, and W. J. Buchanan on "Mrc4: A modified RC4 algorithm using symmetric random function generator for improved cryptographic features" highlights the ongoing efforts in the literature to enhance the security and efficiency of cryptographic algorithms [9]. In particular, researchers have focused on refining existing algorithms like RC4 to address vulnerabilities and bolster cryptographic capabilities. One of the key areas of exploration involves introducing modifications to established algorithms to mitigate weaknesses and enhance cryptographic characteristics. By refining algorithms like RC4, researchers aim to improve their resistance against potential attacks and ensure

the confidentiality and integrity of encrypted data. A notable aspect of this research is the exploration of symmetric random function generators to augment the unpredictability and strength of encryption protocols. By incorporating these generators into modified algorithms like Mrc4, researchers seek to enhance the randomness of encryption keys, thereby fortifying the security of cryptographic operations. Moreover, there is a concerted effort to balance algorithmic efficiency with security considerations. While enhancing cryptographic features, researchers also strive to optimize the computational performance of algorithms to ensure practical usability in real-world applications. This entails minimizing computational overhead while maintaining robust security protocols.

The research conducted by M. A. Siddiqi, C. Doerr, and C. Strydis on "Imdfence: Architecting a secure protocol for implantable medical devices"[10] sheds light on the growing concerns surrounding the security of implantable medical devices (IMDs). As the use of IMDs becomes increasingly prevalent in healthcare settings, ensuring the security and integrity of these devices has emerged as a critical priority. The literature review highlights a growing awareness of potential cybersecurity threats associated with IMDs, prompting researchers to investigate various aspects of IMD security comprehensively. Key focus areas include communication protocols, authentication methods, and encryption approaches tailored specifically for IMDs. Researchers endeavor to identify and address inherent weaknesses in these devices, proposing innovative solutions to enhance their security posture and mitigate potential risks. In healthcare environments, ensuring patient privacy, data integrity, and device reliability are paramount considerations. Researchers recognize the importance of implementing robust security measures to safeguard sensitive patient information and prevent unauthorized access or tampering with IMDs. By establishing strong security standards and protocols tailored specifically for IMDs, researchers aim to uphold patient privacy, maintain data integrity, and ensure the safe and effective operation of these devices within healthcare ecosystems.

The research conducted by Lim, C.K., Ipinje, V.J., Tan, K.L., & Hambira, N. [11] on "Design and development of message authentication process for telemedicine application" provides valuable insights into the realm of message authentication methods specifically tailored for telemedicine applications. Telemedicine, with its

increasing importance in modern healthcare delivery, necessitates robust security measures to ensure the authenticity and integrity of transmitted data. The study likely delves into various authentication techniques, such as digital signatures, cryptographic protocols, and other security mechanisms, that are crucial for securing communications in telemedicine scenarios. By exploring existing research and advancements in this field, the research aims to identify effective authentication methods suitable for the unique requirements of telemedicine applications. Telemedicine poses distinct challenges, including the need for real-time communication, ensuring patient data privacy, and mitigating the risks associated with security breaches in healthcare contexts. The review likely discusses these challenges in detail, highlighting the importance of addressing them to ensure the successful implementation of telemedicine solutions. Furthermore, the research may shed light on emerging technologies and standards in the field of telemedicine authentication. By examining these advancements, the study provides valuable insights into potential solutions and areas for further research and development aimed at enhancing the security and reliability of telemedicine applications.

"LightIoT: Lightweight and secure communication for energy-efficient IoT in health informatics" by Jan, M.A., Khan, F., Mastorakis, S., Adil, M., Akbar, A., & Stergiou, N. [12] likely encompasses a comprehensive review of existing research and developments in lightweight and secure communication protocols specifically tailored for energy-efficient Internet of Things (IoT) applications in health informatics. It probably examines various communication protocols, encryption techniques, and authentication methods optimized for resource-constrained IoT devices in healthcare settings. The survey may delve into the challenges associated with energy efficiency, security, and scalability in IoT-based health informatics systems, while also exploring the potential benefits of adopting lightweight communication protocols such as LightIoT. Additionally, it might discuss relevant standards, emerging technologies, and best practices in the field, providing insights into current trends and future directions for research and implementation in this domain.

The "Man-in-the-Middle attack mitigation in the Internet of Medical Things (IoMT)" [13] is probably going to entail a thorough analysis of the methods and research that have already been done to address security flaws related to Man-in-the-Middle

(MitM) attacks in IoMT systems. It would investigate different MitM attack scenarios, like listening in on conversations between servers and medical devices or introducing malicious material into Internet of medical things traffic, which might jeopardize patient privacy and data integrity. A review of various mitigation strategies and solutions, including digital signatures, encryption protocols (like SSL/TLS), secure communication protocols (like MQTT with TLS), and authentication mechanisms (like mutual authentication, certificate-based authentication), would probably be included in the survey. It may also cover methods for safe key management, safe device bootstrapping, and customized intrusion detection/prevention systems.

"Optimal haptic communications over nanonetworks for e-health systems"[14] will entail. In order to transmit tactile feedback over networks, it would first explore the field of haptic communication and look at the devices, protocols, and methods now in use. Research on force feedback systems, haptic rendering methods, and tactile displays may fall under this category. Subsequently, the survey is expected to explore nanonetworks, with a particular emphasis on communication protocols and technologies intended for inter-nanoscale device communication. This could include studying molecular communication, nanoscale electromagnetic communication, nanoscale networking topologies, and other nanoscale communication approaches.

A thorough analysis of research on secure IoT-based healthcare systems and fault-tolerant decision-making processes is part of the literature review for the topic "A secure IoT-based modern healthcare system with fault-tolerant decision-making process." [15] In addition to studies on encryption methods, authentication protocols, and access control mechanisms intended to protect sensitive healthcare data, it reviews the body of literature already in existence on IoT designs, communication protocols, and security mechanisms customized for healthcare applications. The survey also explores algorithms for anomaly detection, redundancy mechanisms, and resilience measures to minimize failures in IoT-based healthcare environments, as well as fault-tolerant decision-making processes. The goal of the survey is to find methods that combine fault-tolerant decision-making processes with secure communication protocols in order to improve the resilience and dependability of

healthcare systems. This will be accomplished by examining the relationship between security and fault tolerance.

An extensive review of research on authentication protocols, elliptic curve cryptography (ECC), anonymity techniques, and wearable health monitoring systems is probably what the literature survey for the topic "An elliptic curve cryptography-based enhanced anonymous authentication protocol for wearable health monitoring systems" [16] entails. First, it would examine current authentication procedures designed with wearable health monitoring systems in mind, taking efficiency, security, and anonymity into account. Studies on more modern advancements in anonymous authentication protocols as well as more conventional techniques like password-based or biometric authentication may fall under this category. Subsequently, the survey will explore the topic of elliptic curve cryptography, analyzing its benefits for security and effectiveness in contrast to alternative cryptographic methods. This could entail examining ECC-based authentication systems and how wearable technology can use them.

A review of studies on RFID (Radio Frequency Identification) systems, anonymous authentication protocols, and physically unclonable functions (PUFs) would probably be included in the literature review for the topic "Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions"[17]. It would first review the body of research on RFID technology, emphasizing its uses, security issues, and authentication techniques. Research into RFID tag architectures, communication protocols, and vulnerabilities including tag cloning and eavesdropping may fall under this category. Second, the survey would explore the state of the art in anonymous authentication techniques, especially those intended for practical and low-cost integration into RFID systems. This could entail researching methods for maintaining scalability and efficiency in RFID systems while maintaining anonymity, taking into account things like communication overhead and processing overhead.

"Fog computing for 5G tactile Industrial Internet of Things (IIoT): QoE-aware resource allocation model"[18] would include an extensive analysis of studies pertaining to fog computing, quality of experience (QoE)-aware resource allocation

models, 5G networks, and tactile IIoT applications. Initially, it would review the literature that has already been written about fog computing, emphasizing its tenets, designs, and benefits for Internet of Things applications—particularly those in industrial settings. Studies on the positioning of fog nodes, methods for processing data, and approaches for reducing latency can fall under this category. Second, the study will explore studies on 5G networks, looking at their characteristics, capabilities, and future improvements to support haptic IIoT applications with high dependability and low latency. This could entail investigating radio access, edge computing capabilities, and 5G network slicing.

"Multiview Cauchy estimator feature embedding for depth and inertial sensor-based human action recognition"[19]. First, it would examine the body of research on human action recognition, with a particular emphasis on techniques that make use of inertial sensors, depth sensors, or a mix of the two. To increase recognition accuracy, this might involve research on multimodal fusion techniques, inertial sensor data fusion methods, and depth-based skeletal models. Second, the survey would probably look on feature embedding techniques, especially those that aim to extract discriminative data from various perspectives or modalities. This could entail looking at feature extraction methods like sparse coding and deep learning architectures.

3. Enhanced Medical Data Security Framework

The proposed attribute-based encryption technique aims to secure patient data in a telemedicine or healthcare system. In this approach, patient data is encrypted as a text file using the Advanced Encryption Standard (AES) encryption technique, which is widely recognized for its strong security properties. To control access to the encrypted patient data, access permissions are assigned based on specific attributes or criteria. For example, access may be granted only to the respective doctor responsible for the patient's care. This ensures that unauthorized users, such as other healthcare professionals or external parties, cannot access the sensitive patient information. Users who have been granted access permissions can view the encrypted patient data. However, in order to download and decrypt the data for viewing, they must first request a security key from the data owner. This request process adds an extra layer of security by requiring explicit permission from the data owner before accessing the decrypted patient data. Importantly, access control policies are enforced to ensure that only authorized users with the appropriate attributes or credentials are granted access to the security key. If a user's access control policy is not satisfied (e.g., they do not have the necessary credentials or permissions), they will be denied access to the security key and, consequently, unable to download and view the patient data. This attribute-based encryption technique provides a secure and flexible means of protecting patient data in healthcare systems, ensuring that only authorized users can access and view sensitive information while maintaining strict control over access permissions and data security.

3.1 Registration and Login

The proposed method introduces an interface or framework designed to streamline communication between patients and their requested doctors in a healthcare setting. To access this interface, both patients and doctors are required to either register or log in. Upon accessing the interface, users are directed to the login page, which serves as the initial point of entry. Here, doctors are prompted to input their email and password. If a doctor is not yet registered, there is an option for new users to click and complete the registration process. During registration, doctors are required to provide

various details, including their name, email, a new password, date of birth, and contact number. An essential attribute section prompts doctors to specify their specialty, making it easier to match cases with relevant specialists. Once all required information is provided, the registration process is completed, and the login page reappears. Subsequently, doctors can log in by entering the password created during registration, gaining access to the interface. This interface serves as a platform for facilitating seamless communication and collaboration between patients and doctors, optimizing the delivery of healthcare services.

3.2 File Upload and Key generation

Following patient registration and login, the platform offers patients a straightforward process for uploading their medical concerns and generating unique encryption keys. the patient gains access to a platform where they can upload a text file containing their medical concerns. In addition to uploading the file, the patient is required to input their ID and select the type of doctor they wish to consult. Once uploaded, the patient can then view the data within the platform. A key generation option is provided, allowing the patient to generate a unique key for the uploaded file. Upon clicking this option, the platform confirms successful key generation with a prompt displaying "Key generated successfully".

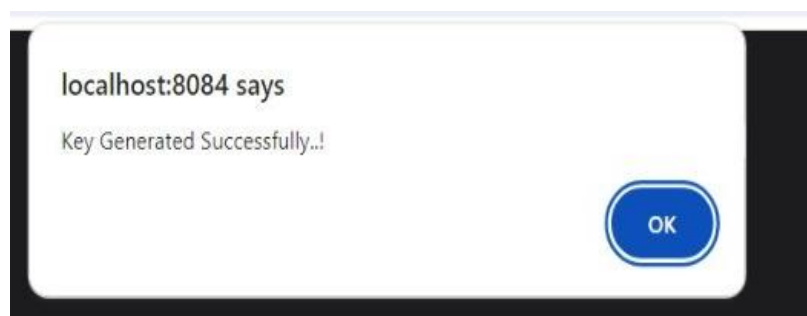


FIGURE 3.2.1 SUCCESSFUL KEY GENERATION

Furthermore, the patient can access a section where all generated keys are listed, along with details such as the encryption time of the data and the corresponding cipher text. This ensures the security and integrity of patient data while enabling efficient communication with healthcare providers.

3.2.1 AES Algorithm

AES (Advanced Encryption Standard) operates on a fixed block size of 128 bits and supports three different key lengths: 128, 192, or 256 bits. The key expansion process transforms the original encryption key into a set of round keys, which are crucial for the encryption and decryption procedures. The encryption process begins with the Initial Round, where the initial round key, derived from the encryption key, is added to the plaintext block. Following this, multiple Rounds are executed, the number of which depends on the key length: 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds for a 256-bit key. Each round of the AES encryption process further iterates through four transformation stages.

SubBytes

In the SubBytes stage, each byte of the state matrix is substituted with a corresponding byte from the S-box, a predefined lookup table. This substitution provides non-linearity and confusion, enhancing the security of the encryption.

The S-box used in AES replaces each byte with another byte based on a fixed transformation. This transformation is designed to resist cryptanalysis techniques such as linear and differential attacks.

ShiftRows

The ShiftRows operation cyclically shifts the rows of the state matrix to the left. Each row is shifted by a varying number of positions, with the second row shifted by one position, the third row by two positions, and the fourth row by three positions. This permutation ensures that each byte of data is spread out across different rows, providing diffusion and making the encryption process more resistant to linear and differential cryptanalysis.

MixColumns

In the MixColumns stage, a matrix multiplication operation is applied to each column of the state matrix. This operation provides diffusion by mixing the bytes within each column. The MixColumns transformation ensures that changes in one byte affect

multiple bytes in subsequent rounds, thereby increasing the complexity of the encryption process and making it more resistant to various cryptographic attacks.

AddRoundKey

In the AddRoundKey stage, each byte of the state matrix is XORed with a corresponding byte from the round key derived during the key expansion process. This stage introduces the key material into the state matrix, ensuring that the encryption is performed using both the original key and additional key material generated during the key expansion process.

The Final Round of the encryption process excludes the MixColumns transformation. After this final round, the resulting state matrix represents the ciphertext. During decryption, the process reverses the encryption steps.

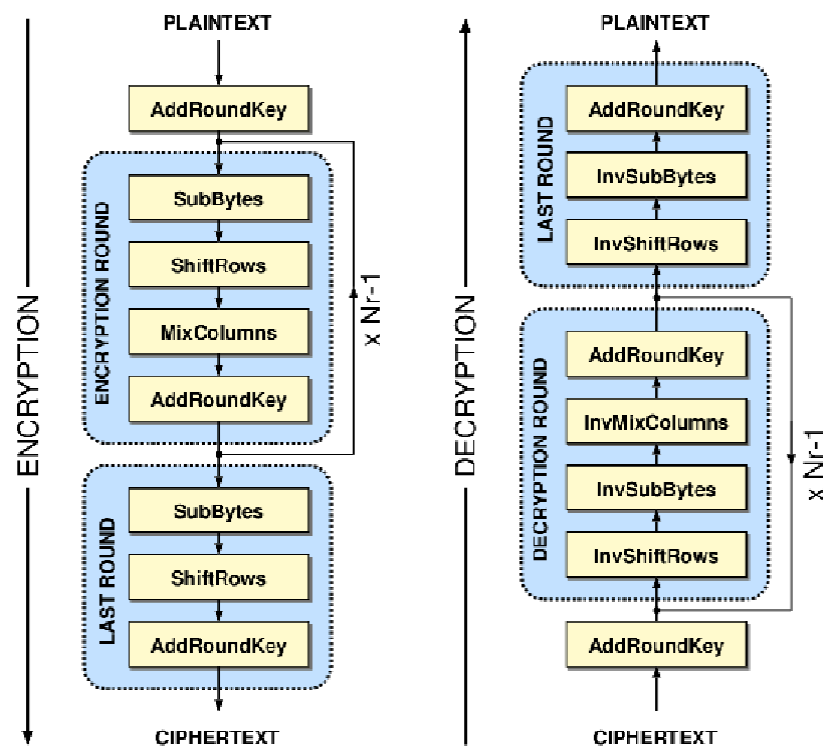


FIGURE 3.2.1.1 AES ALGORITHM

Need for AES

The Advanced Encryption Standard (AES) is a widely used symmetric encryption algorithm, essential for ensuring the confidentiality and integrity of data transmitted

over networks or stored in systems. The adoption of the AES algorithm within medical data security frameworks is imperative for ensuring the confidentiality, integrity, and availability of patient information. By leveraging AES encryption, healthcare organizations can effectively mitigate the risk of unauthorized access, prevent data breaches, achieve regulatory compliance, and foster trust among patients, ultimately safeguarding the sanctity of medical data in an increasingly digitized healthcare landscape. Medical records are a treasure trove of sensitive information, ranging from personal details to intricate medical histories, treatments, and prescriptions. Safeguarding this data against unauthorized access is not only essential for patient confidentiality but also a legal imperative under healthcare privacy regulations such as HIPAA. Utilizing AES encryption ensures that patient data remains strictly confidential, inaccessible to unauthorized parties, and compliant with regulatory standards. Unauthorized access to medical records poses grave risks, including identity theft, fraudulent activities, and even medical malpractice. Implementing AES encryption fortifies medical data against such intrusions, effectively thwarting cyber attackers and unauthorized individuals from exploiting sensitive patient information. With AES encryption in place, only authorized users possessing the requisite decryption keys can access the data, providing a robust defense mechanism. Healthcare data breaches have become alarmingly common, with cybercriminals increasingly targeting healthcare organizations. The repercussions of such breaches, including financial losses, reputational damage, and legal liabilities, can be devastating. AES encryption serves as a formidable shield against data breaches by securing medical data both at rest and in transit. Even in the event of a breach, encrypted data remains indecipherable without the encryption keys, significantly minimizing the impact of the breach. Patients entrust healthcare providers with their most sensitive information, expecting it to be safeguarded with the utmost care. Integrating AES encryption into medical data security frameworks not only fulfills this expectation but also instills confidence in patients regarding the security of their information. Enhanced data security measures, including encryption, contribute to a positive patient experience, reinforcing trust in healthcare providers and strengthening organizational reputation. The AES algorithm is essential in medical data security frameworks to ensure the confidentiality, integrity, and privacy of patient information. By encrypting medical data with AES, healthcare organizations can prevent unauthorized access, protect against data breaches, comply

with regulatory requirements, and build trust among patients, ultimately enhancing the security and confidentiality of medical data.

Advantages of using AES Algorithm

- **Strong security**

AES is a widely recognized and trusted encryption standard that provides robust security for sensitive data, including medical records. It employs symmetric key encryption, making it highly secure against unauthorized access and decryption without the correct key.

- **Efficiency**

AES encryption and decryption processes are highly efficient, allowing for fast and seamless encryption and decryption of medical data. This efficiency is essential in healthcare settings where quick access to patient information is critical for providing timely and effective care.

- **Versatility**

AES supports different key lengths providing flexibility to tailor encryption strength based on specific security requirements and regulatory compliance standards. This versatility ensures that healthcare organizations can implement AES encryption for their security needs.

- **Scalability**

AES encryption is scalable, allowing healthcare organizations to encrypt large volumes of medical data without sacrificing performance or security. Whether encrypting individual patient records or entire databases, AES can efficiently handle varying data sizes and encryption requirements.

- **Compatibility**

AES encryption is widely supported by modern computing systems, devices, and software applications, ensuring seamless integration into existing medical data management systems and workflows. This compatibility simplifies the implementation and adoption of AES encryption within healthcare organizations.

- **Protection Against Data Breaches**

AES encryption provides a critical layer of defense against data breaches by rendering encrypted medical data unreadable to unauthorized individuals or

cyber attackers. Even if attackers gain access to encrypted data, they cannot decrypt it without the encryption key, reducing the risk and impact of data breaches.

- **Patient Trust and Confidence**

Implementing AES encryption demonstrates a commitment to protecting patient privacy and confidentiality. By safeguarding medical data with AES encryption, healthcare organizations can build trust and confidence among patients, enhancing their reputation and fostering positive patient-provider relationships.

3.3 Key requests and receiving files

On the doctor's interface, a list of encrypted patient files is displayed. If a doctor wishes to access and review a specific patient's information for treatment purposes, they can initiate a key request. Upon sending the key request, it is processed successfully, triggering an immediate notification to the patient containing the doctor's credentials. Once the patient verifies that the request originates from the desired specialist, they can approve and transmit the encryption key. Subsequently, the encryption key is delivered to the doctor via email, allowing them to copy it for subsequent decryption. Access to the patient's file is granted only upon providing the correct encryption key. Any attempt with an incorrect key results in denied access.

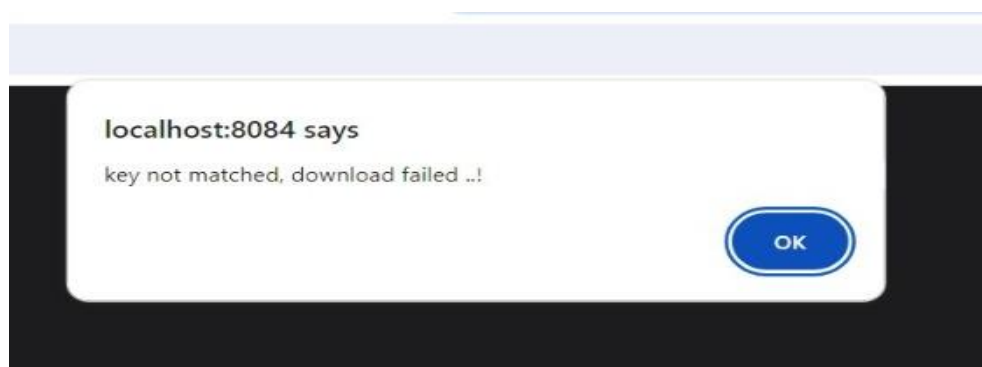


FIGURE 3.3.1 KEY NOT MATCHED

With the correct key in hand, the doctor can proceed to download and review the patient's medical information, enabling them to accurately diagnose and address the patient's concerns. This secure exchange of information between the patient and

doctor ensures that communication is established effectively while maintaining the confidentiality and integrity of the medical data throughout the process.

3.4 Prescriptions

After successfully reviewing and diagnosing the patient's concerns, establishing secure and effective communication between both parties is crucial. One-way communication is insufficient for addressing medical issues comprehensively; therefore, it's essential for the doctor to be able to respond to the patient's concerns by providing a prescription. To facilitate this, a feature called "share prescription" is available, enabling the doctor to compose a prescription outlining medications and further instructions for managing the condition. Upon sending the prescription, the system confirms its successful delivery. On the patient's end, upon receiving the prescription, they can directly access and view it. This ensures that patients promptly receive treatment guidance and can adhere to the prescribed medications and instructions as advised by their healthcare provider.

Throughout this process, all relevant information, including patient and doctor details, uploaded files, and prescriptions, are securely stored in the database. This ensures comprehensive documentation of the patient's medical history and interactions with healthcare providers, promoting continuity of care and ensuring data integrity for future reference.

| id | username | password | email | dob | phoneno | status |
|--------|----------|-----------|--------------------------|------------|------------|--------|
| 8 | Sowmya | 123456 | sowp8029@gmail.com | 2021-10-20 | 8019746153 | NO |
| 9 | Shruthi | sutti@19 | 20egl05419@anurag.edu.in | 2001-12-29 | 8074659627 | NO |
| 10 | Bhanu | 123456 | bhanu@gmail.com | 2000-01-16 | 6309475210 | NO |
| 11 | Kiran | kiran@29 | kiran@gmail.com | 2000-06-29 | 6300067632 | NO |
| 12 | Shivam | shivam@5 | shivam@gmail.com | 2000-07-05 | 7240340022 | NO |
| 13 | Bhavya | bhavya123 | bhavya@gmail.com | 1990-01-05 | 9876546103 | NO |
| 14 | Megha | megha25 | 20egl05425@gmail.com | 2024-03-01 | 7890765432 | NO |
| (NULL) | (NULL) | (NULL) | (NULL) | (NULL) | (NULL) | (NULL) |

| id | pid | pname | filename | data | doctor | skey | patt |
|--------|--------|--------------------------|------------|---------|--------|--------------------------|------------------------------------|
| 12 | 5441 | sowp8029@gmail.com | phyl.txt | waiting | 7 b | svnkutti@gmail.com | C9XjuDoid9S93kpHH/eAsA== Physician |
| 13 | 5419 | 20egl05419@anurag.edu.in | skin1.txt | waiting | 7 b | 20egl05434@anurag.edu.in | k3RDcyZ3xsnBLzCTOr+7rg== Skin |
| 14 | 5404 | bhanu@gmail.com | heart1.txt | waiting | 7 b | 20egl05418@anurag.edu.in | ghjeF/OE09RZJKuPX3vr6Q== HJeart |
| 15 | 5441 | sowp8029@gmail.com | phyl.txt | waiting | 7 b | svnkutti@gmail.com | C9XjuDoid9S93kpHH/eAsA== Physician |
| 16 | 5160 | kiran@gmail.com | heart2.txt | waiting | 7 b | 20egl05418@anurag.edu.in | 5P2Q3+qUMKUXENalk9blaA== HJeart |
| 17 | 5472 | shivam@gmail.com | skin2.txt | waiting | 7 b | 20egl05434@anurag.edu.in | 6hmG5oFmW9bddd6pA32dpQ== Skin |
| 18 | 54410 | bhavya@gmail.com | skin1.txt | waiting | 7 b | 20egl05434@anurag.edu.in | ksg0sKyt7bjMnPggo8VU1Q== Skin |
| 19 | 54425 | 20egl05425@gmail.com | phy2.txt | waiting | 7 b | svnkutti@gmail.com | 5jPRDeAVVkgCWbE/3DqTZg== Physician |
| (NULL) | (NULL) | (NULL) | (NULL) | (NULL) | 0 Kb | (NULL) | (NULL) |

FIGURE 3.4.1 DATABASE EXAMPLES

3.5 Overview

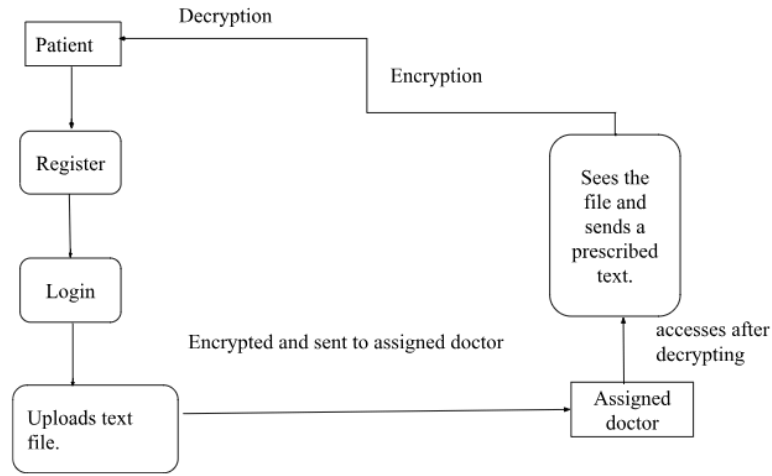


FIGURE 3.5.1 CONCEPT OVERVIEW

The figure 3.5.1 illustrates the concept of the proposed method, describing the roles and functions of patients and doctors in a brief way. A proposed attribute-based encryption (ABE) technique is introduced, wherein patient data is encrypted into text files using the AES encryption method. Access permissions are then assigned to specific doctors to restrict access to other users. This ensures that only authorized individuals, such as the respective doctor, can access the encrypted patient data. Under this system, users are able to view encrypted patient data and can send requests to the data owner to obtain a security key for downloading and viewing the data. However, access to the security key is contingent upon the satisfaction of the access control policy. If a user does not meet the specified criteria outlined in the access control policy, they will be denied access to the security key, thus preventing them from downloading the patient data.

For an elaborated view of the proposed method, figure 3.5.2 can be referenced.

In this scenario, we have several components:

- F1 represents the uploaded file.
- E1 and D1 denote the encryption and decryption functions, respectively.
- ABE1 signifies attribute-based encryption, a method used for securing data.
- R1 refers to the request sent to obtain the encryption key.

- K1 represents the encryption key required to access the encrypted data.
- RE1 is the response to the request for the encryption key.
- T1 and T2 indicate the time taken for encryption and decryption processes.

These components form a system where files are uploaded and secured using attribute-based encryption. Encryption and decryption functions are applied, and users can request access to encrypted data by sending a key request (R1), receiving a response (RE1), and utilizing the encryption key (K1) to decrypt the data. Additionally, the time taken for encryption (T1) and decryption (T2) processes is measured to evaluate system performance.

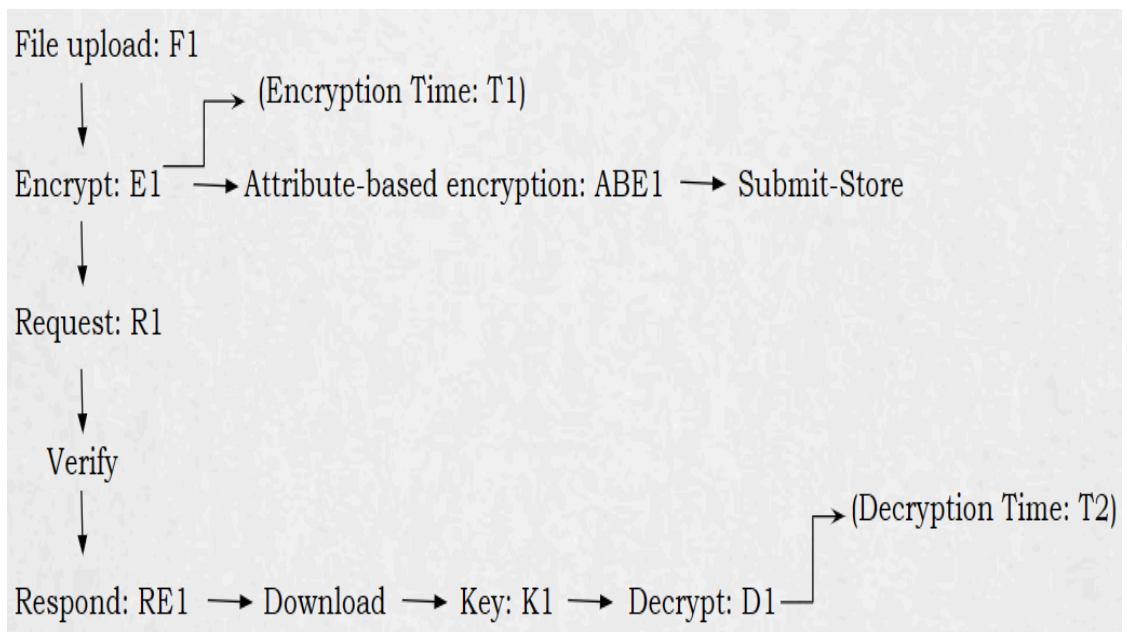


FIGURE 3.5.2 ILLUSTRATION

4. Implementation

Input: Patient text files.

Output: Doctor's prescription .

4.1. FUNCTIONALITIES

4.1.1. Encryption and Decryption

- To safeguard sensitive patient data, we will implement AES encryption and decryption algorithms. These algorithms will ensure that patient information remains secure while permitting authorized healthcare professionals to decrypt and access the data as needed.
- The encryption process will utilize AES encryption to transform patient data into an unreadable format, thus protecting it from unauthorized access. Authorized healthcare professionals will possess the decryption keys required to decrypt the data, allowing them to access and review patient information when necessary.
- By implementing AES encryption, we ensure that patient data is adequately protected from unauthorized access while maintaining accessibility for authorized personnel. This approach ensures the confidentiality and integrity of sensitive medical information, aligning with data security standards and regulations in the healthcare industry.

4.1.2. Key Management

- Establishing a robust key management system is paramount to securely handle the generation, storage, and distribution of encryption keys. Effective key management is essential for upholding the confidentiality and integrity of encrypted data.
- To achieve this, we will implement procedures for key generation, ensuring that keys are generated using strong cryptographic algorithms and randomization techniques. These keys will then be securely stored in a dedicated key repository, employing encryption and access controls to prevent unauthorized access.

- The distribution of encryption keys will be carefully managed, utilizing secure channels and protocols to transmit keys only to authorized entities. Key distribution mechanisms will be designed to minimize the risk of interception or tampering during transit.
- Regular key rotation and updates will also be implemented to mitigate the risk of key compromise and ensure ongoing security. Additionally, comprehensive auditing and logging mechanisms will be in place to monitor key usage and detect any suspicious activities.
- By developing a robust key management system, we can effectively safeguard the confidentiality and integrity of encrypted data, thereby maintaining compliance with regulatory requirements and instilling trust in the security of our systems.

4.1.3. Access Control

- We will implement access control mechanisms to carefully regulate access to patient data based on roles and permissions. This approach guarantees that only authorized healthcare professionals can access decrypted patient information, while patients retain control over their own data.
- To achieve this, we will establish role-based access control (RBAC) policies, assigning specific roles to healthcare professionals based on their responsibilities and privileges within the system. Each role will be associated with a set of permissions that determine the actions they can perform and the data they can access.
- Patients will also have a role in controlling access to their data. They will be granted permissions to manage their own data, including the ability to grant or revoke access to healthcare professionals as they see fit. This empowers patients to maintain control over who can access their sensitive medical information.
- Access control mechanisms will be enforced through authentication and authorization processes. Healthcare professionals will need to authenticate themselves using secure credentials before accessing patient data. Authorization checks will then be conducted to ensure that they have the necessary permissions to view specific data.

- By implementing these access control mechanisms, we ensure that patient data is protected against unauthorized access while allowing authorized healthcare professionals to fulfill their duties effectively. Patients retain ownership and control over their data, promoting transparency and trust in the handling of sensitive medical information.

4.1.4. Prescription Upload

- Doctors will be enabled to securely upload prescriptions in text file format, implementing robust mechanisms to authenticate doctors and verify the integrity of uploaded prescription files. These measures are essential for preventing unauthorized access or tampering with sensitive medical information.
- To achieve this, we will establish a secure authentication process for doctors, requiring them to authenticate using strong credentials such as username and password. This ensures that only authorized doctors can access the prescription upload functionality.
- Furthermore, we will implement measures to verify the integrity of uploaded prescription files.
- Access control mechanisms will be enforced to ensure that only authenticated doctors have permission to upload prescription files. Role-based access control (RBAC) policies can be employed to restrict access to the prescription upload feature based on the doctor's role and privileges within the system.
- By implementing these measures, we ensure that only authenticated doctors can securely upload prescriptions, and that the integrity of uploaded files is verified to prevent unauthorized access or tampering. This enhances the overall security and integrity of the prescription upload process, safeguarding sensitive medical information against unauthorized disclosure or manipulation.

4.1.5. Encryption Time Calculation

- We will implement functionality to measure the time required for encryption processes, which plays a crucial role in evaluating the efficiency of the encryption algorithm and overall system performance. When developing this functionality, we will consider various factors such as file size and system

resources to ensure accurate measurement of encryption time.

- The development of functionality to measure encryption time is essential for evaluating the efficiency of the encryption algorithm and assessing system performance. By considering factors such as file size and system resources, we can ensure accurate measurement of encryption time and make informed decisions to optimize system efficiency.

4.1.6. Secure Communication

- Secure communication channels will be established to facilitate the transmission of encrypted data between patients and healthcare providers.
- We will authenticate the identity of both parties involved in the communication process. This ensures that data is transmitted securely between trusted entities, reducing the risk of man-in-the-middle attacks or unauthorized access to patient data.
- We will implement robust encryption algorithms and key management practices to ensure the confidentiality and integrity of transmitted data. This includes using strong encryption keys, rotating keys regularly, and securely managing cryptographic materials to prevent unauthorized access or tampering with encrypted data.
- By implementing these measures, we can establish secure communication channels for transmitting encrypted data between patients and healthcare providers. This helps to protect the privacy and confidentiality of patient information, ensuring compliance with regulatory requirements and instilling trust in the security of healthcare communication systems.

4.1.7. Logging and Auditing

- Integrating logging and auditing functionalities into our system will meticulously track access to patient data and encryption/decryption activities. These measures will enable us to maintain detailed logs, which are essential

for forensic analysis and ensuring compliance with regulatory requirements.

- To accomplish this, we will implement logging mechanisms that capture and record all relevant events related to access to patient data and encryption/decryption activities. This includes logging details such as the user or system entity involved, the type of action performed (e.g., data access, encryption, decryption), and any pertinent metadata.
- We will ensure that the logged information is stored securely and is tamper-evident to maintain the integrity of the audit trail.
- By implementing logging and auditing functionalities, we can effectively track access to patient data and encryption/decryption activities, maintain detailed records for forensic analysis, and demonstrate compliance with regulatory standards. This enhances the overall security and accountability of our system, providing assurance that patient data is handled responsibly and in accordance with privacy regulations.

4.1.8. User Interface

- Designing a user-friendly interface that facilitates secure interaction between patients and healthcare professionals while considering usability principles and accessibility requirements to ensure ease of use for all users.
- The interface will feature intuitive navigation and layout, allowing users to easily access and navigate through different functionalities of the system. Clear and concise labels, buttons, and instructions will be provided to guide users through the interface, minimizing the need for extensive training or technical knowledge..
- For patients, the interface will provide a streamlined process for accessing their medical information, uploading documents, and communicating with healthcare professionals. Patients will have the ability to securely view and manage their data, control access permissions, and communicate their concerns effectively.

- For healthcare professionals, the interface will offer features for securely accessing patient data, uploading prescriptions, and communicating with patients. It will provide tools for efficiently managing patient records, reviewing diagnostic information, and securely exchanging messages with patients.
- The interface will prioritize user experience and accessibility, ensuring that both patients and healthcare professionals can interact with the system securely and efficiently.

4.1.9 Secure Storage

- We will ensure that encrypted patient data is securely stored in a protected environment, employing measures such as a secure database or encrypted file system. These safeguards will prevent unauthorized access to stored data and maintain the confidentiality and integrity of patient information.
- To achieve this, we will implement encryption of data at rest, ensuring that patient data remains encrypted when stored in the database or file system. This means that even if unauthorized individuals gain access to the storage environment, they will be unable to view or decipher the encrypted data without the appropriate decryption keys.
- Enforcing access controls at the storage level to restrict access to patient data based on role-based permissions and authentication mechanisms. This includes implementing granular access controls that limit access to sensitive data only to authorized personnel with the necessary privileges.
- By implementing these measures, we can ensure that encrypted patient data is securely stored in a protected environment, safeguarding it from unauthorized access or breaches. This not only helps to maintain compliance with regulatory requirements but also instills trust and confidence in the security of the healthcare system among patients and stakeholders.

4.2. Attributes

- **Confidentiality:** This attribute emphasizes the importance of preserving the confidentiality of sensitive medical information, safeguarding it against unauthorized access or disclosure. It entails the encryption of patient data to prevent unauthorized parties from viewing or accessing the information. To uphold confidentiality, robust encryption algorithms are utilized alongside secure key management practices. By encrypting patient data, sensitive medical information is transformed into ciphertext that can only be deciphered with the appropriate decryption key. This ensures that even if unauthorized individuals gain access to the data, they are unable to interpret its contents without the corresponding decryption key.
- **Integrity:** Integrity serves as a cornerstone in ensuring the accuracy and reliability of patient data across its entire lifecycle. This involves safeguarding against unauthorized alterations, deletions, or tampering with patient records, thereby preserving the trustworthiness of the information. Techniques such as cryptographic hashing and digital signatures play a pivotal role in verifying data integrity and detecting any unauthorized modifications to the data.
- **Availability:** Availability is crucial for ensuring that patient data remains accessible to authorized users whenever required, without any interruption or downtime. This attribute entails the implementation of robust infrastructure, redundancy measures, and comprehensive disaster recovery plans to minimize service disruptions and ensure continuous access to medical information.
- **Authentication:** Authentication plays a crucial role in verifying the identity of users and guaranteeing that only authorized individuals can access encrypted patient data. This process entails the implementation of robust user authentication mechanisms, such as passwords, biometrics, or multi-factor authentication (MFA), to validate the identity of users before granting access to sensitive medical information. User authentication mechanisms serve as a critical barrier against unauthorized access to patient data.

- **Authorization:** Authorization plays a pivotal role in determining the privileges and permissions assigned to users according to their roles and responsibilities within the healthcare system. Effective authorization mechanisms are essential for maintaining the confidentiality, integrity, and availability of patient data. By defining access control policies based on role-based access control (RBAC) principles, we can ensure that users are granted access to the minimum amount of data necessary to perform their duties effectively. This helps minimize the risk of unauthorized access to sensitive information and mitigates the potential for data breaches or misuse.

- **Encryption strength:** It is a critical factor that gauges the resilience of both the encryption algorithm and the key management practices employed to safeguard patient data. This encompasses the careful selection of robust encryption algorithms, such as the Advanced Encryption Standard (AES), and the implementation of effective key management practices to maintain the confidentiality of encryption keys. Ensuring the encryption strength of patient data involves utilizing encryption algorithms that offer high levels of security, such as AES, which is widely recognized for its strength and reliability. By employing AES or similarly robust encryption algorithms, we can fortify the protection of patient data against unauthorized access and decryption attempts.

- **Performance:** Performance is a crucial aspect that evaluates the efficiency and speed of encryption and decryption processes within the system. This encompasses the optimization of encryption algorithms and key management practices to minimize encryption time and system resource utilization, all while maintaining sufficient security levels. Efficient performance in encryption and decryption processes is essential to ensure timely access to patient data without compromising security. Algorithms like AES are preferred for their strong encryption capabilities and efficient processing speeds. Effective key management practices also contribute to performance optimization by ensuring that encryption keys are generated, stored, and distributed efficiently.

4.3. Experimental Screenshots

The screenshot shows the 'Doctor Registration' page of the 'Enhanced Medical Data Security Framework'. The page has a dark blue header with the title and navigation links: 'Home', 'Patient', and 'Doctor' (highlighted in orange). Below the header is a large, faded image of a person. The registration form is centered and includes the following fields: 'Name' (text input), 'Email' (text input with placeholder 'Email Address'), 'Password' (text input), 'DOB' (date picker with format 'dd-mm-yyyy'), 'ContactNO' (text input with placeholder 'Contactno'), and 'Attribute' (dropdown menu with 'Select' as the current value). At the bottom of the form are two buttons: 'Register' and 'Reset'.


FIGURE 4.3.1 DOCTOR REGISTRATION PAGE

The screenshot shows the 'Doctor Login' page of the 'Enhanced Medical Data Security Framework'. The page has a dark blue header with the title and navigation links: 'Home', 'Patient', and 'Doctor' (highlighted in orange). Below the header is a large, faded image of a person. The login form is centered and includes the following fields: 'E-Mail' (text input with placeholder 'Enter E-Mail') and 'Password' (text input with placeholder 'Password'). At the bottom of the form are two buttons: 'Submit' and 'Reset'. Below the buttons is a link: 'New User [Click Here](#) To Register'.

FIGURE 4.3.2 DOCTOR LOGIN PAGE

Enhanced Medical Data Security Framework

Home Patient Doctor



Patient Registration

Name:

Password:

Email:


DOB:

ContactNO:

FIGURE 4.3.3 PATIENT REGISTRATION PAGE

Enhanced Medical Data Security Framework

Home Patient Doctor



Patient Login

E-Mail:

Password:

New User [Click Here](#) To Register

FIGURE 4.3.4 PATIENT LOGIN PAGE

Upload Patient Data

Patient ID:

Attribute:

File:

Doctor, I've been experiencing persistent abdominal pain and discomfort for the past few weeks. The pain is localized to the upper right side of my abdomen and often radiates to my back and shoulder. It's a dull, constant ache that sometimes intensifies after eating fatty or greasy foods. Additionally, I've noticed that my appetite has decreased recently, and I've been feeling nauseous, especially in the mornings.

I haven't had any significant changes in my diet or lifestyle that could explain these symptoms. I try to eat healthily and exercise regularly, but the abdominal pain is making it difficult for me to enjoy meals or engage in physical activity. I don't have a history of gastrointestinal issues, but I'm concerned that these symptoms could be indicative of a more serious underlying condition.

I've tried over-the-counter antacids and pain relievers to alleviate the discomfort, but they only provide temporary relief. The pain persists, and it's starting to affect my daily life and productivity. I don't have any allergies or sensitivities to food, and I haven't experienced any recent trauma or injury to the abdomen.

Doctor, what could be causing these symptoms? Could it be related to a digestive disorder or a problem with my gallbladder or liver? Are there any tests you recommend to help diagnose the issue? Also, what steps can I take to manage the abdominal pain and improve my overall digestive health?

FIGURE 4.3.5 PATIENT FILE UPLOAD

View Data & Encrypt the Data

| Patient Id | Filename | Data | Attribute | Key Gen |
|------------|----------|--|-----------|---------|
| 54425 | phy2.txt | <p>Doctor, I've been experiencing persistent abdominal pain and discomfort for the past few weeks. The pain is localized to the upper right side of my abdomen and often radiates to my back and shoulder. It's a dull, constant ache that sometimes intensifies after eating fatty or greasy foods. Additionally, I've noticed that my appetite has decreased recently, and I've been feeling nauseous, especially in the mornings.</p> <p>I haven't had any significant changes in my diet or lifestyle that could explain these symptoms. I try to eat healthily and exercise regularly, but the abdominal pain is making it difficult for me to enjoy meals or engage in physical activity. I don't have a history of gastrointestinal issues, but I'm concerned that these symptoms could be indicative of a more serious underlying condition.</p> <p>I've tried over-the-counter antacids and pain relievers to alleviate the discomfort, but they only provide temporary relief. The pain persists, and it's starting to affect my daily life and productivity. I don't have any allergies or sensitivities to food, and I haven't experienced any recent trauma or injury to the abdomen.</p> <p>Doctor, what could be causing these symptoms? Could it be related to a digestive disorder or a problem with my gallbladder or liver? Are there any tests you recommend to help diagnose the issue? Also, what steps can I take to manage the abdominal pain and improve my overall digestive health?</p> | Physician | click |

FIGURE 4.3.6 PATIENT SIDE- VIEW DATA

View Keys

| Patient Id | Filename | Data | Cipher Data | SKey | Encryption Time |
|------------|----------|---|--|---------------------------|-----------------|
| 54425 | phy2.txt | <p>Doctor, I've been experiencing persistent abdominal pain and discomfort for the past few weeks. The pain is localized to the upper right side of my abdomen and often radiates to my back and shoulder. It's a dull, constant ache that sometimes intensifies after eating fatty or greasy foods. Additionally, I've noticed that my appetite has decreased recently, and I've</p> | <p>Y3AaH6//3Xp1l14GpXEkS44SBPr1rMz6w6uXZ6YH4j9KbPU05jB2sQ78tocvjwCO+E5Ko/dq3hLSaDLo0szvKyUu8fL6NTEAkY0eVKSnmfPxCPiIISy+G0PB6K7UJ7IVgXLjch7FqosVQJgMbRKrED47Lo0u7PWmczMYG0B30V9MAwJjLni+rrnnlsui8NS23PxhwQ5etqP6sYMRnbZTKswxSozQyqC8RsemudPusQmC0158UZDq1aRhwQaLEISvu/gQ9N1rvCW0HCct9rD/nrcJEJgLa9hr6KHZ6VCg+ACVfVrRXjaLX38kehBEaEPA0CCF4bgFv6tQ7m3mj0etkH++6p58eby</p> | 5jPRDeAVVkgCwBfE/3DqTzG== | 39 |

FIGURE 4.3.7 VIEW KEYS

| View Data & Send request | | | | |
|--------------------------|--------------------------|------------|--|--------------|
| Patient Id | Patient Name | Filename | Data | Send request |
| 5441 | sowp8029@gmail.com | phy1.txt | YY6bJI4LPT0C+JQRpomb qEu/t1SnRFokfoLrM1JS | click |
| 5419 | 20eg105419@anurag.edu.in | skin1.txt | HrFdp6c7xNKTEqdL8qnE gUZMwtp3DLewzsI19HjD | click |
| 5404 | bhanu@gmail.com | heart1.txt | 50sRG7jP0i+Eqb/KoKHx TWHwiIdwIevv52UjttWE | click |
| 5160 | kiran@gmail.com | heart2.txt | pUU6fVa6aakgNx/iGXj9 K1tp22axwTHnFkVB0hnF | click |
| 5472 | shivam@gmail.com | skin2.txt | A2wLbQRKUaz6I7ZkvEeu /QXhwgdh3pH0n2uhhkYp | click |
| 54410 | bhavya@gmail.com | skin1.txt | NreApSuJvv1B5jacYZXE 0PW4LgKPWUFQTMDeGfp9 | click |
| 54425 | 20eg105425@gmail.com | phy2.txt | YJAaH6//3Xp1i14GpXEk S44SBPr1rMz6w6uXZ6YH | click |

FIGURE 4.3.8 DOCTOR SIDE-SEND REQUEST

| View Doctor's Request & Send Skey | | | | |
|-----------------------------------|------------|----------|------------------|-----------------------------|
| Doctor | Patient Id | Filename | Doctor Attribute | Verify Attribute & Send Key |
| svnkkutti@gmail.com | 54425 | phy2.txt | Physician | click |

FIGURE 4.3.9 PATIENT SIDE- SEND KEY

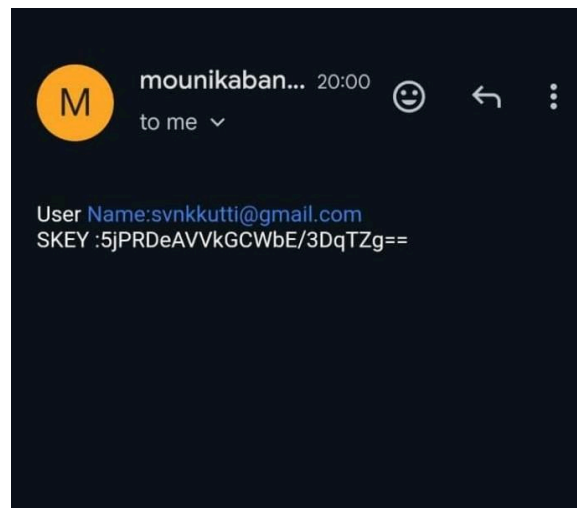



FIGURE 4.3.10 MAIL

Enhanced Medical Data Security Framework

Home View Data **Download Data** Prescription Logout



Download

Secure Data Transmission In Medical Things

Patient ID

Filename

Skey

Download

FIGURE 4.3.11 FILE DOWNLOAD

Share Prescription

Secure Data Transmission In Medical Things

Patient ID

Patient Name

Prescription

For abdominal pain and discomfort associated with digestive issues such as gallbladder inflammation (cholecystitis) or gallstones, healthcare providers may prescribe pain relievers, antacids, proton pump inhibitors (PPIs), or antibiotics, depending on the underlying cause.

One specific medication commonly used for gallstone-related symptoms is ursodeoxycholic acid (UDCA). UDCA is a medication that can help dissolve small cholesterol gallstones and reduce the severity of symptoms. It's often prescribed in cases where surgery to remove the gallbladder (cholecystectomy) is not immediately necessary or desired.

Share

FIGURE 4.3.12 PRESCRIPTION

| View Doctor's Prescription | | | |
|----------------------------|------------|----------------------|---|
| Doctor Name | Patient Id | Patient Name | Doctor's Prescription |
| svnkkutti@gmail.com | 54425 | 20eg105425@gmail.com | <p>For abdominal pain and discomfort associated with digestive issues such as gallbladder inflammation (cholecystitis) or gallstones, healthcare providers may prescribe pain relievers, antacids, proton pump inhibitors (PPIs), or antibiotics, depending on the underlying cause.</p> <p>One specific medication commonly used for gallstone-related symptoms is <u>ursodeoxycholic acid (UDCA)</u>. UDCA is a medication that can help dissolve small cholesterol gallstones and reduce the severity of symptoms. It's often prescribed in cases where surgery to remove the gallbladder (cholecystectomy) is not immediately necessary or desired.</p> |

FIGURE 4.3.13 PRESCRIPTION VIEW

5. Experimental Setup

5.1 NetBeans

IDE Selection and Installation

- Selecting NetBeans as the Integrated Development Environment (IDE) for Java development is a strategic choice owing to its robust features and seamless compatibility with web applications.
- To begin, navigate to the official NetBeans website at <https://netbeans.apache.org/> and proceed to download the software. Follow the installation instructions provided for your specific operating system to ensure a smooth installation process.
- By opting for NetBeans, developers can leverage its comprehensive set of tools and functionalities tailored for Java development, facilitating the creation of high-quality web applications with ease and efficiency.

Project Creation in NetBeans

- To commence the project in NetBeans, initiate the application and proceed to create a new Java web project by following these steps:
 - Step 1: Navigate to the "File" menu.
 - Step 2: Choose "New Project" from the dropdown list.
 - Step 3: From the project types, select "Java Web."
 - Step 4: Specify a name for your project, designate the project location, and then click "Next."
 - Step 5: Next, opt for "Java EE Web" as the server and "Java Server Faces" as the framework.
 - Step 6: Once configured, click on "Finish" to initiate the project setup process.
- By adhering to these steps, you'll establish a new Java web project within NetBeans, primed for the development of your medical application.

5.2 BackEnd Development(Java)

For the backend development of the application using Java within the NetBeans

project, commence by implementing the following steps:

- **Coding Backend Logic:** Initiate the coding process within the NetBeans project to define the backend logic of the application. This involves creating classes and methods responsible for handling various functionalities such as data processing, encryption, decryption, and database operations.
- **Define Classes and Methods:** Organize the backend logic by defining appropriate classes and methods to encapsulate different aspects of functionality. These classes and methods will facilitate data manipulation, encryption/decryption processes, and interactions with the database.
- **Implement AES Encryption and Decryption:** Utilize the Java Cryptography Architecture (JCA) to integrate AES encryption and decryption functionalities into the application. Implement methods to securely encrypt sensitive data before storage and decrypt it when required for processing or display.
- **Database Operations with MySQL:** Establish connections to the MySQL database using JDBC (Java Database Connectivity) to facilitate CRUD (Create, Read, Update, Delete) operations. Implement methods to interact with the database, including inserting new records, retrieving data, updating existing entries, and deleting records as necessary.

By following these steps and leveraging the capabilities of Java within the NetBeans environment, you can develop a robust backend for your medical application, ensuring efficient data processing, secure encryption, and seamless database operations.

5.3 FrontEnd Development(HTML, CSS, JavaScript)

For the frontend development of the web application using HTML, CSS, and JavaScript within the NetBeans project, follow these steps:

- ➔ **Creating User Interface:** Begin by crafting the user interface (UI) for the web application using HTML. Design HTML web pages to accommodate various functionalities, including user login, prescription upload, data display, and any other relevant features.
- ➔ **Designing Web Pages:** Develop HTML web pages that align with the different functionalities of the application. Each page should be intuitively designed to provide users with a seamless experience. Consider the user flow and ensure that the layout is clear and easy to navigate.
- ➔ **Styling with CSS:** Utilize CSS to style the HTML web pages and enhance the visual appeal of the application. Apply CSS rules to customize the appearance of elements, including fonts, colors, spacing, and layout. Ensure consistency across pages to maintain a cohesive design.
- ➔ **Implementing JavaScript:** Enhance the functionality of the web pages by implementing JavaScript code. Use JavaScript to add dynamic behavior, such as form validation to ensure data accuracy, and interactive features to improve user engagement. Implement event listeners and DOM manipulation techniques to make the UI more responsive and interactive.

By following these steps and leveraging the capabilities of HTML, CSS, and JavaScript within the NetBeans environment, you can create a compelling frontend for your web application, providing users with an intuitive and visually appealing interface.

5.4 Database Setup (SQLyog)

To set up the database for your application using SQLyog, follow these steps:

- ➔ **Install SQLyog:** Begin by installing SQLyog, a MySQL database management tool, on your system. You can download SQLyog from the official website and follow the installation instructions provided.
- ➔ **Create MySQL Database:** Once SQLyog is installed, launch the application and create a new MySQL database to store patient data, prescriptions, and other relevant information. Define the database schema by specifying tables, columns, and relationships according to the requirements of your application.

- **Define Database Schema:** Design the database schema by creating tables to represent different entities in your application, such as patients, prescriptions, doctors, etc. Define appropriate columns for each table to store relevant data, and establish relationships between tables using foreign keys where necessary.
- **Establish Connection with Java Application:** After defining the database schema, configure JDBC drivers in your Java application to establish a connection with the MySQL database. Ensure that you provide the correct connection properties, including the database URL, username, password, and other required parameters.

By following these steps and leveraging the capabilities of SQLyog, you can set up a MySQL database for your application and establish a seamless connection between the Java application and the database, enabling efficient storage and retrieval of data.

5.5 Integration and Testing

To integrate the backend Java code with the frontend HTML, CSS, and JavaScript files and create a cohesive web application, follow these steps:

- **Integration of Backend and Frontend:** Merge the backend Java code, responsible for data processing, encryption, decryption, and database operations, with the frontend HTML, CSS, and JavaScript files. Ensure that the backend functionality is seamlessly integrated with the frontend UI to provide a cohesive user experience.
- **Testing Functionality:** Test the functionality of the integrated application by running it locally within NetBeans or deploying it to a local server environment. Verify that all components, including user authentication, data encryption, database operations, and UI interactions, work as expected.
- **Unit Testing:** Perform unit testing to validate individual components of the application, such as Java classes and JavaScript functions. Test each component in isolation to ensure that it performs its intended function accurately.
- **Integration Testing:** Conduct integration testing to verify the seamless interaction between backend and frontend components. Test the integration points to ensure that data is transferred correctly between the frontend and

backend, and that all interactions work as intended.

- **User Acceptance Testing:** Lastly, conduct user acceptance testing to evaluate the application's usability and functionality from the end user's perspective. Involve stakeholders, such as healthcare professionals and patients, to test the application and provide feedback on its usability, features, and overall user experience.

By following these steps and performing thorough testing at each stage, you can ensure that the integrated web application functions reliably, meets user requirements, and provides a seamless experience for both healthcare professionals and patients.

5.6 Deployment

To deploy the web application to a production environment, follow these steps:

- **Testing Completion:** Ensure that testing is complete, and the application is stable and free of any critical issues. Thoroughly validate all functionality, including data processing, encryption, decryption, database operations, and user interface interactions.
- **Select Hosting Environment:** Choose a suitable hosting environment for deploying the web application. This may include a web server for hosting the frontend files, a database platform for storing data, and any other necessary infrastructure components.
- **Package Application:** Package the web application into a deployable format, such as a WAR (Web ARchive) file for Java-based applications. Ensure that all dependencies, libraries, and configuration files are included in the package.
- **Deploy to Production:** Deploy the packaged application to the selected hosting environment. Follow the deployment instructions provided by the hosting platform, and ensure that all configurations are properly set up, including database connections, security settings, and environment variables.
- **Test Deployment:** Once the application is deployed, perform testing in the production environment to verify that it functions correctly and performs as expected. Test all critical functionality and user interactions to ensure a seamless user experience.

- **Monitor Performance:** Monitor the performance of the deployed application in the production environment. Keep an eye on factors such as response times, resource utilization, and error rates to identify any potential issues and optimize performance.

By following these steps, you can successfully deploy the web application to a production environment, making it accessible to users while ensuring reliability, scalability, and performance.

5.7 Monitoring and Maintenance

After deployment, ongoing monitoring and maintenance are essential to ensure the continued performance, security, and reliability of the application:

- **Performance Monitoring:** Regularly monitor the application's performance using monitoring tools and techniques. Keep track of key performance indicators such as response times, server load, and resource utilization to identify any performance bottlenecks or issues.
- **Security Measures:** Implement robust security measures to protect the application and its data from potential threats. This includes regular security audits, applying security patches and updates, and implementing best practices for data encryption, access control, and authentication.
- **Data Management:** Ensure the integrity and availability of data by implementing regular backup procedures. Backup critical data at regular intervals to prevent data loss in the event of system failures, security breaches, or other unforeseen events.
- **Support and Maintenance:** Provide ongoing support and maintenance to address any issues that arise and ensure the smooth operation of the application. This includes responding to user inquiries, troubleshooting technical issues, and making enhancements or improvements based on user feedback and evolving requirements.
- **Scalability and Optimization:** Monitor the application's usage patterns and performance metrics to identify opportunities for scalability and optimization. Adjust resources and infrastructure as needed to accommodate growing user demand and improve overall system performance.

By prioritizing monitoring and maintenance activities, you can ensure that the web application remains secure, reliable, and performant over time, providing a positive user experience and maximizing its value to stakeholders.

5.8. Libraries Used:

5.8.1. For JSP Codes

1. Java I/O Libraries

Java provides several I/O libraries for handling input and output operations:

- **java.io.InputStreamReader**: `InputStreamReader` is a class used for reading bytes from an `InputStream` and decoding them into characters using a specified character set. It allows you to specify the character encoding, such as UTF-8 or ISO-8859-1, to properly decode the bytes into characters.
- **java.io.BufferedReader**: `BufferedReader` is a class used for reading text from a character-input stream efficiently. It reads characters from a `Reader` (such as an `InputStreamReader`) and buffers them to provide efficient reading of characters, arrays, and lines. `BufferedReader` is often used to read text files line by line.
- **java.io.InputStream**: `InputStream` is an abstract class representing an input stream of bytes. It serves as the superclass for all classes representing an input stream of bytes. `InputStream` provides basic functionality for reading bytes from various sources, such as files, network connections, or in-memory byte arrays.

These I/O libraries in Java provide versatile capabilities for handling input and output operations, allowing developers to read and process data from different sources efficiently and reliably.

2. Java SQL Libraries

- Java SQL Libraries include the `java.sql.*` package, which provides essential interfaces and classes for the JDBC (Java Database

Connectivity) API. JDBC facilitates the interaction between Java applications and relational databases, allowing developers to perform database operations such as querying, updating, and managing data.

- ➔ This package encompasses a range of functionalities necessary for database connectivity and manipulation. It includes interfaces like `Connection`, `Statement`, `PreparedStatement`, and `ResultSet`, along with classes for handling SQL exceptions and managing database metadata.
- ➔ With JDBC, developers can establish connections to databases, execute SQL queries, retrieve query results, and perform transactions programmatically within Java applications. This enables seamless integration between Java applications and various database systems, empowering developers to build robust and data-driven software solutions.

3. Custom Database Connection Class

- ➔ The `'novefficient.Dbconnection'` class seems to be a custom implementation for managing database connections in a Java application. It is likely designed to encapsulate the logic required for establishing connections to a database using JDBC (Java Database Connectivity).
- ➔ This custom class may include methods for initializing database connections, executing SQL queries, handling database transactions, and managing resources associated with database operations. It abstracts away the complexities of establishing and managing database connections, providing a convenient and reusable solution for interacting with the underlying database system.
- ➔ By encapsulating database connectivity logic within a custom class, developers can achieve better code organization, reusability, and maintainability. It promotes cleaner and more modular code, simplifying the process of integrating database functionality into Java applications while adhering to best practices for database interaction.

4. Session Management Directive

- ➔ The `<%@ page session="true" %>` directive is used within a JavaServer Pages (JSP) file to enable session management for that specific page. When this directive is included at the top of a JSP file, it signifies that the page requires session tracking capabilities.
- ➔ Enabling session management allows the JSP page to maintain user sessions across multiple HTTP requests. This means that session attributes, such as user authentication details or shopping cart items, can be stored and accessed throughout the user's interaction with the application.
- ➔ By including this directive, developers ensure that the JSP page has access to session-related functionalities provided by the servlet container. This includes features like creating and accessing session objects, setting session attributes, and managing session lifecycles.
- ➔ In essence, the `<%@ page session="true" %>` directive facilitates the implementation of session-based functionality within JSP pages, enabling developers to build dynamic and interactive web applications that maintain stateful interactions with users.

5. HTML, CSS, JavaScript Libraries

- ➔ `templatemo_style.css`: This CSS file is used for styling the HTML elements of the web page, providing visual layout and design.
- ➔ `css/ddsmoothmenu.css`: This CSS file is likely used for styling dropdown menus, providing a smooth navigation experience.
- ➔ `js/jquery.min.js`: This JavaScript library is jQuery, a popular JavaScript framework used for simplifying client-side scripting and DOM manipulation.
- ➔ `js/ddsmoothmenu.js`: This JavaScript file contains code for initializing and configuring the Smooth Navigational Menu, a dropdown menu plugin based on jQuery.
- ➔ `js/jquery.ennui.contentslider.css`: This CSS file styles the content slider plugin used for displaying sliding content sections on the web page.
- ➔ `js/jquery.easing.1.3.js`: This JavaScript file contains easing functions for smooth animation transitions used by the content slider.

→ ``js/jquery.ennui.contentslider.js``: This JavaScript file contains the code for the content slider plugin used on the web page.

5.8.2. Encryption & Decryption

1. ``com.sun.org.apache.xerces.internal.impl.dv.util.Base64``:

This library provides utility methods for encoding and decoding data using the Base64 encoding scheme. Base64 encoding is commonly used to represent binary data as ASCII text, making it suitable for tasks like encoding binary data for transmission over text-based protocols or storing binary data in text-based formats.

2. ``java.io.ByteArrayOutputStream``:

The ``ByteArrayOutputStream`` class from the ``java.io`` package is used for writing binary data into a byte array. It's often used in conjunction with other stream classes to collect binary data in memory before further processing, such as encryption or serialization.

3. ``java.io.FileInputStream``:

This library is used to read data from files. The ``FileInputStream`` class from the ``java.io`` package reads bytes from a file and is often used to supply data to processes that require input from files, such as encryption or reading configuration files.

4. ``java.io.FileWriter``:

The ``FileWriter`` class from the ``java.io`` package is used for writing character-oriented data to files. While not directly related to encryption, it could be used to log or write encrypted data to a file for storage or further processing.

5. ``java.util.Scanner``:

The ``Scanner`` class from the ``java.util`` package is used for parsing primitive types and strings from input streams. It can be helpful for reading input from various sources, including files or user input, which may be necessary for providing data to encryption processes.

6. ``javax.crypto.Cipher``:

The ``Cipher`` class from the ``javax.crypto`` package is a fundamental class in the Java Cryptography Architecture (JCA). It provides encryption and decryption capabilities and supports various cryptographic algorithms. In this context, it's used to perform AES encryption on data.

7. `javax.crypto.KeyGenerator`:

The `KeyGenerator` class from the `javax.crypto` package is used to generate secret keys for symmetric encryption algorithms like AES. It provides a convenient way to generate random keys of the appropriate length for use in encryption operations.

8. `javax.crypto.SecretKey`:

The `SecretKey` interface from the `javax.crypto` package represents a cryptographic secret key used for encryption and decryption. It's a common interface for handling symmetric keys generated by `KeyGenerator` or created from raw key bytes.

9. `javax.crypto.spec.SecretKeySpec`:

The `SecretKeySpec` class from the `javax.crypto.spec` package provides a way to create a `SecretKey` from raw key bytes. It's useful for specifying keys in encryption and decryption operations when the key material is known in advance.

10. `javax.swing.JOptionPane`:

The `JOptionPane` class from the `javax.swing` package is used to create dialog boxes for displaying messages or prompting the user for input. While not directly related to encryption, it could be used for displaying messages or interacting with the user during encryption operations.

11. `sun.misc.BASE64Encoder`:

The `BASE64Encoder` class from the `sun.misc` package is used to encode binary data as a Base64-encoded string. Base64 encoding is commonly used for representing binary data as text, making it suitable for scenarios where binary data needs to be transmitted or stored as text.

12. `sun.misc.BASE64Decoder`:

The `BASE64Decoder` class from the `sun.misc` package is used to decode Base64-encoded data into its original binary form. It complements the Base64 encoding functionality provided by the `Base64` class, enabling the decoding of Base64-encoded strings back into their original binary representation.

5.8.3. Database Connection:

1. `java.sql.Connection`: This class represents a connection to a database. It is part of the JDBC (Java Database Connectivity) API and is used to establish a connection to a database server.

2.java.sql.DriverManager: The DriverManager class is part of the JDBC API and is used to manage a set of JDBC drivers. It provides methods for registering drivers, establishing database connections, and controlling the drivers' behavior.

3.java.lang.Class: This class is part of the core Java API and is used for obtaining information about classes loaded in the Java Virtual Machine (JVM). In this context, it's used to load the MySQL JDBC driver dynamically at runtime.

4.com.mysql.jdbc.Driver: This class is the MySQL JDBC driver implementation provided by MySQL. It allows Java applications to connect to MySQL databases using JDBC. The Class.forName() method is used to load this driver class dynamically.

5.9 Parameters

1. **Encryption:** Process of converting plaintext data into ciphertext using an encryption algorithm and a secret key.

$$EAES=f(s,k)$$

Where, EAES: Efficiency of AES encryption.

s: Size of the input plaintext data.

k: Length of the AES encryption key

2. **Decryption:** Process of converting ciphertext back into plaintext using a decryption algorithm and the corresponding decryption key.

$$DAES=f(s,k)$$

Where, DAES: Efficiency of AES decryption.

s: Size of the encrypted ciphertext data.

k: Length of the AES decryption key

3. **Encryption Time (ET):** This parameter measures the time taken to encrypt a given amount of data. It is typically expressed in seconds. The formula to calculate encryption time is:

$$ET=Te-Ts$$

Where:

T_e is the time when the encryption process finishes.

T_s is the time when the encryption process starts.

4. **Decryption Time (DT):** Similar to encryption time, decryption time measures the time taken to decrypt encrypted data. It is also expressed in seconds. The formula to calculate decryption time is:

$$DT = T_d - T_e$$

Where:

T_d is the time when the decryption process finishes.

T_e is the time when the encryption process finishes.

5. **Key Generation:** It generates a random public key and a random private key for encryption using AES. It then converts the generated AES secret key into a string representation using Base64 encoding. The formula for key generation success probability (PAES) can be expressed as:

$$P_{AES} = \frac{1}{R^L}$$

Where, PAES: Probability of AES key generation success.

L: Length of the AES secret key.

R: Range of possible values for the generated keys.

6. **Access Control Policy:** It defines rules and regulations governing access to resources in a system. A generic representation of an access control policy can be expressed as:

$$\text{Policy} = \{\text{Rule1}, \text{Rule2}, \dots, \text{Rulen}\}$$

Where, Policy: Access control policy.

Rule1, Rule2, ..., Rulen: Individual access control rules.

Each rule can be defined as: Rule $_i$ = (Conditions, Actions)

Where, Conditions: Set of conditions that must be satisfied for the rule to apply.

Actions: Set of actions or permissions granted if the conditions are met.

6. Discussion of Results

The efficacy of data encryption techniques lies in their ability to uphold confidentiality, ensuring that sensitive information remains beyond the reach of unauthorized entities. Additionally, the encryption process serves to bolster data integrity and confidentiality, enhancing overall security measures. Various encryption algorithms, including AES, are employed to encrypt data of diverse types and sizes. Parameters such as key lengths and modes of operation are deliberately altered to assess their influence on encryption strength and speed. Through comparative analysis, encryption times for both existing and proposed methods are evaluated based on the values presented in Table 6.1

| Input File (KB) | Existing Method | Proposed Method |
|------------------------|------------------------|------------------------|
| 1 | 50 | 45 |
| 2 | 120 | 105 |
| 3 | 207 | 190 |
| 4 | 250 | 230 |
| 5 | 303 | 275 |

TABLE 6.1 ENCRYPTION TIME

The findings are visually depicted in Figure 6.1, showcasing the graph of encryption times for comprehensive analysis and interpretation.



FIGURE 6.1 ENCRYPTION TIME GRAPH

Similar to the encryption process, the decryption of encrypted data involves the utilization of AES algorithms to decipher the encoded information. This entails evaluating the speed and precision of the decryption process to ensure efficient and accurate data retrieval. A comparative analysis of decryption performance is conducted based on the data presented in Table 6.2, allowing for an assessment of decryption speed and accuracy across different scenarios or methods.

| Input File (KB) | Existing Method | Proposed Method |
|-----------------|-----------------|-----------------|
| 1 | 90 | 85 |
| 2 | 160 | 147 |
| 3 | 210 | 190 |
| 4 | 285 | 250 |
| 5 | 315 | 275 |

TABLE 6.2 DECRYPTION TIME

The findings are visually represented in Figure 6.2, illustrating the results of the decryption process for further analysis and interpretation.

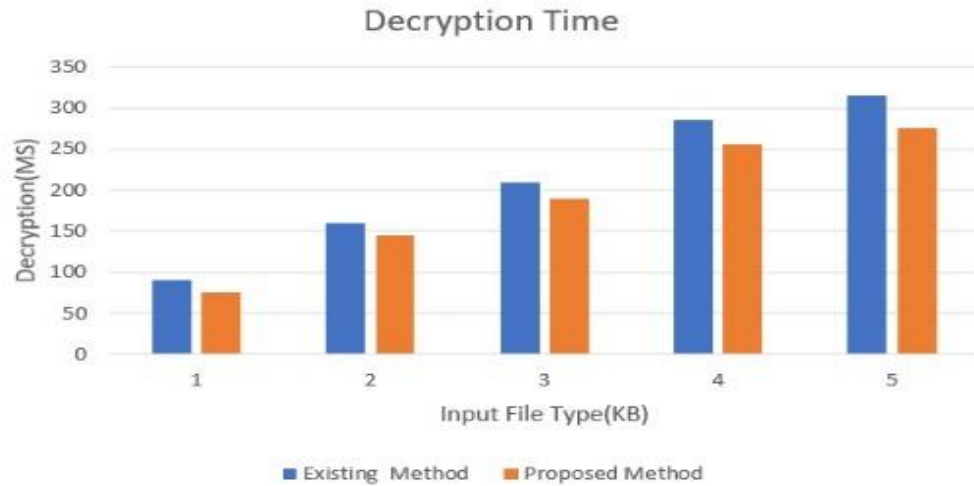


FIGURE 6.2 DECRYPTION TIME GRAPH

7. Conclusion

The findings of this research highlight the efficacy of the proposed security and privacy mechanisms in thwarting unauthorized access to patient data. This significantly enhances confidentiality and fortifies healthcare services against potential vulnerabilities and cyber threats. Through the implementation of robust measures such as public and private key encryption and stringent password protection, remote access is carefully controlled, thereby safeguarding the confidentiality of sensitive medical information.

By reinforcing security protocols and safeguarding patient data, the overarching objective is to enhance patient care, elevate the standards of medical practice, and ensure the integrity of healthcare services in the modern digital landscape. This ensures that healthcare organizations can operate with confidence in the face of evolving cybersecurity challenges, ultimately benefiting both patients and healthcare providers alike.

8. Future Enhancements

For future enhancements, several avenues could be explored to further improve the security, efficiency, and usability of the healthcare system:

Enhanced Data Encryption Techniques: Researching and implementing advanced encryption techniques beyond AES, such as homomorphic encryption or post-quantum cryptography, can provide stronger protection for patient data against emerging threats and vulnerabilities.

Blockchain Technology for Data Integrity: Integrating blockchain technology can ensure the integrity and immutability of patient records, enhancing trust and transparency in the healthcare system while preventing unauthorized alterations to medical data.

AI-Powered Threat Detection: Leveraging artificial intelligence (AI) and machine learning algorithms for proactive threat detection can help identify and mitigate cybersecurity risks in real-time, ensuring continuous protection against evolving threats.

Improved User Experience: Continuously refining the user interface and experience based on user feedback and usability studies can enhance the accessibility and efficiency of the healthcare system for both patients and healthcare professionals.

Interoperability with External Systems: Enhancing interoperability with external healthcare systems and medical devices can facilitate seamless data exchange and collaboration, improving the overall efficiency and effectiveness of patient care.

Continuous Monitoring and Auditing: Implementing robust monitoring and auditing mechanisms to track access to patient data, detect anomalies, and ensure compliance with security policies can further enhance the overall security posture of the healthcare system.

By focusing on these areas for future enhancements, the healthcare system can continue to evolve and adapt to meet the growing demands for security, privacy, and efficiency in healthcare delivery.

9. References

- [1] S. M. Nagarajan, G. G. Deverajan, U. Kumaran, M. Thirunavukkarasan, M. D. Alshehri and S. Alkhalaf, "Secure Data Transmission in Internet of Medical Things Using RES-256 Algorithm," in IEEE Transactions on Industrial Informatics
- [2] O. Kocabas, T. Soyata, and M. K. Aktas, "Emerging security mechanisms for medical cyber-physical systems," IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 13, no. 3, pp. 401–416, May/Jun. 2016
- [3] P. Gope, Y. Gheraibia, S. Kabir, and B. Sikdar, "A secure IoT-based modern healthcare system with fault-tolerant decision-making process," IEEE J. Biomed. Health Informat., vol. 25, no. 3, pp. 862–873, Mar. 2021.
- [4] H. Qiu, M. Qiu, M. Liu, and G. Memmi, "Secure health data sharing for medical cyber-physical systems for the healthcare 4.0," IEEE J. Biomed. Health Informat., vol. 24, no. 9, pp. 2499–2505, Sep. 2020
- [5] Z. Chkirbene, R. Hamila and A. Erbad, "Secure Medical Data Sharing For Healthcare System," 2022 IEEE 33rd Annual International Symposium Communications (PIMRC), Kyoto, Japan, 2022.
- [6] J. Kassem, C. De Laat, A. Taal and P. Grosso, "The EPI Framework: A Dynamic Data Sharing Framework for Healthcare Use Cases", 2020.
- [7] N. Tsafack et al., "A new chaotic map with dynamic analysis and encryption application in internet of health things," IEEE 2020.
- [8] S. Wijethilaka and M. Liyanage, "Survey on network slicing for Internet of Things realization in 5G networks," IEEE 2021.
- [9] R. Saha, G. Geetha, G. Kumar, T.-H. Kim, and W. J. Buchanan, "Mrc4: A modified RC4 algorithm using symmetric random function generator for improved cryptographic features," IEEE 2019.
- [10] M. A. Siddiqi, C. Doerr, and C. Strydis, "Imdfence: Architecting a secure protocol for implantable medical devices," IEEE 2020.
- [11] Lim, C.K., Ipinige, V.J., Tan, K.L., & Hambira, N. Design and development of message authentication process for telemedicine application. In 2018 IEEE Conference on Wireless Sensors.

- [12] Jan, M.A., Khan, F., Mastorakis, S., Adil, M., Akbar, A., & Stergiou, N. (2021a). LightIoT: Lightweight and secure communication for energy-efficient IoT in health informatics. *IEEE Transactions on Green Communications and Networking*.
- [13] O. Salem, K. Alsubhi, A. Shaafi, M. Gheryani, A. Mehaoua and R. Boutaba, "Man in the middle attack mitigation in internet of medical things", *IEEE Trans. Ind. Informat.*.
- [14] L. Feng, A. Ali, M. Iqbal, A. K. Bashir, S. A. Hussain and S. Pack, "Optimal haptic communications over nanonetworks for e-health systems", *IEEE*, May 2019.
- [15] P. Gope, Y. Gheraibia, S. Kabir and B. Sikdar, "A secure IoT-based modern healthcare system with fault-tolerant decision making process", *IEEE* Mar. 2021.
- [16] K. Sowjanya, M. Dasgupta and S. Ray, "An elliptic curve cryptography based enhanced anonymous authentication protocol for wearable health monitoring systems", 2020.
- [17] P. Gope, J. Lee and T. Q. Quek, "Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions", *IEEE* Nov. 2018.
- [18] M. Aazam, K. A. Harras, and S. Zeadally, "Fog computing for 5G tactile industrial Internet of Things: QoE-aware resource allocation model," *IEEE* May 2019.
- [19] Y. Guo, D. Tao, W. Liu and J. Cheng, "Multiview cauchy estimator feature embedding for depth and inertial sensor-based human action recognition", *IEEE*, Apr. 2017.